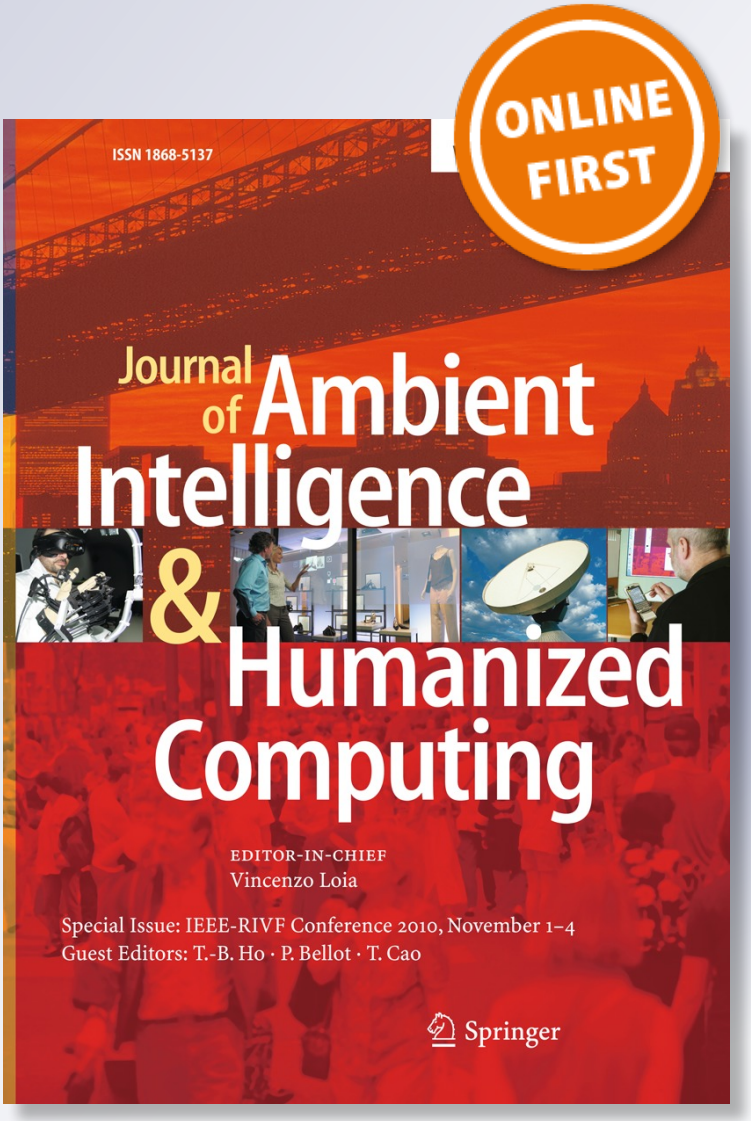
Inferring social ties between users with human location history

Xiangye Xiao, Yu Zheng, Qiong Luo & Xing Xie

Journal of Ambient Intelligence and Humanized Computing

ISSN 1868-5137

J Ambient Intell Human Comput
DOI 10.1007/s12652-012-0117-z



Springer

J Ambient Intell Human Comput
DOI 10.1007/s12652-012-0117-z

ORIGINAL RESEARCH

Inferring social ties between users with human location history

Xiangye Xiao · Yu Zheng · Qiong Luo · Xing Xie

Received: 20 October 2011 / Accepted: 26 March 2012

Abstract The location-based social networks have been becoming flourishing in recent years. In this paper, we aim to estimate the similarity between users according to their physical location histories (represented by GPS trajectories). This similarity can be regarded as a potential social tie between users, thereby enabling friend and location recommendations. Different from previous work using social structures or directly matching users' physical locations, this approach model a user's GPS trajectories with a semantic location history (SLH), e.g., *shopping malls* → *restaurants* → *cinemas*. Then, we measure the similarity between different users' SLHs by using our maximal travel match (MTM) algorithm. The advantage of our approach lies in two aspects. First, SLH carries more semantic meanings of a user's interests beyond low-level geographic positions. Second, our approach can estimate the similarity between two users without overlaps in the geographic spaces, e.g., people living in different cities. When matching SLHs, we consider the sequential property, the granularity and the popularity of semantic locations. We evaluate our method based on a real-world GPS dataset collected by 109 users in a period of 1 year. The results show that SLH outperforms a physical-location-based approach and MTM is more effective than several widely used sequence matching approaches given this application scenario.

X. Xiao · Y. Zheng (✉) · X. Xie
Microsoft Research Asia, Beijing, China
e-mail: yuzheng@microsoft.com

X. Xiao
e-mail: xiangye.xiao@gmail.com

X. Xiao · Q. Luo
Hong Kong University of Science and Technology,
Hong Kong, China

Keywords Location-based social networks · User similarity · Social ties · GPS trajectory · Location history · Semantic location history · Sequential matching

1 Introduction

As the location-based social networks (Zheng 2011a) become popular in recently years, an increasing number of people start using GPS-enabled devices to log their outdoor movements with GPS trajectories (Zheng et al. 2008c, 2009a, 2010c, 2011e). These trajectories do not only record users' location histories in the physical world but also imply their personal interests and preferences (Eagle et al. 2006; Zheng et al. 2011a, b). Meanwhile, trajectories generated by a large number of people imply rich social and community intelligent (Zhang et al. 2011). Figure 1 demonstrates the mobility of four individuals (*A*, *B*, *C*, and *D*) who respectively recorded a one-day trip with a GPS trajectory. According to the outdoor movement, we can observe the following three insights.

1. *Geographic overlaps*: People having similar outdoor location histories (or mobility patterns) in the geographic spaces could share some similar life interests. For instance, the users *A* and *B* might share some similar interests as both of them have visited the same cinema, museum, coffee, and shopping mall (although they may not know each other personally).
2. *Semantic overlaps*: People could share some similar interests if they have similar mobility patterns in the space of semantic locations. For example, though the user *C* does not access the same locations with *B*, the semantic meanings (categories) of the locations

Published online: 02 December 2012

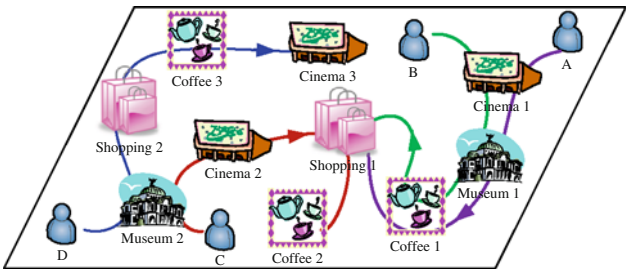


Fig. 1 GPS trajectories and user interests

(museum, cinema and coffee) are the same with that of *B*. That is, they could still share similar interests.

3. *Location sequence*: Although the user *D* also visited a museum, a coffee shop, a shopping mall, and a cinema, the sequence between these locations ("museum → shopping mall → coffee → cinema") is different from that of the users *B* and *C* ("cinema → museum → coffee → shopping mall"). Thus, the similarity between *D* and *B* might not be as significant as that between *C* and *B*.

In this paper, we aim to estimate the similarity between users according to the semantic location histories (SLH) inferred from their GPS trajectories. This similarity can regarded as potential social ties between users, thereby enabling friend and location recommendation. For instance, as shown in Fig. 1, if knowing the users B and C are similar according to their location histories, we are able to recommend user B to user C in a social networking service. As a result, they may connect to each other, i.e., creating a social tie between them. Further, we can recommend the museum 1 and cinema 1 to the user C, and provide the user B with the museum 2 and cinema 2 as a recommendation.

The two essential steps of finding similar users are (1) modeling users' interests from their historical GPS trajectories and (2) measuring the similarity between them based on their location histories. Both tasks are non-trivial.

First, we model a user's movements as a sequence of semantic locations (categories), e.g., "museum → cinema → restaurant", instead of physical locations. Semantic locations are more informative in capturing the interests of users than physical geo-positions. Additionally, this semantic history enables us to detect the similar users without overlaps in the geographic space, e.g., the users B and C shown in Fig. 1. However, there exists uncertainty in identifying the location category of a GPS point due to the positioning errors of GPS devices and the crowded distribution of points of interests (POIs) in a city (sometimes, many POIs are located in the same building).

Second, it is non-trivial to estimate the similarity between users by measuring the semantic location sequences. Our first intuition is that users sharing a longer sequence of semantic locations would be more similar than those sharing a shorter one. For example, people sharing a sequence "museum → restaurant" would be more similar than those visiting these two categories separately. In addition, a fine semantic location, e.g., "art museum" is more informative in reflecting users' interests than a coarse one, e.g., "museum". Thus, users sharing semantic locations with a coarse granularity would be less similar than those sharing a finer granularity. Furthermore, semantic locations with different popularities contribute differently to the similarity between users. Intuitively, users sharing a category visited by a few people, e.g., "museum" would be more similar than those sharing a very common category, e.g., "restaurant".

To address these problems, we first construct for each user a SLH based on their historical GPS trajectories. Then, we compute the similarity between different users in terms of their SLHs, considering the sequence, granularity and popularity features mentioned above. The contributions of our work include:

1. The SLH well models a user's interests and considers the uncertainty of the semantic meanings of a place where a user stayed. Specifically, the SLH transfers a user's location history from raw GPS trajectories to a set of high-level semantic-location-sequences on different levels of a hierarchy representing different granularities of location categories.
2. We design the maximal travel match (MTM) algorithm to compare location histories of different users. MTM finds out the maximal subsequence matches instead of simply counting common locations. Moreover, by incorporating travel time between two locations, MTM is more capable of finding common sub-sequences with meaningful visiting orders beyond existing work on sequence matching, such as Edit Distance (ED) (Levenshtein 1966), Longest Common Sub-Sequences (LCSS) (Vlachos et al. 2002) and Dynamic Time Warping (DTW) (Yi et al. 1998).
3. We evaluated our approach on real-world GPS data collected by 109 users over a year. The results demonstrate the advantages of SLH and MTM over baselines respectively (the dataset has been released to the public (GeoLife GPS trajectories 2010) to facilitate other professionals' research).

The rest of this paper is organized as follows. Section 2 introduces the preliminary and architecture of this work. Section 3 describes the location history modeling, and Sect. 4 details the location history matching. Section 5 gives an optimal solution using the similarity in a real system. Later, we report on the evaluation results in Sect. 6 and discuss the related work in Sect. 7. Finally, we conclude our work in Sect. 8.

2 Preliminary

Definition 1 (*GPS Trajectory*) A GPS trajectory *Tra* is a sequence of time-stamped points, $Tra = p_0 \rightarrow p_1 \rightarrow ,\ldots, \rightarrow p_k$ where $p_i = (x, y, t)(i = 0, 1, \ldots, k)$; (x, y) are latitude and longitude respectively, and t is a timestamp. $\forall 0 \le i \le k, p_{i+1} \cdot t > p_i \cdot t$.

Definition 2 (*Stay Point*) A stay point s is a geographical region where a user stayed over a time threshold θ_t within a distance threshold θ_d. In a trajectory, s is characterized by a set of consecutive points $P = \langle p_m, p_{m+1}, \ldots, p_n \rangle$, where $\forall m < i \le n$, $Dist(p_m, p_i) \le \theta_d$, $Dist(p_m, p_{n+1}) > \theta_d$ and $Int(p_m, p_n) \ge \theta_t$. Therefore, $s = (x, y, t_a, t_l)$, where

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

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

respectively stands for the average x and y coordinates of the collection P; $s.t_a = p_m \cdot t$ is the user's arriving time on s and $s.t_l = p_n \cdot t$ represents the user's leaving time.

As depicted in Fig. 2, $\{p_1, p_2, \ldots, p_8\}$ formulate a GPS trajectory, and a stay point would be detected from $\{p_3, p_4, p_5, p_6\}$ if $d \le \theta_d$ and $Int(p_3, p_6) \ge \theta_t$. In contrast to a raw point p_i, a stay point carries a particular semantic meaning, such as a shopping mall or a restaurant a user accessed.

Figure 3 presents the architecture of our work, which consists of two major steps: location history modeling and location history matching. Given (1) GPS trajectories of multiple users and (2) a POI database, our objective is to infer the user similarity score of each pair of users. Later, this similarity can be used by some existing clustering algorithms, like K-means and KNN, as a distance function to cluster users into different groups. Therefore, we can easily find out top k similar users of a person by ranking others in the person's group according to the similarity scores.

In order to make different users' location histories comparable, we first put all users' GPS trajectories together and create a shared framework of location history. Here, a POI database is employed to transfer a user's location history from geographic spaces into the semantic spaces. The POI database contains a corpus of POI entities, each of which includes the properties of category, latitude and longitude, etc. Then, based on the framework we can respectively build a location history for each user. Later, for each pair of users, we explore their similarity by matching their location histories. We will provide more details of the architecture in the following sections.

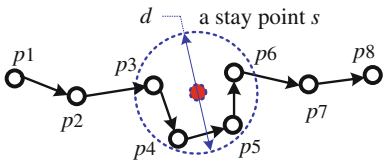


Fig. 2 A GPS trajectory and a stay point

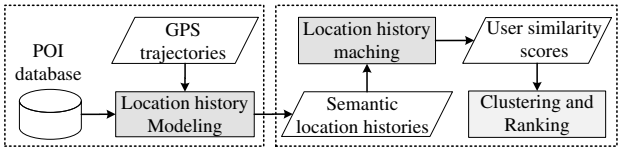


Fig. 3 The architecture of similar user discovery

3 Location history modelling

Figure 4 shows the process of modeling location history for each user, and Fig. 6 gives a demonstration. This step is comprised of three components denoted as grey boxes in Fig. 4 and described respectively in the following subsections.

3.1 Stay points representation

In this component, we first extract stay points from each user's GPS trajectories by using a stay point detection method proposed in paper (Li et al. 2008). These stay points carry more semantic meanings beyond raw GPS points, and allow us to filter the places where a user only passed by, e.g., crossroads.

However, it is almost impossible to identify the exact POI a user visited according to a stay point, given the GPS positioning error and crowded distribution of POIs in a

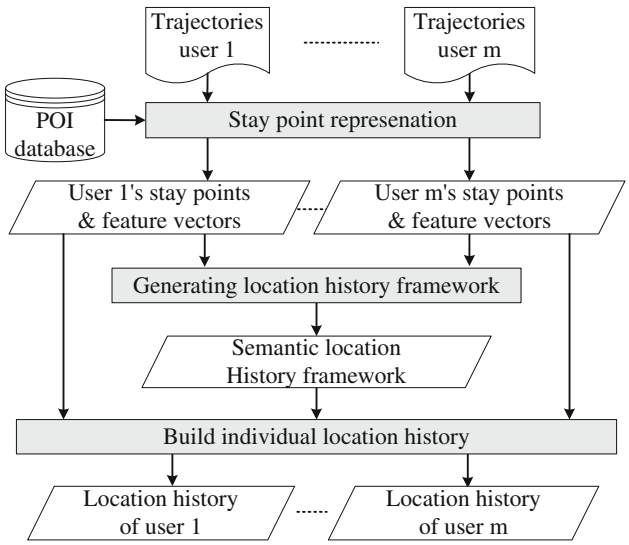


Fig. 4 The procedure of modeling user location history

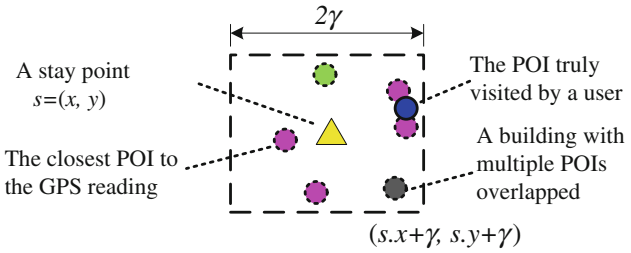


Fig. 5 The architecture of similar user discovery

city. In practice, as shown in Fig. 5, a GPS reading may have a 10 m or more error to the real position. Naturally, there could be multiple POIs pertaining to different categories exist in such a distance range, while the nearest POI to the stay point may not be the real place that a user visited. Sometimes, restaurants, shopping malls, and cinemas are even overlapped in the same building.

In this work, we represent a stay point as a [s.x $-$ γ, s.x $+$ γ] $\times$ [s.y $-$ γ, s.y $+$ γ] region (refer to Fig. 5 for an example), where γ is a parameter related to the GPS positioning error. After that, we construct a feature vector for each stay region according to the POIs fallen in the region. Here, we employ the idea of TF-IDF (term frequency-inverse document frequency), which is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Similarly, we regard categories of POIs as words and treat the users' stay regions as documents. Intuitively, if POIs of a category occur in a region many times, this POI category is important in representing this region. Furthermore, a POI category (e.g., "museum" and "natural parks") that occurs rarely in other regions is more representative for the region than a common POI category, e.g., "restaurant", which could appear in many places. Thus, we consider both the occurring frequency of a POI category in a region (similar to TF) and the inverse location frequency (equivalent to IDF) of this category. Combining these two factors, we design the feature vector as follows.

Definition 3 (*Feature Vector*) The feature of a stay region r in a collection of regions R is $f_r = < w_1, w_2 \ldots w_F >$, where w_i is the weight of POI category i in region r. F is the number of unique POI categories in a POI database.

$$w_i = \frac{n_i}{N} \times \log \frac{|R|}{|\{Regionscontaining\, i\}|'} \tag{3}$$

where n_i is the number of POIs of category i located in region r and N stands for the total number of POIs in region r. The first part of Eq. 3 represents the occurring frequency of a category and the second part denotes the inverse location frequency of a category, in which $|R|$ is the number of regions in the collection.

According to Eq. 3, we can represent a stay region with a feature vector (refer to the top part of Fig. 6 for an example). Although we still cannot identify the exact POI category visited by a user, this feature vector catches the interests of a user to some extent by representing the semantic meaning of a location. In short, the feature vector reflects on the uncertainty of accessed categories while bypasses the difficulties in identifying the exact POI visited by a user.

3.2 Generating location history framework

In the second component, as demonstrated in Fig. 6, we cluster the stay regions into some groups according to their feature vectors. The stay regions in the same cluster can be regarded as locations of the similar type and having similar semantic meanings. However, a flat clustering is insufficient in differentiating similar users of different degrees. Intrinsically, we are more capable of discriminating similar users given categories with a finer granularity. For example, "restaurant" helps identify users who like dining outside, while "Indian restaurant" and "Japanese restaurant" enable us to differentiate people interested in different types of food.

Considering this factor, we hierarchically cluster the feature vectors in a divisive manner and build a tree-structured semantic location hierarchy. As shown in the middle part of Fig. 6, we start with putting feature vectors of all users into one cluster and treat this cluster as the root (i.e., cluster at layer 1). For each cluster c at layer $j (j \geq 0)$, we split c into a set of sub-clusters by using a flat clustering algorithm. The result sub-clusters of c are considered as c's child nodes at layer $j + 1$. This procedure repeats a given number of times L. As a result, we create a tree-structured hierarchy where clusters at the same layer share the same granularity and a lower layer denotes a finer granularity.

Definition 4 (*Semantic Location*) A semantic location c is a feature vector cluster and represents a set of stay regions sharing similar semantic meanings of a certain granularity.

Definition 5 (*Semantic Location Hierarchy*) A semantic location hierarchy $\mathcal{F}$ is a tree-structured framework in the feature vector space, $\mathcal{F} = \cup_{l=1}^{L}\{C^l\}$, where L is the total number of layers; $C^l = \{c_{l1}, c_{l2}, \ldots, c_{lk}\}$ is the set of semantic locations at layer l, and c_{lk} denotes the kth semantic location on the lth layer.

3.3 Building individual location history

In this component, we construct a location history for each user based on the semantic location hierarchy $\mathcal{F}$ and the

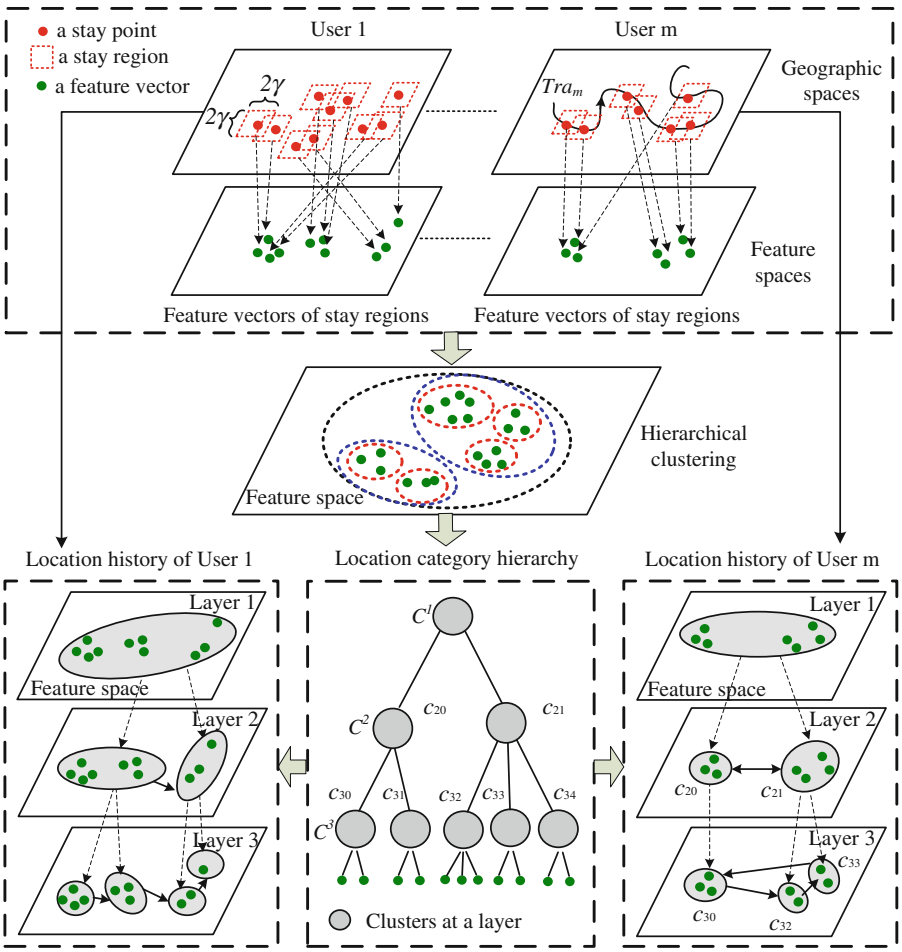


Fig. 6 The demonstration of location history modeling

user's stay points (illustrated in Fig. 6). Originally, a user's location history in the geographic spaces is represented by a sequence of stay points with traveling time between each two consecutive stay points. Then, on each layer of the semantic location hierarchy $\mathcal{F}$, we respectively substitute a stay point with the semantic location that the stay point's feature vector pertains to. After this projection, different users' location histories become comparable.

Definition 6 (*Semantic Location History*) A user's semantic location history is a sequence of semantic locations on each layer of $\mathcal{F}$, $H = \cup_{l=1}^{L}\{S^l\}$, where $S^l = (c_{l0} \xrightarrow{\Delta t_1} c_{l1} \xrightarrow{\Delta t_2} \ldots \xrightarrow{\Delta t_k} c_{lk})$ is the sequence on the lth layer of $\mathcal{F}$. Suppose having two consecutive stay points s_{k-1} and s_k, $s_{k-1} \in c_{l,k-1}$ and $s_k \in c_{lk}$, then $\Delta t_k = s_k.t_a - s_{k-1}.t_l$ is the traveling time from $c_{l,k-1}$ to c_{lk}.

Example 1 (*Location History*) As demonstrated in the up-right part of Fig. 6, according to trajectory Tra_m user m's location history can be represented by

$$H = (s_1 \xrightarrow{\Delta t_1} s_2 \xrightarrow{\Delta t_2}, \ldots, \xrightarrow{\Delta t_7} s_7),$$

where $\Delta t_i = s_{i+1}.t_a - s_i.t_l$ is the traveling time between s_i and s_{i+1}. Then, we extend each stay point into a stay region and calculate the feature vector of each stay region as follows. Suppose that s_1 contains two restaurants and one museum, and s_2 only have four restaurants. The total number of stay regions created by all the users is 100, in which 50 have restaurants and two contain museums. So, the feature vectors of s_1 and s_2 are f_1 and f_2 respectively:

$$f_1 = \left(\frac{2}{3} \times \log\frac{100}{50}, \frac{1}{3} \times \log\frac{100}{2}, \ldots,\right),$$

$$f_2 = \left(\frac{4}{4} \times \log\frac{100}{50}, \quad 0, \ldots,\right),$$

After hierarchically clustering these stay regions in the feature spaces, we build a $\mathcal{F}$. Later, by replacing a stay point with the cluster ID (semantic location) the point's feature vector pertaining to, we can obtain two sequences, S^2 and S^3, on the second and third layer of $\mathcal{F}$ separately.

$S^2 = (c_{20} \xrightarrow{\Delta t_1} c_{20} \xrightarrow{\Delta t_2} c_{21} \xrightarrow{\Delta t_3} c_{21} \xrightarrow{\Delta t_4} c_{21} \xrightarrow{\Delta t_5} c_{21} \xrightarrow{\Delta t_6} c_{20})$,

$S^3 = (c_{30} \xrightarrow{\Delta t_1} c_{30} \xrightarrow{\Delta t_2} c_{32} \xrightarrow{\Delta t_3} c_{32} \xrightarrow{\Delta t_4} c_{33} \xrightarrow{\Delta t_5} c_{33} \xrightarrow{\Delta t_6} c_{30})$,

So, user m's location history can be represented as $H = \{S^2, S^3\}$.

4 Location history matching

In this section, we explore the similarity between each pair of users by matching their semantic location histories. As shown in Fig. 7, we first match two users' semantic sequences at each layer of $\mathcal{F}$ (i.e., MaxTravMatch), and then calculate a similarity score for each pair of user (i.e., CalculateSimilarity) by aggregating the matched sub-sequences at all layers in a weighted manner. Table 1 shows the notations used in this paper.

4.1 Maximal travel match

A simple way coming to people's mind is to count the number of items shared by two sequences. However, this method will lose lots of information about a user's behavior and preferences, and leaves sequences sharing the same number of common locations indistinguishable. To address this issue, we consider both the visiting order and the travel time between two locations. Intuitively, users sharing the habit of "*cinema* → *restaurant* → *shopping*" are more similar with each other than users visiting these three places separately or in a different order. So, given two different users' location histories, we detect the shared sub-sequences at each layer of the $\mathcal{F}$. To guarantee the visiting order between two locations is meaningful, we require the travel time between the two locations to be similar.

Algorithm 1 MatchLocationHistory (H_1, H_2)

Input: Two users' semantic location histories H_1 and H_2
Output: A similarity score between the two users
Method
1: $MTS = \emptyset$;
2: **for** l from 1 to L //L is the number of layers of $\mathcal{F}$
3: MT^l = MaxTravelMatch($H_1.S^l, H_2.S^l$);
4: $MTS = MTS \cup MT^l$;
5: *sim*= CalculateSimilarityScore(MTS);
6: **Return** *sim*;

Fig. 7 The algorithm for location history matching

Definition 7 (*Sub-sequence*) Given a sequence $S = (c_1 \xrightarrow{\Delta t_1} c_2 \xrightarrow{\Delta t_2} \ldots \xrightarrow{\Delta t_{m-1}} c_m)$, we denote the ith item of S as $S[i]$ (e.g., $S[1] = c_1$) and represent its subsequence as $S[a_1, a_2, \ldots a_k]$ where $1 \le a_1 < a_2 < \ldots \le m$.

For instance, $S[1, 3, 6, 7] = c_1 \rightarrow c_3 \rightarrow c_6 \rightarrow c_7$ in the above definition. Note that, we allow *holes* in a sub-sequence, i.e., discontinuous, for a better sequence match.

Definition 8 (*Travel Match*) Given a temporal constraint factor $\rho \in [0, 1]$ and sub-sequences $S_1[a_1, a_2, \ldots a_k]$ and $S_2[b_1, b_2, \ldots b_k]$ from two sequences S_1 and S_2 respectively, these two sub-sequences formulate a k-length travel match if they hold the following two conditions.

1. $\forall i \in [1, k], \quad a_i = b_i$, and
2. $\forall i \in [2, k], \frac{|\alpha_i - \alpha_i'|}{Max(\alpha_i, \alpha_i')} \le \rho$, where $\alpha_i = S_1[a_i].t_a - S_1[a_{i-1}].t_l$ and $\alpha_i' = S_2[b_i].t_a - S_2[b_{i-1}].t_l$, i.e., the travel time between two locations.

In the latter of this paper, we represent the travel match as $(a_1, b_1) \rightarrow (a_2, b_2) \rightarrow \cdots \rightarrow (a_k, b_k)$.

Essentially, a travel match is a common sequence of semantic locations visited by two users in similar travel times. Note that the semantic locations in a travel match do not have to be consecutive in the user's original location history. For instance, one user went hiking from a lake (i.e., *lake* → *hiking park*). Another one had the similar route but

Table 1 Notations

Symbols	Descriptions
c	A semantic location
$S = (c_1 \rightarrow c_2 \rightarrow \ldots)$	A semantic location sequence
$\|S\|$	The length of S, i.e., number of nodes
$S[i]$	The ith item of S, e.g., $S[1] = c_1$
$S[i].t_a$, $S[i].t_l$	The arriving/leaving times at/from $S[i]$
$S[a_1, a_2, \ldots a_k]$	A subsequence of S.a_i is an index
$(a_1, b_1) \rightarrow (a_2, b_2) \ldots \rightarrow (a_k, b_k)$	A travel match between two sequences; a_i and b_i are indices of locations in the sequence and $S[a_i] = S'[b_i]$.
(a_i, b_i)	A 1-length travel match, or trivial match
G, G'	A precedent graph/a refined graph of G
P	A path in G or G'
$P_1 + P_2$	Concatenating P_1 with P_2 sequentially

stopped by a hotel for booking a room or having a lunch (i.e., *lake* → *hotel* → *hiking park*). In this case, "*lake* → *hiking park*" should still be considered as a common sequence (between the two users' location histories) as long as the gap between the two users' travel times from the lake to the hiking park is not very big. However, if the second user stayed in the hotel for a few days and accessed some other places before approaching the hiking park, we do not regard "*lake* → *hiking park*" as a common sequence any longer due to the big time difference.

Definition 9 (*Maximal Travel Match*) A travel match $(a_1, b_1) \rightarrow (a_2, b_2) \ldots \rightarrow (a_k, b_k)$ between two sequences S_1 and S_2 is a maximal travel match if,

1. No left increment: $\nexists\, a_0 < a_1, b_0 < b_1$, s.t., $(a_0, b_0) \rightarrow (a_1, b_1) \rightarrow (a_2, b_2) \cdots \rightarrow (a_k, b_k)$;
2. No right increment: $\nexists\, a_{k+1} > a_1, b_{k+1} > b_k$, s.t., $(a_1, b_1) \rightarrow (a_2, b_2) \cdots \rightarrow (a_k, b_k) \rightarrow (a_{k+1}, b_{k+1})$, and
3. No internal increment: $\forall i \in [1, k], \nexists\, a_i < a_{i'} < a_{i+1}$ and $b_i < b_{i'} < b_{i+1}$, s.t.,

$$(a_1, b_1) \rightarrow (a_2, b_2) \cdots \rightarrow (a_i, b_i) \rightarrow (a_{i'}, b_{i'}) \rightarrow (a_{i+1}, b_{i+1}) \rightarrow \cdots \rightarrow (a_k, b_k).$$

Example 2 (*Maximal Travel Match*) Figure 8 demonstrates an example of the maximal travel match between two sequences S_1 and S_2. Here, a node stands for a semantic location and the letter in a node represents the ID of the location. The numbers on the top of the box denotes the index of a node in a sequence, e.g., location A is the first node in both S_1 and S_2. The number appearing on a solid edge means the travel time between two consecutive nodes, and the number shown on a dashed edge denotes the stay time in a location.

Let $\rho = 0.2$ in this example. First, $(1, 1) \rightarrow (2, 2)$, i.e., $A \rightarrow B$, is a travel match, because the travel times $(A \rightarrow B)$ in S_1 and S_2 are identical, $|2 - 2|/2 = 0$. Then, we find that $(2, 2) \rightarrow (3, 4)$, i.e., $B \rightarrow C$, also satisfies the conditions defined in Definition 8. Though B and C is not directly connected in S_2, the travel time between these two locations is $4 + 0.5 + 0.5 = 5$, which is very similar to that of S_1. In short, $|5 - 4|/5 = 0.2$. However, both $A \rightarrow B$ and $B \rightarrow C$ are not the maximal travel match in this example as they are contained in $A \rightarrow B \rightarrow C$, i.e., $(1, 1) \rightarrow (2, 2) \rightarrow (3, 4)$.

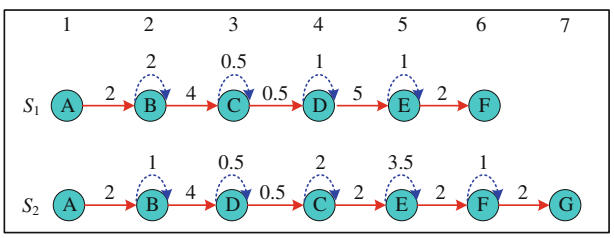


Fig. 8 An example of the maximal travel match

Later, $C \rightarrow E$ and $C \rightarrow F$ cannot formulate travel matches due to the difference between corresponding travel times. Using the same approach, we find $(1, 1) \rightarrow (2, 2) \rightarrow (4, 3) \rightarrow (5, 5) \rightarrow (6, 6)$, i.e., $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$, is another maximal travel match. Overall, we detect two maximal travel matches, $A \rightarrow B \rightarrow C$ and $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$ from S_1 and S_2.

4.2 Discovering maximal travel matches

Figure 9 presents the process of discovering the maximal travel matches. First, we detect the 1-length travel matches (coined as *trivial matches* in Definition 10) between two sequences and identify a precedence relation (refer to Definition 11) between these trivial matches. Then, the trivial matches and their precedence relation are transformed into a graph G, where a node is a trivial match and an edge corresponds to the precedence relation between trivial matches. Second, we prove that a maximal match is equivalent to a maximal length path in the graph G', which is a refined graph from G. Note that, for the sake of efficiency, we directly build G' instead of building G first and then removing redundant edges from G (see Sect. 4.2.1). Later, by searching for the maximal length path in the G', we can find out the maximal travel matches (refer to Sect. 4.2.2 for details).

Definition 10 (*Trivial Match*) A trivial match is a 1-length travel match, e.g., location A (1, 1) in Fig. 8.

Definition 11 (*Precedence Relation*) Let (i, j) and (i', j') be two trivial travel matches between S_1 and S_2. (i, j) is a precedence of (i', j') if:

1. $i < i'$ and $j < j'$, and
2. $\frac{|\alpha_i - \alpha'_i|}{Max(\alpha_i, \alpha'_i)} \leq \rho$, where $\alpha_i = S_1[i'].t_a - S_1[i].t_l$ and $\alpha'_i = S_2[j'].t_a - S_2[j].t_l$.

The precedence relation does not satisfy reflexivity, i.e., (i, j) is not a precedence of (i, j). In addition, the precedence relation does not have transitivity since the second condition may be violated. In Fig. 8, $A(1, 1)$ is a precedence of $B(2, 2)$ and $B(2, 2)$ is a precedence of $C(3, 4)$. However, $A(1, 1)$ is not a precedence of $D(4, 3)$ because the difference of the travel time from A to D in S_1 and S_2 is $9 - 7/9 > 0.2$.

Algorithm 2 MaxTravelMatch (S_1, S_2)

Input: Two semantic location sequences S_1 and S_2
Output: The set of maximal travel matches
Method
1: G'= BuildGraph(S_1, S_2);
2: **Return** $\bigcup_{u \in Zeroln}(G')\, PartialMax(G', u)$;

Fig. 9 The process of finding the maximal travel matches

4.2.1 Building the precedence graph

Following the case demonstrated in Figs. 8, 10 depicts an example of building the precedence graph $G = (V, E)$ based on the trivial matches and corresponding precedent relations detected from S_1 and S_2. Basically, each node in V is a trivial match between S_1 and S_2, and an edge in E from node (i, j) to (i', j') stands for a precedent relation between the two trivial matches. The Algorithm 3 shown in Figs. 11 describes the process in detail.

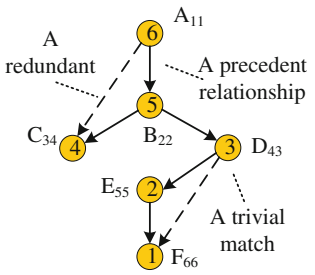
Using Fig. 10 as an example, we illustrate Algorithm 3. In Fig. 10b, each node corresponds to a trivial match, and the number in a node indicates its order in the sorted list, i.e., F_{66}, E_{55}, D_{43}, C_{34}, B_{22}, and A_{11} (see line 2 of Algorithm 3).

We first mark all nodes to white, which means the node is unreachable from the existing edges in G'. Then, the main loop (from Line 5 to 10) adds non-redundant edges starting from each node v_l into G'. If v_l has a precedence relation with v_t, we can formulate a candidate edge $e = v_l \rightarrow v_t$. Here, e is non-redundant only if v_t is marked white, i.e., unreachable from another path starting from v_l. After adding a non-redundant edge e, we mark all reachable nodes from e to black.

In Fig. 10b, the main loop starts from E_{55}. According to Definition 11 E_{55} is a precedence of F_{66}. As F_{66} is labeled as white so far, we add $E_{55} \rightarrow F_{66}$ to G' and then mark F_{66} to black. After that, the main loop comes to D_{43}. In the sorted list, the nodes before D_{43} are E_{55} andF_{66}. Since D_{43} is a precedence of E_{55} and E_{55} is white, we add $D_{43} \rightarrow E_{55}$ into $G'$and then mark E_{55} to black. Though D_{43} is also a precedence of F_{66}, F_{66} is already marked as black. Thus, the edge $D_{43} \rightarrow F_{55}$ is redundant and cannot be inserted into G'. Now, the main loop for D_{43} is over.

Index	1	2	3	4	5	6
	A	B	C	D	E	F
1 A	1	0	0	0	0	0
2 B	0	1	0	0	0	0
3 D	0	0	0	1	0	0
4 C	0	0	1	0	0	0
5 E	0	0	0	0	1	0
6 F	0	0	0	0	0	1
7 G	0	0	0	0	0	0

(a) The match matrix



(b) The precedent graph

Fig. 10 The precedence graph for S_1 and S_2

Algorithm 3 BuildGraph (S_1, S_2)

Input: Two semantic location sequences S_1 and S_2
Output: A directed acyclic graph G'.
1: **For** $\forall i \in [1, |S_1|], \forall j \in [1, |S_2|]$
2: **If** $S_1[i].c = S_2[j].c$
3: Add the node (i, j) into a list Ψ;
4: $\Psi \leftarrow$ Sort (Ψ); //sort in a decreasing lexicographical order.
//Suppose $\Psi = (v_1 = (i_1, j_1), \ldots, v_k = (i_k, j_k))$.
5: **For** l from 2 to k
6: Mark all nodes white
7: For t from $l - 1$ down to 1
8: **if** v_t is white
9: **if** v_l is a precedence of v_t
10: Build an edge $v_l \rightarrow v_t$ in G'.
11: Mark all nodes reachable from v_t black
12: **Return** G';

Fig. 11 Building refined graph directly based on two sequences

Lemma 1 A precedence graph G is a directed acyclic graph. Each travel match corresponds to a path in G.

More specifically, if $(a_1, b_1) \rightarrow (a_2, b_2) \ldots \rightarrow (a_k, b_k)$ is a path in G, $S_1[a_1, a_2, \ldots a_k]$ and $S_2[b_1, b_2, \ldots b_k]$ form a travel match, and vice versa. For instance, the path $A_{11} \rightarrow B_{22} \rightarrow C_{34}$ in Figure 10b corresponds to the travel match (1, 1) → (2, 2) → (3, 4) in Fig. 8.

Definition 12 (*Maximal Length Path*) A path P in G is a maximal length path if the first node of P has zero in-degree and the last node has zero out-degree.

For example, in Fig. 10b, the path $A_{11} \rightarrow B_{22} \rightarrow C_{34}$ is a maximal path, which corresponds to a maximal travel match. However, the maximal path $A_{11} \rightarrow C_{34}$ does not correspond to a maximal match.

To address this issue, we refine G into a new graph G' by removing the redundant edges. Let $Reach_G(u)$ be the set of nodes reachable from u in G. We define an edge $u \rightarrow v$ as redundant in G, if $Reach_G(u) = Reach_{G/\{u \rightarrow v\}}(u)$, where $G/\{u \rightarrow v\}$ $(V, E/\{u \rightarrow v\})$. In other words, an edge $u \rightarrow v$ in G is redundant if there is another alternative path from u to v. For example, $A_{11} \rightarrow C_{34}$ in Fig. 10b is redundant.

Lemma 2 The following two statements are equivalent:

1. $S_1[a_1, a_2, \ldots a_k]$ and $S_2[b_1, b_2, \ldots b_k]$ form a maximal travel match M between S_1 and S_2.
2. $P = (a_1, b_1) \rightarrow (a_2, b_2) \rightarrow \ldots \rightarrow (a_k, b_k)$ is a maximal path in G', i.e., (a_1, b_1)has zero in-degree and (a_k, b_k) has zero out-degree in G'.

Proof (1) ⇒ (2). According to Definition 9, a maximal travel match cannot be extended any longer on both left and right sides. So, the path corresponding to the maximal travel match in G' does not have any precedent nodes and successors. That is (a_1, b_1) has zero in-degree and (a_k, b_k) has zero out-degree in G'. In short, M is a maximal path in G'.

Proof (2) ⇒ (1). First, as (a_1, b_1) has zero in-degree in G', the condition 1 of Definition 9 holds. Second, (a_k, b_k) has zero out-degree. Therefore, condition 2 of Definition 9 also holds. Third, since we cannot find any redundant edges in the refined graph G', the condition 3 holds. That is, given any two consecutive nodes in P, we cannot find other paths passing these two nodes except for P.

Lemma 2 enables us to find maximal matches by exploring maximal paths in G'. However, obtaining G'

from G is quite time consuming. Consequently, in the implementation, we directly construct G' by using Algorithm 3 instead of first building G and then removing redundant edges from G. In Line 2, (i, j) is before (i', j') in the decreasing lexicographical order if $i > i'$, or $i = i' \wedge j > j'$.

Lemma 3 Algorithm 3 outputs the correct graph G' for two semantic location sequences S_1 and S_2.

Proof: Let $(v_1 = (i_1, j_1), \ldots, v_k = (i_k, j_k))$ be the sequence in Line 2 of Algorithm 3. Let G be the graph that contains all precedence relationships. Clearly, the edge set of G' outputted by Algorithm 3 is a subset of that of G.

First, we show that each non-redundant edge of G is in G'. Let $e = v_l \rightarrow v_t$ be a non-redundant edge in G. If e is not an edge in G', it must be the case that v_t is black (see Line 7 of Algorithm 3). However, v_t is only marked black when there is another node v'_t so that v_t is reachable from v'_t, and $v_l \rightarrow v'_t$ is an edge in G'. Then, there exists another path from v_l to v_t in G' as well as in G. This contradicts the assumption that e is non-redundant.

Second, we show that each edge of G' is non-redundant in G. Let $e = v_l \rightarrow v_t$ be an edge in G', and assume on the contrary e is redundant in G. Since removing redundant edges does not change reachability, there must exist another path P of non-redundant edges from v_l to v_t in G. According to the first argument in the previous paragraph, G' also contains this path. Let P be $(v_l \rightarrow v_{l_1} \rightarrow, \ldots, \rightarrow v_{l_k} \rightarrow v_t)$. For each node in P, let us consider the main loop processing node v_l. When building edge $v_l \rightarrow v_{l_1}$ in Line 8 of Algorithm 3, all other edges in P have been built. Therefore, v_t is already marked black since it is reachable from v_{l_1}. As a result, the algorithm will not build $v_l \rightarrow v_t$. It is a contradiction to the assumption. So, each edge of G' is non-redundant in G.

4.2.2 Finding maximal paths

This subsection describes how to output all maximal travel matches given a refined graph G', i.e., Line 2 in Algorithm 2 Here, we refer to a path from a node u to a zero out-degree node in G' as a *partial maximal path* from u. As shown in Fig. 12, Algorithm 4 generates all partial maximal paths from a given node u in G'. Let $No_{G'}(u)$ be the set of nodes that u has outgoing edges to. For two sets of paths $\mathcal{P}1$ and $\mathcal{P}2$ in G', if the last node of any path $P_1 \in \mathcal{P}_1$ is the same with the first node of any path $P_2 \in \mathcal{P}_2$, $\mathcal{P}_1 \otimes \mathcal{P}_2 = \{P_1 + P_2 | P_1 \in \mathcal{P}_1, P_2 \in \mathcal{P}_2\}$, where $P_1 + P_2$ the path simply concatenating P_1 and P_2.

For completeness, we state the complete algorithm shown in Algorithm 2. Let $ZeroIn(G')$ be the set of nodes with zero in-degree in G'. Each path in the set returned by Algorithm 2 corresponds to a maximal match. For example, the maximal matches between S_1 and S_2 in Fig. 8 are $A \rightarrow B \rightarrow C$ and $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$.

Algorithm 4 PartialMax $(G',\ u)$

Input: A directed acyclic graph G', and a node u.
Output: The set of partial maximal paths in G'.
Method
1: **Return** $\bigcup_{v \in No_{G'}(u)} (u \rightarrow v) \otimes PartialMax(G', v)$;

Fig. 12 Output partial maximal paths

4.2.3 Calculating similarity using maximal matches

We consider three factors when measuring the similarity between two users in terms of their location histories.

1. *Sequential property*: Users sharing longer semantic location sequences would be more similar. We detect the common sequences between two users' location history by using the maximal travel match algorithm and give a higher weight to a longer match.
2. *The granularity of a semantic location*: Users sharing semantic locations with a finer granularity could be more similar. We give a relatively higher weight to the maximal travel matches detected at a lower layer of the hierarchy $\mathcal{F}$.
3. *The popularity of a semantic location*: Users sharing semantic locations that are less frequently visited by people would be more similar. Here, we propose inverted user frequency to give an unpopular location a high importance score: $iuf(c) = \log \frac{N}{n}$, where N is the total number of users in the dataset and n is the number of users visiting the semantic location c.

Combining the three factors, we calculate an overall similarity score for each pair of users in terms of the Eq. 4.

$$SimUser(H_1, H_2) = \sum_{l=1}^{L} f_w(l) \times SimSq(S_1^l, S_2^l); \tag{4}$$

$$SimSq(S_1, S_2) = \frac{\sum_{j=1}^{m} sg(t_j)}{|S_1| \times |S_2|}, \tag{5}$$

$$sg(s) = g_w(k) \times \sum_{i=1}^{k} iuf(c_i). \tag{6}$$

Given two users' location histories H_1 and H_2, we compute the similarity between them by summarizing the weighted similarity of semantic location sequences detected at each layer of the hierarchy $\mathcal{F}$. We use a function $f_w(l)$ to assign a higher weight to the similarity of sequences occurring at a

lower layer, e.g., $f_w(l) = 2^{l-1}$. Then, the similarity between two semantic location sequences S_1 and S_2 at a layer, $SimSq(S_1, S_2)$, is represented by the sum of the similarity score, $sg(t_j)$, of each maximal match between S_1 and S_2. Here, m is the total number of maximal matches. Meanwhile, $SimSq(S_1, S_2)$ is normalized by the production of the lengths of the two sequences, since a longer sequence have a high probability to have long matches. That is, a user with a longer history are more likely to be similar to others (than a user having a short period of history) without performing the normalization. Later, we calculate the similarity of a maximal travel match t, $sg(t)$, by summing up the *iuf* of each semantic location c in t and weighting $sg(t)$ in terms of the length k of t, e.g., $g_w(k) = 2^{k-1}$.

5 Finding similar users

With the user similarity calculated above, we can hierarchically cluster users into some groups in a divisive manner by using some clustering algorithms, like K-mean. As a result, as depicted in Fig. 13, we can build a user cluster hierarchy, where a cluster denotes a group of users sharing some similar interests and different layers represent different levels of similarity (Hung et al. 2009). The clusters shown on a higher layer could stand for big communities in which people share some high-level interests, such as sports. The clusters that occur on the lower layers denote people sharing some finer interests, like hiking (the layer of the hierarchy can be determined based on the needs of applications). Meanwhile, we can find out one representative user (the center) of each cluster according to the similarity scores between each pair of users.

This user hierarchy brings us two aspects of advantages.

1. Fast retrieval of similar users: Instead of checking all the users, we can retrieve top k similar users for a person by only ranking the users in the same cluster as the person (in terms of the similarity score). This retrieval process can start from the bottom layer of the hierarchy, as depicted by the blue dash arrow. Finding more than k users in the bottom-layer cluster that the person pertains to, we can directly rank these users in the cluster for the person. If the number of users is less than k, we can further check the parent node (cluster) of this cluster until finding out a cluster with more than k users.
2. Insert new users: When a new user u' comes to the system, it is not necessary to compute the similarity score between u' and each user in the system. This process is very time consuming and will keep on increasing with the number of users. Instead, we only need to insert this user into the most proper clusters on each layer of the hierarchy by computing the similarity between u' and the representative user in a cluster. For example, as demonstrated by the red solid arrows in Fig. 13, we first compute the similarity between u' and $(u_1, u_2, \ldots, u_k)$ If u_2 is the most similar user to u' out of the k users, we insert u' into u_2's cluster C_2. Then, we further check the child clusters of C_2 and insert u' into the clusters whose representative user is the most similar to u'. This process is performed iteratively until reaching the bottom layer of the hierarchy.

In practice, we do not need to re-build this hierarchy unless the numbers of newly inserted users exceed a certain threshold. That is, in most case we can find similar users for a person very efficiently.

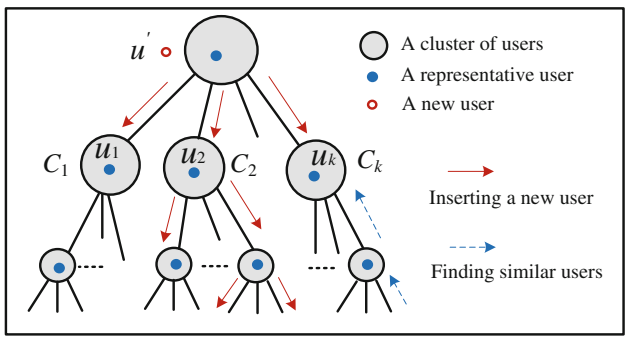


Fig. 13 Finding similar users and inserting new users in a hierarchical user clusters

6 Evaluation

In this section, we evaluate our method based on a real-world GPS trajectory dataset collected by 109 users in a period of over 1 year.

6.1 Settings

6.1.1 GPS devices, users, and trajectories

As shown in Fig. 14, our GPS devices include Magellan Explorist 210/300, G-Rays 2 and QSTARZ and GPS-enabled phones. These devices are configured to record a

Fig. 14 GPS-enabled devices used for the user study

GPS reading every 5 s, and are delivered to 109 users with diverse backgrounds, including college students, housewives, employees of different companies and organizations, etc. We collected GPS logs of these volunteers in a period of 1 year.

As a result, the collected GPS trajectories cover 36 cities and include totally 8,027,911 GPS points, from which we detected 18,074 stay points. We used 5,366 stay points in weekends as our data set to make sure that these stay points reflect users' interests in leisure time instead of routine paths between homes and offices.

6.1.2 Parameter selections

The default values of parameters in location history construction are as follows. (1) Stay point detection: we test a set of thresholds and find out $\theta_d = 200$ m and $\theta_t = 30$ min is more proper than others (refer to papers (Li et al. 2008; Zheng et al. 2011d) for more justifications). These values allow us to detect meaningful places that users visited and filter places where users only passed by. Also, these two parameters are relatively robust to traffic jams. (2) Stay point extension: we test the performance of our method changing over γ and set $\gamma = 200$ m after the study. (3) Feature vector construction: our POI database contains 6,828,951 POIs of 13 categories including restaurants, markets, natural parks, and museums, etc. (4) Feature vector clustering: we use k-means to hierarchically cluster feature vectors into three layers in a divisive manner. As a result, the second layer contains 15 clusters and the third layer contains 48 clusters.

The default values of parameters in user similarity exploration are as follows. We test a set of ρ, and set $\rho = 0.2$ to find maximal matches (refer to Fig. 24 for details). We use $g_w(k) = 2^{k-1}$ to give a larger weight to longer match. As shown in Fig. 15, we observe that the occurrence of k-length travel matches drops exponentially as the k increases. Thus, the significance of an occurrence of a k-length travel match increases exponentially with k. Similarly, we set $f_w(l) = 2^{l-1}$ to give a larger weight to lower layers because the number of maximal matches of 132, 056 at the second layer is much larger than 48, 252 at the third layer.

6.2 The evaluation approach

6.2.1 Ground truth

To obtain the ground truth of a user's interests, we conduct a questionnaire-style user study, in which each user answers the questions we proposed by giving a rank (1–4), as shown in Fig. 16a. Then, we regard a user's answer, e.g., Figure 16b, as an interest vector, in which each entry is the user's rank to a corresponding question. Later, we calculate a cosine similarity between two users' interest vectors. A user's true ranking list of ten most similar users is those having the closest interest vectors with her.

6.2.2 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 ≥ 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 (Jarvelin and Kekalainen 2002). The discounted cumulative gain of G is computed as follows: (In our experiments, $b = 2$.)

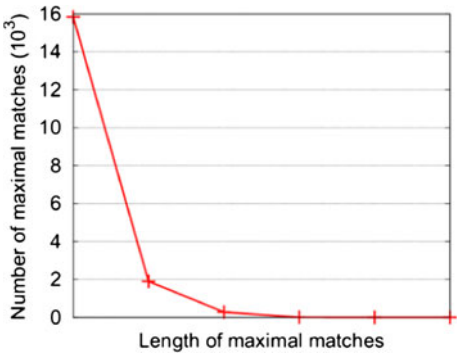


Fig. 15 The distribution of the maximal travel matches

Where do you like to go in weekends? Please rank from 1(dislike) to 4(favorite).	Example response
1. Shopping	3
2. Theatre	2
3. Karaoke	1
4. Go out for dinner	4
5. Outdoor sports, e.g., hiking	3
6. Indoor sports, e.g., gym and bowling	1
7. Natural parks	3
8. Exhibition, museum	1
9. Stay home; not go to any places	2
10. Go to office; over-time working	3
11. Visit parents, relatives, or friends	1
12. Campus	1
(a)	**(b)**

Fig. 16 A questionnaire (**a**) and an example of answers (**b**)

$$DCG[i] = casesG[1], ifi = 1DCG[i-1] + G[i], ifi < bDCG[i-1] + \frac{G[i]}{\log_b i}, ifi \geq bcases \quad (7)$$

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

6.2.3 Evaluation framework

The experiment aims to validate the four contributions we claimed: (1) We propose to use semantic location history instead of physical positions to represent users' interests. (2) We build a semantic location history with multiple granularities for each user so as to measure two users' similarity more precisely. (3) We propose the maximal travel match to compare the semantic location sequences of two users. The maximal match has a better performance beyond existing sequence matching approaches in this application scenario. (4) We propose *iuf* to give high weights to unpopular semantic locations. We evaluate the effectiveness of the four points as follows.

1. We compare the effectiveness of the semantic location history with that of a physical-location-based approach HGSM (Li et al. 2008) proposed by us previously. In addition, to evaluate the effectiveness of the feature vector used to represent the semantic meaning of a stay region, we also compared our method with two basic semantic-based approaches. One basic approach is called *NearestType*, which assigns the category of the nearest POI to a stay point. The other approach *LargestType* assigns a stay region the category having the largest number of POIs within the stay region. Since these two approaches cannot produce hierarchical location histories, we only used the second layer of our SLH in the comparison. The rest of settings of the two baselines are the same with that of SLH-MTM.
2. To validate the effectiveness of the multiple granularities of semantic locations, we compared the performance of our approach using multiple layers of the hierarchy $\mathcal{F}$ with that only using the second layer or only using the third layer.
3. To evaluate the effectiveness of MTM, we first compared the MTM with three approaches that do not consider the sequential property: *Count*, *Cosine*, and *Pearson*. Suppose N semantic locations $\{c_i, 1 \leq i \leq N\}$ are generated on a certain layer of the hierarchy $\mathcal{F}$. If in 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.

$$u_1 = (k_1, k_2, \ldots k_i, \ldots, k_N), \quad u_2 = (l_1, l_2, \ldots, l_i, \ldots, l_N).$$

The similarity of two users by *count* is computed as Eq. (8):

$$sim_{count}(u_1, u_2) = \sum_{i=0}^{N} \min(k_i, l_i) \quad (8)$$

For instance, the similarity between two users' location histories shown in Fig. 8 is 6 (*A*, *B*, *C*, *D*, *E*, *F*).When conducting the Cosine and Pearson methods, we compute the similarity between two users' location histories according to Eqs. (9) and (10) respectively:

$$sim_{\cos ine}(u_1, u_2) = \frac{\sum_i k_i l_i}{\sqrt{\sum_i l_i^2}\sqrt{\sum_i k_i^2}} \quad (9)$$

$$sim_{pearson}(u_1, u_2) = \frac{\sum_i (k_i - \bar{u}_1)(l_i - \bar{u}_2)}{\sqrt{\sum_i (k_i - \bar{u}_1)^2 \sum_i (l_i - \bar{u}_2)^2}} \quad (10)$$

When performing these three baselines, we substitute the *SimSq* in Eq. 4 with corresponding similarity shown in Eqs. 8, 9, and 10, and keep the remaining settings the same as our SLH-MTM.We also compared MTM with four existing sequence matching approaches that consider the visiting order of locations but without taking into account the travel time constraints between locations. These approaches include τ-containment for trajectory pattern mining (Giannotti et al. 2007) and three well-known sequence matching approaches, consisting of ED, LCSS (Vlachos et al. 2002) and DTW (Yi et al. 1998).
4. To evaluate the effectiveness of *iuf*, we compared *iuf* with the baseline using the same weights for all semantic locations, denoted as Same Weight.
5. In addition to the four aspects, we examined the effects of the scale of stay regions (γ) and the difference ratio on travel time (ρ) on the performance of our approach.

6.3 Results

6.3.1 Effectiveness of SLH

Figure 17 shows that our approach has higher *nDCG* scores beyond HGSM. This justifies the advantage of semantic locations over geographic positions. The reason is that HGSM are relatively weak in detecting the similarity between users without overlaps in the geographic spaces. Consider two users visiting "*museum 1* → *shop 1*" and "*museum 2* → *shop 2*", respectively. Our approach regards them as similar since they share a semantic sequence

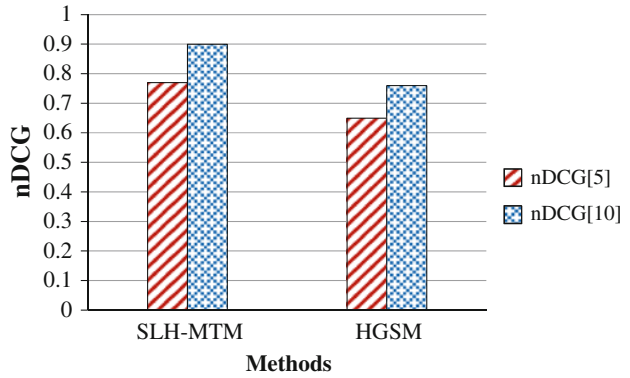


Fig. 17 The semantic locations versus physical locations

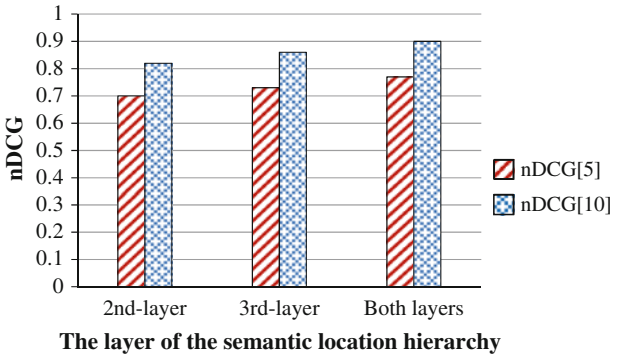


Fig. 19 The effectiveness of the semantic granularity

of"*museum* → *shop*", while HGSM could not handle this case.

Figure 18 shows that our approach obtains higher *nDCG* than two basic semantic-based approaches: *NearestType* and *LargestType*. It suggests that our approach using a feature vector to represent a stay region is more capable of modelling a user's interests beyond these two baselines. The reason is that our approach is aware of the GPS positioning errors and captures the uncertainty of location categories by feature vectors.

6.3.2 *Effectiveness of multiple granularities*

As shown in Figure 19, the finer granularity (using the third layer) achieves higher *nDCG* than the coarse granularity (using the second layer, i.e., only cluster the feature vectors once). Moreover, our approach achieves the highest *nDCG* when using multiple granularities and assigning large weights to a fine granularity. This result indicates that a finer granularity of semantic locations improves SLH's capability in discriminating similar users (optimizing for precision), while a coarser one enhances SLH to find coarsely similar users (optimizing for recall). By combining the capabilities of multiple granularities, SLH is able to distinguish users more precisely and detect users similar at different degrees. For example, when combining multiple granularities, a coarse granularity discover the group of similar persons who like outdoor sports, and a fine granularity further distinguishes users who like hiking from these users.

6.3.3 *Effectiveness of MTM*

We first validate the effectiveness of the sequential property. As shown in Fig. 20, MTM achieves higher *nDCGs* than the three approaches only considering individual semantic locations shared by two users. This means a sequence of semantic locations is more capable of representing a user's interests than considering locations individually. Actually, common semantic locations are some 1-length travel matches. We can obtain the same user similarity by only using 1-length travel matches in our method. Naturally, we are able to further differentiate users by considering longer matches. For example, a couple visiting together three semantic locations in a sequence "*museum* → *restaurant* → *shopping mall*" are more similar than a user visiting these locations separately, or in a different order "*shopping mall* → *museum* → *restaurant*".

Then, we compare MTM with other sequence matching approaches. Figure 21 shows that MTM outperforms three

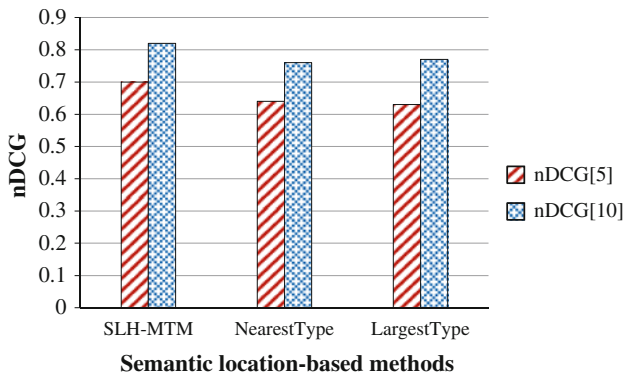


Fig. 18 SLH versus two baseline semantic-based methods

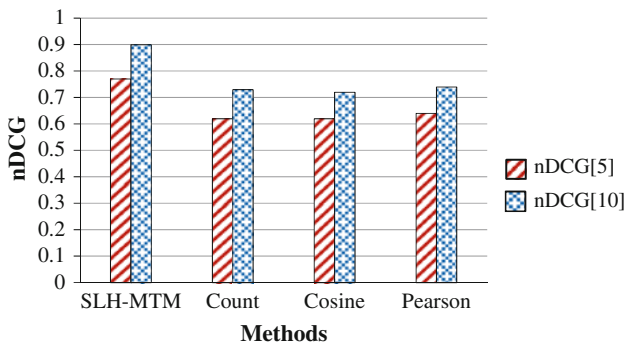


Fig. 20 MTM versus methods regardless of sequential properties

widely used sequence matching approaches ED, LCSS, and DTW in this application scenario. By taking into account the travel time between two locations, MTM is more capable of modeling the sequential property of a user's outdoor movement, and the behavior and intension behind the movement. For instance, two users share a sequence of"*restaurant* → *shopping mall*" in their location histories. One user went to the two places in different trips occurring in different days. The other visited the two places in one trip (in the same day). MTM regards the two sequences as different, while the three sequence matching approaches mistakenly consider them as a common sequence and use it to measure user similarity. Intuitively, different time interval denotes different extents of sequentiality.

Figure 21 also shows that MTM outperforms the traditional trajectory pattern mining approach τ-containment, which aims to mine the sequential patterns with travel time constraints from given trajectories. This method enables us to detect some frequent sequential patterns from a user's location history, while ignoring some common sequences (shared by two users) that occur infrequently but carry important meanings of a user's interests. For instance, two user shares a sequence of "*lake* → *hiking park*", which is not a frequent pattern in these two users' location histories as these types of events are relatively rare in people's daily life (compared to shopping and dining). Though not frequent, such kinds of sequences are still important in reflecting a person's interests. Sometimes, these infrequent sequences are even more valuable in differentiating people than frequent patterns. Besides, τ-containment is much less efficient than MTM. τ-containment used 292 s to mine patterns from our data set. In comparison, MTM only ran 19 s.

6.3.4 *Effectiveness of iuf*

As shown in Fig. 22, according to *nDCG*, our approach using *iuf* outperforms the method using the same weight for different semantic locations. This indicates that the scheme

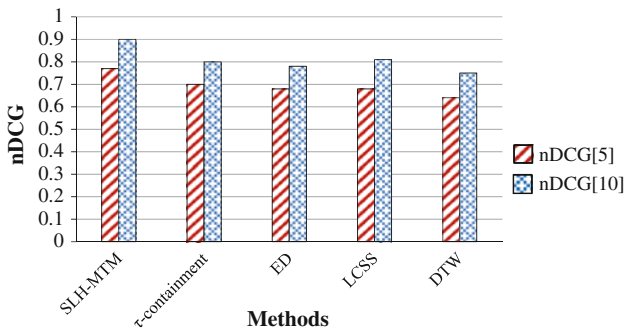


Fig. 21 MTM versus other sequence matching algorithms

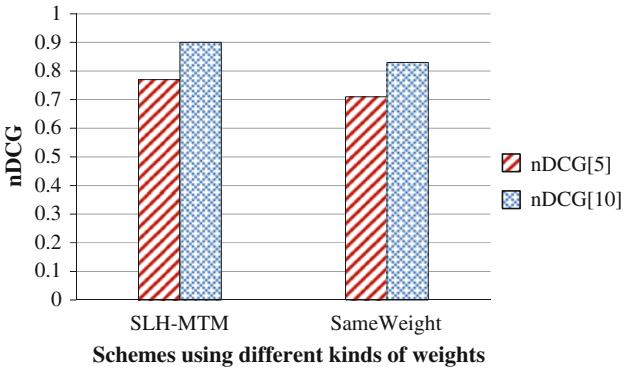


Fig. 22 *iuf* versus the method using the same weight for locations

assigning larger weights to less popular semantic locations is more capable of measuring the similarity between users. Intuitively, the semantic location of *restaurant* could appear in all users' location histories. Obviously, it is hard to say all of them are similar given this observation. However, the semantic locations, like *hiking park* and *lake*, which do not frequently occur in a user's location history, can reflect the user's real interests and are much more distinguishing than a common location like *restaurant*. Without *iuf*, the similarity between users will be dominated by those common semantic locations, and cannot reveal the true correlation between users.

6.3.5 *Effects of γ and ρ*

Figure 23 shows that when the scale of stay regions is 200 m, the *nDCG* is the highest in our data set. It is related to the positioning errors of our devices. When the region is too small, it may exclude the real places that users visited. When the region is too large, it may include many noisy POIs.

Figure 24 shows that we achieve the highest *nDCG* when ρ is round 0.2–0.4. When ρ drops below 0.2, it becomes too strict to find long matches. When ρ becomes too loose, some matches are meaningless, which cause the same problem as ED, LCSS, and DTW.

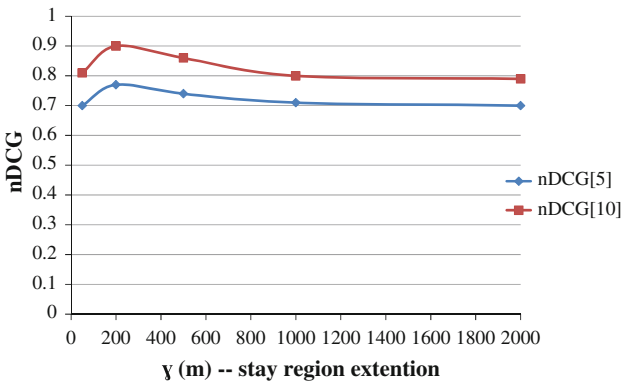


Fig. 23 Effect of scale of stay regions γ

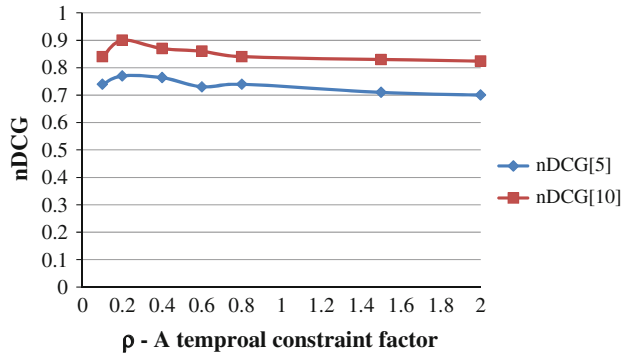


Fig. 24 Effect of difference ratio ρ on travel time

7 Related work

7.1 Mining human location history

A branch of research has been performed based on individual location history recorded in GPS trajectories. These researches include detecting significant locations of a user (Ashbrook and Starner 2003; Hariharn and Toyama 2004; Liu et al. 2006; Cao et al. 2010), predicting the user's movement among these locations (Ye et al. 2009), and recognizing user-specific activities at each location (Patterson et al. 2003). As opposed to these works, we aim to model multiple users' location histories and learn patterns from numerous individuals' behaviors.

Giannotti et al. (2007) mined similar sequences from users' trajectories, and MSMLS (Krumm and Horvitz 2007) used a history of a driver's destinations, along with data about driving behavior extracted from multiple users' GPS traces, to predict where a driver's may be going as a trip progresses. Zheng et al. (2008a, b, 2010b) classified the transportation modes of a GPS trajectory into driving, walking, taking a bus, and riding a bike. Instead of exploring users' behaviors, in this paper, we aim to understand the correlations between different users' activities and mine the similarity between users.

7.2 User similarity

7.2.1 In cyber systems

User similarity has been studied in some recommender systems and online social networks. The goal is to discover content of interest of a user from her similar users on the Web. One example is Amazon's book recommendation system (Linden et al. 2003), which recommends a user with some books she would like to read while have not been found by her. Other examples are LinkedIn and Facebook, where the user similarity can help a person find out some users sharing similar interests and backgrounds. Two widely used techniques are collaborative filtering (CF) (Breese et al. 1998) and nearest neighbor (NN) search (Sarwar et al. 2000). CF assumes that people agreed in the past tend to agree in the future. NN regards users with similar interests as nearest neighbors with highly correlated preference data.

Our work is different from these techniques in two aspects. First, we study user behavior in the physical world instead of online behavior. The learned user similarity can bridge the gap between the virtual world and physical world. Second, we consider some properties of human location histories, such as the sequence and the granularity of semantic locations, when measuring the user similarity.

7.2.2 In the physical world

Eagle et al. (2006) aimed to recognize the social patterns in users' daily activities from the dataset collected by users with Bluetooth-enabled mobile phones, and extract the social structures from the mobile phone data provided by wireless communication operators (Eagle et al. 2009a, b). Cranshaw et al. (2010) examines the location traces of 489 users of a location sharing social network for relationships between the users' mobility patterns and structural properties of their underlying social network. Hung et al. (2009) targeted at the problem of discovering communities among users, where users in the same community have similar trajectory patterns. Our work differs from the above-mentioned work in the following two aspects. First, we measure user similarity based on the semantic meanings of a location instead of physical positions. Second, we consider the sequential property between locations, and the hierarchy and popularity of a semantic location.

Li et al. (2008) proposed HGSM, which mine the similarity between users in terms of their physical location histories. When calculating the similarity, HGSM also considers a user's travel behaviors and the properties of geographical spaces. Further, Zheng et al. (2011d) incorporate this user similarity into a user-centric CF model to conduct a personalized friend and location recommendation. Though HGSM is very similar to the work reported in this paper, the major difference still lies in the semantic location we inferred from a given physical stay region. According to the statement in this paper, modeling the semantic meaning of a physical location is nontrivial. Moreover, in terms of the evaluation, the proposed MTM is more effective and efficient than the sequence matching algorithm used in HGSM. Finally, this article is an expansion of the poster paper (Xiao et al. 2010) with more details of methodology and experiments presented.

7.3 Location recommendation

Zheng et al. (2009c), recommended a user with the top interesting locations and travel sequences mined from a

large number of user-generated GPS trajectories. The experienced users in a given region are also recommended. Zheng et al. (2010a) use a collaborative learning approach to enable an activity-location recommendation based on GPS traces associated with user-generated comments. That is, given an activity recommend the best k locations, and given a location recommend the best k activities. Chen et al. (2010) recommended a user with some trajectories according to a set of user-specified point locations. As these recommendations are generic recommendation, they do not consider the similarity between users.

Froehlich et al. (2006) proposed to vote a single user's personal location history to recommend locations. City-Voyager (Takeuchi and Sugimoto 2006) recommended shops to users based on their past location histories. Zheng et al. (2009b) first learned the correlation between locations in terms of multiple users' location histories. In turn, the location correlation is employed by an item-based CF model to conduct a personalized location recommender (Zheng et al. 2011c). Extended from paper (Zheng et al. 2010a), a user-centric location-activity recommender is further performed (Zheng et al. 2010d). Although implicitly involving the user similarity (based on location history), these recommenders still use traditional CF techniques without considering specific properties of location histories, e.g., the sequence of users' movements and hierarchy of geographic locations.

8 Conclusion and future work

In this paper, instead of using online social structures, we estimate the similarity between users in terms of their location histories in the physical world. This similarity can lead to social ties between users in a social networking service, hence bridging the gap between online social networks and the physical world. Rather than directly matching different users' location histories in the geographic spaces, we model a user's GPS trajectories with a semantic location history (SLH). The SLH carries more semantic meanings of a user's interest (beyond physical location), and can find out similar users without geospatial overlaps. Also, we believe that users sharing (1) a finer semantic location, (2) a longer sequence of locations and (3) less popular semantic locations would be more similar to each other. Then, we compare different users' SLHs by using the maximum travel match (MTM), which considers both the sequence information and the travel time between locations. The evaluation results based on real-world GPS data show that SLH shows clear advantages over a physical-location-based approach and the basic semantic approaches. When simultaneously incorporating the three factors mentioned above, our approach achieves the best performance. Additionally, MTM is more effective than several widely used sequence matching approaches, such as LCSS and dynamic time wrapping DTW, in this application scenario.

In the future, we aim to compare this user similarity with users' relationship in online social networks. Second, we plan to further improve the efficiency of our approach and employ this similarity to conduct a personalized location recommendation system.

References

Ashbrook D, Starner T (2003) Using GPS to learn significant locations and predict movement across multiple users. Pers Ubiquitous Comput 7(5):275–286

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering, In: Proceedings of the international 14th conference on uncertainty in artificial intelligence, Madison, Wisconsin, USA, July 1998

Cao X, Cong G, Jensen CS (2010) Mining significant semantic locations from GPS Data. In: Proceedings of the VLDB Endowment

Chen Z, Shen H T, Zhou X, Zheng Y, Xie X (2010) Searching trajectories by locations—an efficiency study. In: Proceedings of ACM SIGMOD International Conference on Management of Data. ACM Press, New York, pp 255–266

Cranshaw J, Toch E, Hong J, Kittur A, Sadeh N (2010) Bridging the gap between the physical location and online social networks. In: Proceedings of the International Conference on Ubiquitous Computing. ACM Press, New York

Eagle N, Pentland A (2006) Reality mining: sensing complex social systems. Pers Ubiquitous Comput 10(4):255–268

Eagle N, Pentland A, Lazer D (2009a) Inferring social network structure using mobile phone data. Proc Nat Acad Sci (PNAS) 106(36):15274–15278

Eagle N, Montjoye Y-A, Bettencourt LMA (2009b) Community computing: comparisons between rural and urban societies using mobile phone data. IEEE Soc Comput 144–150

Froehlich J, Chen M, Smith I, Potter F (2006) Voting with your feet: an investigative study of the relationship between place visit behavior and preference. In: Proceedings of the international conference on ubiquitous computing. ACM Press, New York

Giannotti F, Nanni M, Pedreschi D, Pinelli F (2007) Trajectory pattern mining. In: Proceedings of the 13rd ACM SIGKDD conference on knowledge discovery and data mining, San Jose, CA, USA, August 2007. ACM Press, New York, pp 330–339

Hariharn R, Toyama K (2004) Project Lachesis: parsing and modeling location histories. In: Proceedings of the 3rd international conference on geographic information science, Park, Utah, October 2004, pp 106–124

Hung CC, Chang CW, Peng WC (2009) Mining trajectory profiles for discovering user communities. In: Proceedings of ACM SIGSPATIAL GIS workshop on location based social networks

Jarvelin K, Kekalainen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 22(1):422–446

Krumm J, Horvitz E (2007) Predestination: where do you want to go today? IEEE Comput Mag 40(4):105–107

Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10:707–710

Li Q, Zheng Y, Xie X, Chen Y, Liu W, Ma WY (2008) Mining user similarity based on location history. In: Proceeding of the 16th

international conference on advances in geographic information system. ACM Press, New York, pp 1–10
Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
Liu G, Wolfson O, Yin H (2006) Extracting semantic location from outdoor positioning systems. In: Proceedings of the 7th international conference on mobile data management. IEEE, p 73
Patterson D, Liao L, Fox D, Kautz H (2003) Inferring high-level behavior from low-level sensors. In: Proceedings of the 8th international conference on ubiquitous computing. Springer, Berlin, pp 73–89
Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction recommender system—a case study. In: Proceeding of ACM WebKDD Workshop, Boston, MA
Takeuchi Y, Sugimoto M (2006) CityVoyager: an outdoor recommendation system based on user location history. In: Proceedings of the 3rd International Conference Ubiquitous Intelligence and Computing, Wuhan, China, September 2006. Springer press, Berlin, pp 625–636
Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. In: Proceedings of the international conference on data engineering
Xiao X, Zheng Y, Luo Q, Xie X (2010) Finding similar users using category-based location history. Poster. In: Proceedings of ACM SIGSPATIAL conference on advances in geographical information systems
Ye Y, Zheng Y, Chen Y, Feng J, Xie X (2009) Mining individual life pattern based on location history. In: Proceedings of the international conference on mobile data management. IEEE press, pp 1–10
Yi B, Jagdish H, Faloutsos C (1998) Efficient retrieval of similar time sequences under time warping. In: Proceedings of the international conference on data engineering
Zhang D, Guo B, Yu Z (2011) The emergence of social and community intelligence. Computer 44(7):21–28
Zheng Y (2011) Location-based social networks: users. In: Zheng Y, Zhou X (eds) Computing with spatial trajectories, 1st edn. Springer, New York
Zheng Y, Xie X (2009b) Learning location correlation from GPS trajectories. In: Proceedings of the international conference on mobile data management, IEEE Press, pp 27–32
Zheng Y, Xie X (2011a) Location-based social networks: locations. In: Zheng Y, Zhou X (eds) Computing with spatial trajectories, 1st edn. Springer, New York
Zheng Y, Xie X (2011b) Learning travel recommendations from user-generated GPS traces. ACM Trans Intell Syst Technol 2(1):2–29
Zheng Y, Zhou X (2011) Computing with spatial trajectories. Springer, New York. ISBN 978-1-4614-1628-9
Zheng Y, Li Q, Chen Y, Xie X, Ma WY (2008a) Understanding mobility based on GPS data. In: Proceedings of 10th International Conference on Ubiquitous Computing, Seoul, South Korea, September 2008. ACM Press, pp 312–321
Zheng Y, Liu L, Wang L, 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, New York, pp 247–256
Zheng Y, Wang L, Zhang R, Xie X, Ma WY (2008c) GeoLife: managing and understanding your past life over maps. In: Proceedings of the 9th international conference on mobile data management. IEEE Press, pp 211–212
Zheng Y, Chen Y, Xie X, Ma WY (2009a) GeoLife2.0: a location-based social networking service. In: Proceedings of International Conference on Mobile Data Management 2009. IEEE Press, pp 357–358
Zheng Y, Zhang L, Xie X, Ma WY (2009c) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of 18th international conference on world wild web, ACM Press, New York pp 791–800
Zheng VW, Cao B, Zheng Y, Xie X, Yang Q (2010a) Collaborative filtering meets mobile recommendation: a user-centered approach. In: Proceedings of 24th AAAI conference on Artificial Intelligence, Atlanta, USA, July 2010. AAAI press, pp 236–241
Zheng Y, Chen Y, Li Q, Xie X, Ma WY (2010b) Understanding transportation modes based on GPS data for web applications. ACM Trans Web 4(1):1–36
Zheng Y, Xie X, Ma WY (2010c) GeoLife: a collaborative social networking service among user, location and trajectory. IEEE Date Eng Bull 33(2):32–40
Zheng VW, Zheng Y, Xie X, Yang Q (2010d) Collaborative location and activity recommendations with GPS History Data. In: Proceeding of the 19th international conference on world wide web, ACM Press, New York, pp 1029–1038
Zheng Y, Zhang L, Ma Z, Xie X, Ma WY (2011) Recommending friends and locations based on individual location history. ACM Trans Web 5(1):5–44
GeoLife GPS trajectories (2010) http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/default.aspx