Information Sciences 236 (2013) 17–32

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Mining user similarity based on routine activities

Mingqi Lv [a,b], Ling Chen [a,*], Gencai Chen [a]

[a] College of Computer Science, Zhejiang University, Hangzhou 310027, PR China
[b] Hangzhou Normal University, Hangzhou 310012, PR China

ARTICLE INFO

Article history:
Received 17 May 2011
Received in revised form 28 January 2013
Accepted 23 February 2013
Available online 4 March 2013

Keywords:
User similarity
Routine activity
Data mining
Trajectory
Location-based social network

ABSTRACT

Mobile user similarity is significant for location-based social network services. With the pervasiveness of location-acquisition technologies, research on measuring mobile user similarity based on their trajectories has attracted a lot of attention. However, trajectories imply only short-term mobile regularities, and thus users' long-term activity similarity is difficult to be captured. In this paper, we address the problem of mining users' long-term activity similarity based on their trajectories. To solve this problem, we propose a two-stage approach. At the first stage, the notion of *routine activity* is proposed to capture users' long-term activity regularities. The routine activities of a user are extracted from his/her daily trajectories. At the second stage, user similarity is calculated hierarchically based on the extracted routine activities. Finally, we evaluated our approach based on both real and artificial datasets. The experimental results show that users with different profiles can be discriminated on the basis of our similarity metric, and thus demonstrate the effectiveness of our approach.

1. Introduction

Recently, a new branch of social network services, called location-based social networks (LBSNs), has emerged [8,11,43]. As compared with the traditional social network services, LBSNs bring users from the virtual world back to the real world and allow the real-life experiences to be shared in the virtual world by incorporating location features. With the proliferation of the mobile devices with locating ability, many LBSNs allow the users not only to be aware of their friends' location, but also to share their historical trajectories [15,43].

Besides raw locations and trajectories, future LBSNs are expected to understand their users' high-level context (e.g. activities, preferences, etc.) [14,45], and provide services more adaptively and intelligently. Potential intelligent LBSNs services include recommendation service which recommends places, activities, friends or other geo-related information to the users [17,33,39,42], information feeding service which shares and disseminates information within the network [29,31] and so on. Obviously, user similarity is crucial to these services. Most existing works which exploited user similarity in LBSNs focused on analyzing the geographic or sequential features of users' trajectories [28,30,40]. However, a trajectory which consists of a sequence of geographic points with timestamps only implies temporary moving activities of the user, and it cannot reflect his/her long-term activity regularities (e.g. where did the user most probably spend the weekend, how long did the user usually stay at work during weekdays, etc.). Therefore, users' long-term activity similarity cannot be captured by merely considering short-term trajectories. For example, even two users who have totally different long-term activity regularities (e.g. a student and a chef) may share similar trajectories (e.g. going from home to a restaurant).

In this paper, the users' long-term activity regularities are exploited by using the concept of routine activity. We define routine activity as the repeating activities at a few highly frequented locations with regular time intervals. For example, the

* Corresponding author. Tel.: +86 13606527774.
E-mail addresses: lvmingqi1104@163.com (M. Lv), lingchen@cs.zju.edu.cn (L. Chen), chengc@zju.edu.cn (G. Chen).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.02.050

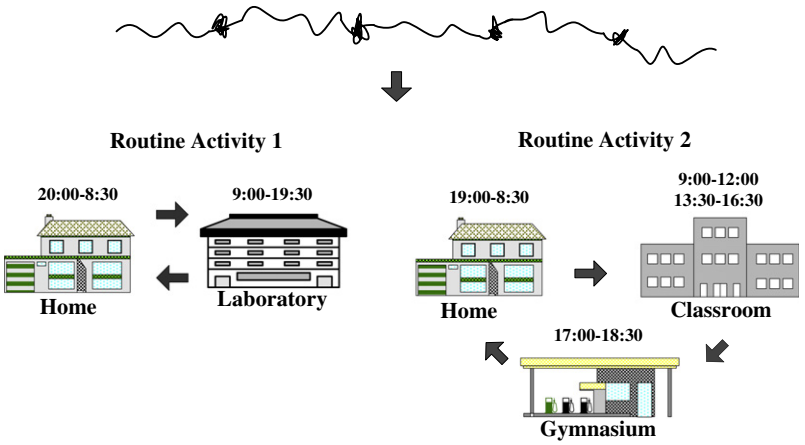

Fig. 1. From raw trajectories to multiple routine activities (the dark line on the top represents the user's raw trajectories, and the two figures in the bottom represent the user's long-term activity regularities, i.e. *Routine Activity* 1 and *Routine Activity* 2).

daily routine activities of a student Tom can be summarized in Fig. 1. Since routine activity reflects both the temporal and the spatial regularities of people's daily lives, we take it as the basis to measure the long-term similarity between two users. However, measuring user similarity based on routine activity is a challenging task. We need to address the following two problems.

How to extract routine activities from raw trajectories? First, the low-level trajectory data (as shown in the top of Fig. 1) does not contain semantic meanings, so it is almost impossible to extract routine activities (as shown in the bottom of Fig. 1) directly from it. Second, even if the semantic meanings of the trajectories are known, the routine activities still need to be mined from the data, because people's activities often vary from day to day, and the temporal and spatial patterns will not be exactly the same even if people repeat the same activities.

How to measure user similarity based on their routine activities? The routine activity is defined as the repeating activities at certain locations with regular time intervals. However, it is impossible to estimate user similarity based on the geographic features of the routine activities, because people are not necessarily to visit the same places even if they share similar routine activities, and people may have totally different routine activities even if they visit the same places. This problem is further complicated by the situation when the users for comparison have multiple routine activities.

To address these challenges, this paper proposes a complete approach to estimate user similarity based on routine activities extracted from their GPS trajectories. First, we extract significant places from a user's trajectories and formulate his/her 1-day activities as place preference matrices. Second, we extract patterns (i.e. routine activities) from these matrices by applying clustering techniques. Third, a novel similarity measurement is proposed to estimate the similarity of routine activities, and the user similarity is estimated by how similar their routine activities are. Our work is devoted to studying both individual and group activities based on their trajectories. It contributes to the so called "computational social science" research [24] which provides a data-driven way to understand our lives, organizations and societies. The primary contributions made in this paper are summarized as follows.

- A 1-day activity model which is constructed on the basis of the frequently visited places extracted from the historical trajectories is presented.
- A 1-day activity clustering algorithm for mining users' long-term activity regularities (i.e. routine activities) is developed.
- A novel user similarity measure based on the notion of routine activities is proposed.
- A comprehensive experiment is conducted, and the results show the proposed method is capable of capturing users' long-term activity regularities and similarities.

The remainder of this paper is organized as follows. Section 2 gives a survey of the related work. In Section 3, after clarifying some definitions, we propose the architecture of our approach. Section 4 presents the system for routine activity mining. Section 5 details the user similarity calculation approach based on routine activities. The evaluation of our approach and the experimental results are reported in Section 6. In Section 7, we conclude our work and give some ideas about the future work.

2. Related work

Our work is most related to routine activity mining from trajectory data, and user similarity measurement based on the real-world activities.

2.1. Trajectory mining

Motivated by the convenience of data collection, many studies have been performed based on GPS trajectories in the recent years. Some of the existing works detected significant locations of a user by applying clustering approach, which can be generally divided into three types, i.e. partitioning clustering [1], density-based clustering [44] and time-based clustering [22]. Other existing works tried to extract route patterns from trajectories [20,25] and use them to predict future movement of the user [6,7,21]. However, these works focused on analyzing short-term movement regularities of individual users, and we tend to extract long-term routine activities from the trajectories.

2.2. Routine activity mining

Some previous works explored the possibility of using low-level sensors for daily routine activity extraction. Blanke and Schiele used occurrence statistics of distinctive low-level activities to train a discriminative classifier for daily routine activities [3]. Ros et al. proposed an approach for learning daily activities in a smart house environment, and used the knowledge to recognize normal and abnormal human activities in real time [34]. However, the routine activities researched in these works are short-term daily activities (e.g. walking, eating, etc.) that are common for all users. Other previous works studied users' long-term activities under wireless network environments. Hsu et al. proposed TRACE, an approach to analyze users' long-term activity patterns by mining wireless network logs [19]. The users' activities are represented by location preference vectors, and matrix decomposition techniques are used to identify major activity patterns. Farrahi and Gatica-Perez adopted both supervised methodology and unsupervised methodology to analyze the human routine activities from mobile phone data. The supervised methodology is designed to classify human routines defined by group type [12], and the unsupervised methodology is developed to discover human routines which characterize both individual and group behaviors in terms of location patterns [13]. For spatiotemporal database, Li et al. addressed the problem of mining periodic activities for moving objects [27]. First, multiple periods in the spatiotemporal data are retrieved using a method that combines Fourier transform and autocorrelation. Then, a probabilistic model is proposed to characterize the periodic activities which are statistically generalized through hierarchical clustering. Although these works have analyzed users' long-term activities from different aspects, they do not take into account how to measure users' similarity based on their activities. For example, the similarity between locations visited by different users could either not be calculated [19,27], or the locations were manually labeled [12,13].

2.3. User similarity measurement

Many existing works have discussed how to measure user similarity in geographic environments. Most of them adopted the idea of trajectory based measurement to derive user similarity by analyzing the movement regularities of mobile users. Zheng et al. proposed a system for measuring user similarity based on historical trajectories [42]. The system first extracts stay points from each individual's trajectories and organizes them as a hierarchical framework. Then some similar sequences which stand for two individuals sharing the property of visiting the same stay points with a similar time interval are discovered in each level, and the user similarity is calculated based on the retrieved similar sequences. Lu et al. proposed a transaction similarity measurement named LBS-Alignment to calculate the similarity of two mobile users by analyzing the longest common sequence within their mobile sequential patterns [30]. Thakur et al. modeled users' location preferences in wireless environment using association matrix and its SVD, and used the eigen vectors to quantitatively measure the similarity of mobile user pairs [36]. However, all these approaches rely on geographic overlapping, so they cannot evaluate the similarity of two users living far away from each other. Some other existing works exploited user similarity based on semantic analysis of the trajectories. Lee and Chung proposed a method to calculate the user similarity using the semantics of visited locations [26]. However, the method constructed the location semantics by leveraging existing social network services, while raw trajectory data does not contain semantic meanings. Ying et al. proposed a user similarity measurement called MSTP-Similarity [40]. It first transforms the geographic trajectories into semantic trajectories by using a geographic information database, and then extracts sequential patterns from the semantic trajectories. The user similarity is measured based on their maximal semantic trajectory patterns. Although this approach releases the geographic constraint on user similarity measurement, it still focuses on analyzing short-term movement activities, and the semantic meaning of a location retrieved from database may often not reflect its personal meanings, e.g. a location cannot be determined as a user's home or workplace by searching in a public geographic database.

2.4. Recommendation systems

One kind of application that makes the most extensive use of user similarity information is recommendation system. Collaborative filtering [2,4], one of the most outstanding approaches used in such systems, is based on user similarity. For example, Zhao et al. proposed a novel approach which leverages the relationship strengths and interest similarities between users to improve the accuracy of video recommendation [41]. De Meo et al. explored user similarity on a social inter-networking context by considering both the explicit and implicit relationships among them to recommend similar users, resources and social networks [9]. For real-world recommendation systems based on location history, Geowhiz [18] and CityVoyager [35]

have been designed to recommend POIs (i.e. points of interest, e.g. shops, restaurants, etc.) by exploring users' preference similarities and moving patterns. Zheng et al. proposed a system for personalized friend and location recommendation based on user similarity extracted from historical trajectories [42]. All the above mentioned systems calculated user similarity based on temporary or short-term interest indicators, e.g. clicks, views or queries for the online systems, and visits or passes for the real-world systems.

3. System overview

In this section, we first clarify some concepts used in this paper. Then, we present the architecture of our system.

3.1. Preliminary

Our system for user similarity estimation is based on routine activity extracted from raw GPS data. First, we clarify some concepts and their data representation, including GPS point, GPS trajectory, visit point, reference place, 1-day activity and routine activity.

Definition 1 (*GPS point and GPS trajectory*). A GPS point is a pair $p = (lng, lat)$, representing the longitude–latitude location. A GPS trajectory is a sequence of pairs $Traj = \langle (p_0, t_0), \ldots, (p_n, t_n) \rangle$, in which p_k is a GPS point and t_k ($k = 0 \cdots n$) is a timestamp ($\forall 0 \leqslant k < n, t_k < t_{k+1}$).

Definition 2 (*Visit point and reference place*). A visit point is a triple $VP = (p, t_{in}, t_{out})$, where p is a GPS point, t_{in} and t_{out} are timestamps, and the visit point stands for a location p around which the user stays for longer than a time threshold (i.e. $t_{out} - t_{in} > \delta_{time}$). A reference place is a collection of visit points $P = \{VP_1, \ldots, VP_n\}$, in which $VP_1.p, \ldots, VP_n.p$ are close to each other.

According to the definition, reference places can be viewed as the significant places (e.g. home, work, etc.) where the user frequently visits. Because people's daily routes can be characterized by a significant probability to return to a few highly frequented locations [16], reference places can better capture users' activity patterns than raw GPS trajectories.

Definition 3 (*One-day activity*). A 1-day activity is a place preference matrix OA, each row of which represents an extracted reference place, and each column of which represents a discrete time span of a day. Each entry e_{ij} ($1 \leqslant i \leqslant d, 1 \leqslant j \leqslant T$, d is the number of reference places visited that day and T is the number of discrete time spans) of OA is the time duration that the user stays at the ith reference place RP_i during the jth time span of that day.

For example, assume that 1 day is divided into 24 time spans, and Tom visited two places (i.e. home, lab) on a specific day. His 1-day activity of that day is partially portrayed in Fig. 2a. The "on the way" row stands for the time Tom spent shuttling between home and lab in each time span.

Definition 4 (*Routine activity*). A routine activity is a probability distribution matrix A, each entry e_{ij} ($1 \leqslant i \leqslant D, 1 \leqslant j \leqslant T$, D is the number of all extracted reference places and T is the number of discrete time spans) of which is the probability that the user is at the ith reference place RP_i during the jth time span.

(a)

	8:00~9:00	9:00~10:00	...	19:00~20:00	20:00~21:00
home	21	0	...	5	60
lab	7	60	...	25	0
on the way	32	0	...	30	0

.....

(b)

	8:00~9:00	9:00~10:00	...	19:00~20:00	20:00~21:00
home	0.4	0	...	0.1	0.95
lab	0.1	0.98	...	0.35	0
...	...	...	...	...	...
on the way	0.5	0.02	...	0.55	0.05

Fig. 2. Data representation of Tom's activities: (a) 1-day activities (the number in the white blank denotes the time duration that Tom stays at the corresponding place during the given time span, and the unit of time duration is minute). (b) Routine activity (the number in the white blank denotes the probability that Tom is at the corresponding place during the given time span).

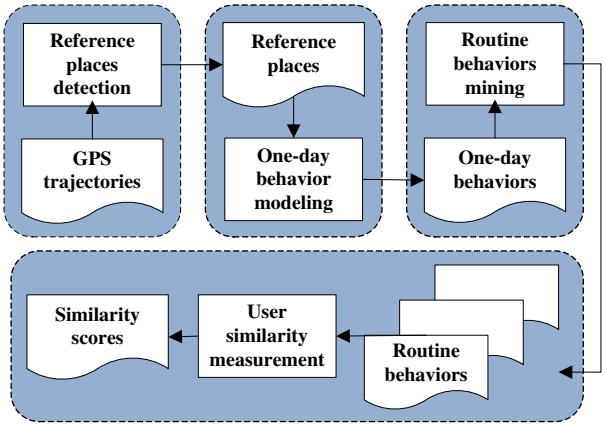

Fig. 3. The system architecture of user similarity mining.

As shown in Fig. 2b, a routine activity is an integrated representation of all the similar 1-day activities. The probability distribution matrix encodes the uncertainties of the user's 1-day activities, and statistically characterizes his/her routine activity.

3.2. Architecture

Fig. 3 gives an overview of the system architecture of our approach for user similarity mining. Given GPS trajectories of multiple users, our approach calculates the similarity score of each pair of users through four steps. For each individual user, the system firstly extracts reference places from their GPS trajectories. Secondly, the system transforms the original GPS trajectories into 1-day activities as a daily style. Thirdly, the system extracts patterns from 1-day activities based on routine activity model. Fourthly, the system measures similarity between multiple users based on their routine activities.

4. Routine activity mining

In this section, we will discuss the process of routine activity mining. For each individual user, we first extract reference places from his/her raw GPS trajectories, and abstract the original trajectories as a daily style based on the 1-day activity model. Second, we construct routine activity models by discovering patterns from all his/her 1-day activities.

4.1. Reference places extraction

This paper proposes a hierarchical clustering algorithm to extract reference places from GPS trajectories with a three-layered architecture as shown in Fig. 4. The algorithm takes GPS trajectories as input and conducts a time-based clustering to identify visit points, and then adopts a density-based clustering to group these visit points into reference places.

The time-based clustering algorithm depicted in Fig. 5 works in an incremental way and processes the GPS points in the GPS trajectory along the time axis. In the algorithm, T is the GPS trajectory that contains all the GPS points sorted by sampling time, CC and LC are the sets of GPS points that respectively belong to the current cluster and the last cluster. For each GPS point in T, the algorithm compares the distance between it and the centroid of the current cluster with the clustering

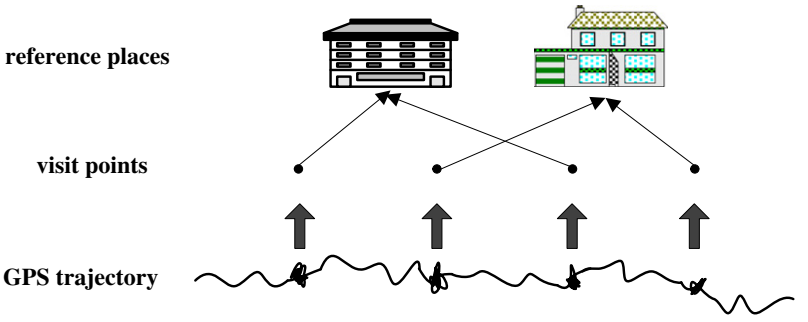

Fig. 4. Reference places extraction based on hierarchical clustering.

ALGORITHM 1 *Visit Point Extraction*(T)

1: current cluster $CC = \emptyset$, last cluster $LC = \emptyset$
2: **for** each GPS point p_i in T **do**
3: **if** distance(CC, p_i) < $\delta_{cluster\_distance}$ **then**
4: Append p_i to CC
5: **else**
6: **if** duration(CC) > δ_{time} **then**
7: Append CC to *VisitPoints*
8: $CC = \emptyset$, $LC = \emptyset$
9: **else**
10: **if** interval(CC, LC) > δ_{time} **and** distance(CC, LC) < $\delta_{tolerated\_distance}$ **then**
11: CC = combine(CC, LC)
12: Append CC to *VisitPoints*
13: $LC = \emptyset$
14: **else**
15: $LC = CC$
16: $CC = \emptyset$

Fig. 5. Time-based clustering algorithm for visit points extraction.

distance threshold $\delta_{cluster\_distance}$. If the distance is less than $\delta_{cluster\_distance}$, the GPS point is added to the current cluster (lines 2–4). Otherwise, the algorithm checks the time duration of the current cluster. If the time duration is longer than the time threshold δ_{time}, the current cluster is considered as a visit point (lines 6–8). If the time duration is not long enough, the algorithm does not simply ignore it, but checks the time interval and distance between the current cluster and the last cluster. If the time interval is longer than δ_{time} and the distance is less than the tolerated distance $\delta_{tolerated\_distance}$, the algorithm combines these two clusters and treats the result as a visit point (lines 10–13). The Visit Point Extraction algorithm has a linear time complexity $O(|T|)$ (where $|T|$ is the number of GPS point in GPS trajectory T).

The reason that we use two cluster variables (i.e. CC and LC) in the algorithm is that the GPS points sampling always has *entrance and exit deviation problems*, which are illustrated in Fig. 6. For example, if a user enters a large building from one side A and leaves from another side B (as shown in Fig. 6a), the GPS signal will be blocked and the GPS points recorded around A and those recorded around B will form two different clusters (i.e. cluster I and II). Apparently, neither cluster I nor II could be identified as visit point even if the user stays a long time in the building. This problem may also exist even when the user enters and leaves the building from the same sides, because GPS device often require a period of time to receive signal when the user may leave the building for a relatively long distance (as shown in Fig. 6b). By considering two consecutive clusters and a tolerated distance threshold $\delta_{tolerated\_distance}$, this problem can be greatly alleviated when the GPS sampling is interrupted for a long period of time between two areas which are not too far from each other.

The output of the visit point extraction algorithm is a set of clusters, each of which represents a visit point $VP = (p, t_{in}, t_{out})$, where p is the centroid of the cluster, t_{in} and t_{out} are the timestamps of the first and the last GPS points of the cluster respectively. Since multiple visit points may belong to the same reference place, we use a density-based clustering algorithm (e.g. DBSCAN) to group the extracted visit points into reference places.

The hierarchical clustering algorithm for reference place extraction outperforms pure density-based clustering [44] and time-based clustering [22] algorithms from the following aspects. First, it is difficult for pure density-based clustering

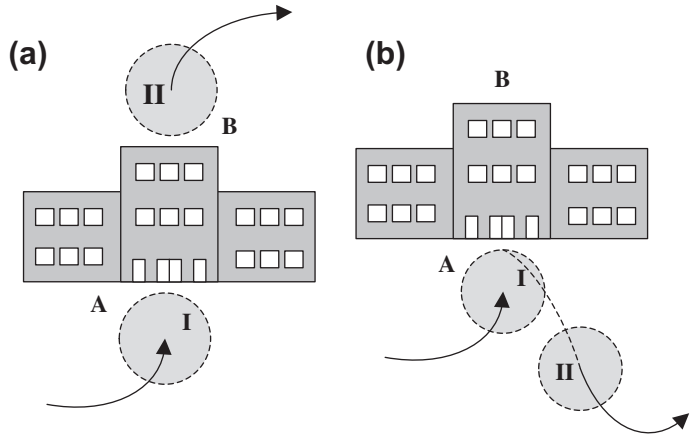

Fig. 6. The entrance and exit deviation problems of GPS sampling.

algorithm to find indoor reference places with sparse data points due to GPS signal loss problem, whereas our hierarchical clustering algorithm can incrementally find both indoor and outdoor reference places along with the GPS points collection procedure. Second, pure time-based clustering algorithm does not consider the revisit to the same places, whereas our hierarchical clustering algorithm can discover reference places which are visited multiple times.

After discovering all the reference places, we abstract the original GPS trajectory in a daily style based on the 1-day activity model. We divide a day into 24 time spans (i.e. one hour as a time span) as the columns of the place preference matrix. For each reference place visited in a specific day and each time span of that day, we check the visit points of the reference place to find how long the user stay in the reference place during the corresponding time span, and fill in the corresponding entry with the result. If the sum of staying time distributed in all reference places during a specific time span is less than an hour, we assign the remaining time to the "on the way" row of the corresponding time span. Take the example shown in Fig. 2a: Tom stayed at lab for 7 min during 8:00–9:00 and 60 min during 9:00–10:00.

4.2. Routine activities mining

One-day activities represent users' temporal and spatial activities for specific days. Routine activities mining problem is essentially a problem of discovering activity patterns from multiple 1-day activities. We accomplish this task in two steps. First, we find groups of 1-day activities which are similar to each other. Second, the 1-day activities within the same group are represented based on the routine activity model.

A potential solution to find similar 1-day activities is to use clustering methods. To apply clustering method, we need to define a similarity measure and choose an appropriate clustering algorithm. Since a 1-day activity is represented by a place preference matrix which can be further decomposed into 24 place preference vectors, each of which stands for the distribution of time spent at each reference place during the specific time span of that day, we use cosine coefficient to measure the similarity between each pair of place preference vectors (i.e. $\boldsymbol{X} = (x_1, \ldots, x_D)$, $\boldsymbol{Y} = (y_1, \ldots, y_D)$) as Eq. (1), in which D is the number of the total extracted reference places of the user, x_i and y_i are the time duration the user stayed at the ith reference place during the corresponding time span in two different days.

$$\text{cosine similarity} = \frac{\sum_{i=1}^{D} x_i y_i}{\sqrt{\sum_{i=1}^{D} x_i^2}\sqrt{\sum_{i=1}^{D} y_i^2}} \tag{1}$$

The similarity between two 1-day activities is the average value of the similarity of their decomposed place preference vectors with the same time span.

The purpose of using clustering algorithm is to partition all the 1-day activities into K groups, the activities within which are similar to each other. However, K (i.e. the number of underlying routine activities) is always unknown, so we propose a bottom-up agglomerative clustering algorithm to group the 1-day activities while determining the optimal number of clusters at the same time. In the algorithm (as shown in Fig. 7), *OAS* is the set of 1-day activities for a specific user, and *CS* is the set of 1-day activity clusters. The algorithm firstly treats each 1-day activity as a singleton cluster (lines 1–3) and constructs a similarity matrix, where the value of the ith row and jth column is the similarity score between the ith and jth singleton clusters (line 4). Then, the algorithm successively merges pairs of clusters until the clustering termination condition is satisfied. At each iteration, the algorithm merges the two clusters with maximum similarity and updates the similarity matrix accordingly (lines 6–9). The similarity of two 1-day activity clusters is calculated by averaging the similarities between each pair of 1-day activities of the two clusters.

An ideal clustering result should maximize the *intra-cluster similarity* (i.e. the average similarity between pairs of 1-day activities in the same cluster) as well as minimize the *inter-cluster similarity* (i.e. the average similarity between pairs of 1-day activities of different clusters), thus we used a variant of the *Dunn index* [10] to measure the clustering quality as Eq. (2), in which $s_{intra}(C_k)$ stands for the intra-cluster similarity of cluster C_k, $s_{inter}(C_i, C_j)$ denotes the inter-cluster similarity of cluster

ALGORITHM 2 *One-day Activity Clustering*(*OAS*)

1: one-day activity cluster set $CS = \emptyset$
2: **for** each one-day activity OA in OAS **do**
3: Append OA to CS as a one-day activity cluster AC
4: Construct similarity matrix SM, each entry $SM[i][j]$ of which is the similarity between clusters AC_i and AC_j of CS
5: **while** clustering termination condition is not satisfied **do**
6: Find clusters AC_x and AC_y such that $x, y = \text{argmax}_{i,j} SM[i][j]$
7: $AC = \text{merge}(AC_x, AC_y)$
8: Remove AC_x and AC_y from CS
9: Update similarity matrix SM by computing the similarity between AC and other clusters

Fig. 7. One-day activity clustering algorithm for routine activity mining.

C_i and C_j. A higher value of Dunn index indicates a better clustering result. To determine whether to terminate the clustering algorithm, we monitor the change of *Dunn index* at each iteration of the clustering procedure, and stop the algorithm if the value of Dunn index decreases dramatically or all the 1-day activities have been merged into a single cluster. A dramatic decrease of Dunn index value indicates that the decreasing degree of the minimum intra-cluster similarity significantly exceeds that of the maximum inter-cluster similarity. It is a sign that the newly merged cluster may contain 1-day activities that belong to different routine activities, and the previous clustering result is likely to be of the optimal quality.

$$\text{Dunn index} = min(s_{intra}(C_k)) - \underset{i \neq j}{max}(s_{inter}(C_i, C_j)), \quad i,j,k = 1 \ldots |CS| \tag{2}$$

The major time cost of the 1-day Activity Clustering algorithm is on the agglomerative clustering process (lines 5–9). Let $|OAS|$ be the number of all the 1-day activities of the user, and N be the number of the currently remaining clusters in the similarity matrix SM. Line 6 requires $O(N^2)$ to search for the pair of 1-day activities with maximum similarity, and line 9 constructs a new similarity matrix to replace the current one which costs $O((N-1)^2)$ time. The agglomerative clustering process would combine all the 1-day activities into one cluster in the worst case. Thus, the total time complexity should be $O(|OAS|^3)$.

When the algorithm terminates, we get a collection of clusters, each of which contains a set of 1-day activities $\{OA_1, \ldots, OA_{|C|}\}$. The representing routine activity for a cluster is characterized by a probability distribution matrix, and each entry $p(x_j = i)$ of the matrix represents the probability that the user is at the ith reference place during the jth time span, which can be calculated based on Eq. (3). Take the example shown in Fig. 2b: Tom stays at lab for 10% confidence during 8:00–9:00 and 98% confidence during 9:00–10:00 as a certain fraction of his daily life.

$$p(x_j = i) = \frac{\sum_{k=1}^{|C|} OA_k \cdot e_{ij}}{T \times |C|} \tag{3}$$

5. User similarity calculation

This section explains how to measure user similarity based on their routine activities. This work can be hierarchically divided into three sub-problems, i.e. calculating the similarity score between two reference places, calculating the similarity score between two routine activities, and calculating the similarity score between two users.

5.1. Reference place similarity calculation

Given two routine activities, they are more similar when they have higher common probability to visit similar reference places during the same time spans. However, users may have similar routine activities even if they never visit the same reference places. For example, two users went to and off work at almost the same time and used to take exercise at gyms during night-time, but they live in different cities. Therefore, we are always unable to find similar routine activities based only on the geographic features of their reference places.

Most existing approaches [5,40] for finding similar places with no geographic overlapping are based on reverse geo-coding technology, i.e. querying the semantics of a place from a geographic database based on its geographic features. However, these approaches are only suitable for public places with no personal meaning. For example, we are always unable to find a user's home or workplace from a geographic database. Besides, even the same place may have different personal meanings to different users, e.g. a customer has dinner in a restaurant whereas a chef may work there. Since a large fraction of reference places as referred in routine activities have personal meanings, we analyze the similarity of the reference places of different routine activities based on the visiting patterns to them instead of reverse geo-coding. To achieve this goal, we first define visiting pattern as follow.

Definition 5 (*Visiting pattern*). A visiting pattern PV_{ik} to a reference place i of a routine activity k is a probability distribution vector, each element e_j ($1 \leqslant j \leqslant T$, T is the number of the discrete time spans) of which is the probability that the user is at the reference place i during the jth time span following routine activity k.

Apparently, each row of a routine activity is a visiting pattern, which can represent the temporal regularity of the user's visit activity to the specific reference place. The visiting patterns reflect the places' personal meanings to a great extent, e.g. users usually stay at home during night time and at restaurants during dinner time. So we calculate the similarity of places based on their visiting patterns. Since visiting pattern is a probability distribution vector, we use the *Kullback–Leibler divergence* as the distance measure. Given two visiting patterns PV_1 and PV_2, their Kullback–Leibler divergence can be calculated based on the following equation:

$$KL(PV_1||PV_2) = \sum_{j=1}^{T} PV_1 \cdot e_j \times log\frac{PV_1 \cdot e_j}{PV_2 \cdot e_j} \tag{4}$$

However, directly applying Kullback–Leibler divergence here has two problems. First, $KL(PV_1||PV_2)$ would become infinite when $PV_2 \cdot e_j = 0$. Second, the sum of all elements in a visiting pattern may not be one, so $KL(PV_1||PV_2)$ might has minus value. Besides, $KL(PV_1||PV_2)$ may not be equal to $KL(PV_2||PV_1)$. For the first problem, we use a smoothing parameter λ ($0 < \lambda < 1$) and a

	Home	Work	On the way
PD_1	1.0	0	0
PD_2	0	1.0	0
PD_3	0.1	0	0.9

s(Home, Work) = 0.2
s(Home, Home) = 1.0
s(Home, Way) = 0.1

$s(PD_1, PD_2) = 1.0\times0.2 = 0.2$
$s(PD_1, PD_3) = 0.1\times1.0+0.9\times0.1 = 0.19$

Fig. 8. An example of place distribution vector similarity calculation problem when considering all reference place pairs (PD_1 and PD_3 should have higher similarity but result in lower similarity score).

background variable e to make the element e_j of a visiting pattern always greater than zero (as shown in Eq. (5)). For the second problem, we use the average value of $KL(PV_1||PV_2)$ and $KL(PV_2||PV_1)$ as the final distance, and calculate their similarity based on Eq. (6).

$$e_j = (1-\lambda)e_j + \lambda e \tag{5}$$

$$s_{place} = log\frac{1}{(KL(PV_1||PV_2) + KL(PV_2||PV_1))/2} \tag{6}$$

5.2. Routine activity similarity calculation

Since a routine activity is composed of place distribution vectors belonging to each time span, we can calculate the similarities of all pairs of place distribution vectors with the same time span of two routine activities and average them to get the similarity between the two routine activities. For each pair of place distribution vectors, we measure their similarity taking into account both the similarity and the common probability of the corresponding reference place pairs (common probability is the smaller one of the two probabilities respectively assigned to the two reference places). However, considering all pairs of reference places between two place distribution vectors may cause an abnormal situation where the calculated similarity is higher when the users stay at dissimilar places with higher common probability than that when the users stay at similar places with lower common probability. As in the example shown in Fig. 8, place distribution vector PD_1 and PD_3 should have higher similarity than that of PD_1 and PD_2 since PD_1 and PD_3 have 10% common probability to stay at similar places (i.e. home), and PD_1 and PD_2 spend all their time to stay at dissimilar places, but we get opposite result when considering all reference place pairs. This is because the high common probability favors the weighted similarity between dissimilar reference places. This problem has also been observed for estimating video similarity [37,38].

To avoid the above mentioned problem, we should consider only the reference place pairs with similar semantics for similarity calculation. Consider the example in Fig. 8, PD_1 and PD_2 do not have common probability to stay at places with similar semantics, whereas PD_1 and PD_3 have 10% common probability to stay at "Home", so $s(PD_1, PD_2) = 0$ and $s(PD_1, PD_3) = 0.1 \times 1.0 = 0.1$. However, it is almost impossible to accurately assign personal semantic meanings to reference places due to the previously mentioned reason. Thus, we propose the concept of *Optimal Matching Sequence* (OMS) to find the most matching place pairs for similarity calculation.

Definition 6 (*OMS*). Given two routine activities A_1 and A_2, and their corresponding reference places sets $PS_1 = \{P_{11}, \ldots, P_{1m}\}$ and $PS_2 = \{P_{21}, \ldots, P_{2n}\}$, the Optimal Matching Sequence is $OMS(A_1, A_2) = \langle(OP_{11}, OP_{21}), \ldots, (OP_{1s}, OP_{2s})\rangle$ ($OP_{1i} \in PS_1$, $OP_{2i} \in PS_2$, $1 \leqslant i \leqslant s$, $s = \min(m, n)$), which maximize the following function.

$$\sum_{i=1}^{s} s_{place}(OP_{1i}, OP_{2i}) \tag{7}$$

The algorithm for discovering OMS from two sets of reference places is shown in Fig. 9. We apply dynamic programming to solve the OMS discovery problem, where a matrix E is used to store the maximum sum of similarities (calculated by Eq. (7)) at each step. For two reference place sets PS_1 and PS_2, the entries of the matrix are gradually filled when the dynamic programming algorithm is performed (lines 4–6), and the last entry stores the final sum of similarities of the OMS. Next, we decode the matrix to find all the matching place pairs of OMS (lines 8–14). Let $|PS_1|$ and $|PS_2|$ be the number of reference places of PS_1 and PS_2 respectively, the time complexity of the OMS Discovery algorithm based on dynamic programming is $O(|PS_1||PS_2|)$.

Finally, given two routine activities A_1 and A_2, and their corresponding reference place sets PS_1 and PS_2, their similarity can be calculated based on Eq. (8), where $P_{1i} \in PS_1$, $P_{2k} \in PS_2$, OMS is the Optimal Matching Sequence of A_1 and A_2, $\min(A_1 \cdot e_{ij}, A_2 \cdot e_{kj})$ is the common probability of reference places i and k within the jth time span, T is the number of time spans.

ALGORITHM 3 *OMS Discovery*(PS_1, PS_2)

1: $row = |PS_2|+1$, $col = |PS_1|+1$, $row \times col$ matrix E
2: **for** each entry $E[i, j]$ in E **do**
3: $E[i, j] = 0$
4: **for** i in 1 to $|PS_2|$ **do**
5: **for** j in 1 to $|PS_1|$ **do**
6: $E[i, j] = \max(E[i, j\text{-}1], E[i\text{-}1, j], E[i\text{-}1, j\text{-}1] + s_{place}(PS_1[j], PS_2[i]))$
7: $OMS = \emptyset$, $i = |PS_2|$, $j = |PS_1|$
8: **while** $i > 0$ and $j > 0$ **do**
9: **if** $E[i, j] == s_{place}(PS_1[j], PS_2[i]) + E[i\text{-}1, j\text{-}1]$ **then**
10: Append $(PS_1[j], PS_2[i])$ to OMS, i--, j--
11: **else if** $E[i, j] == E[i\text{-}1, j]$ **then**
12: i--
13: **else if** $E[i, j] == E[i, j\text{-}1]$ **then**
14: j--

Fig. 9. The Optimal Matching Sequence discovery algorithm.

$$s_{routine} = \frac{\sum_{j=1}^{T}\sum_{(P_{1i},P_{2k})\in OMS} s_{place}(P_{1i}, P_{2k}) \times min(A_1 \cdot e_{ij}, A_2 \cdot e_{kj})}{T} \tag{8}$$

5.3. User similarity calculation

Since a user may have multiple routine activities, we consider the similarity between two users according to the similarities of all their routine activities. Let $AS(U_1) = \{A_{11}, \ldots, A_{1m}\}$ and $AS(U_2) = \{A_{21}, \ldots, A_{2n}\}$ be the routine activities of users U_1 and U_2, respectively. The similarity between U_1 and U_2 is calculated by the following equation:

$$s_{user} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} w(A_{1i}, A_{2j}) \times s_{routine}(A_{1i}, A_{2j})}{\sum_{i=1}^{m}\sum_{j=1}^{n} w(A_{1i}, A_{2j})} \tag{9}$$

where $w(A_1, A_2)$ stands for the weight of $s_{routine}(A_1, A_2)$. The reason to use a weighted value is that different routine activities may be of different importance to the user. Obviously, the higher support of a routine activity (i.e. how many times the user follows the routine activity), the more important the routine activity is. Thus, we define the weight $w(A_1, A_2)$ as the geometric mean of the supports to the two routine activities:

$$w(A_1, A_2) = \sqrt{support(A_1) \times support(A_2)} \tag{10}$$

6. Experiments

In this section, we conduct a number of experiments to evaluate the performance of our approach using both real and artificial datasets. The real dataset is collected from five participants, including students and faculties of our laboratory and their family members. In the data collection phase, we chose the widely used mobile phones as the recording devices. A program running on the mobile phone could connect to an external GPS receiver through Bluetooth and record GPS points at 1 Hz. All participants were instructed to carry out the experiment in an open-ended way to make the recorded movement data reflect their daily lives as truly as possible, and they were also asked to manually make a log using the program whenever they entered and left a place. The logs can be used as the ground-truth information about the places they have visited. The real dataset contains nearly 0.5 million GPS points collected from the five participants for nearly two months.

To collect the artificial dataset, we established a website which is developed upon the Google Maps API to enable the users to manually record their 1-day activities. To input 1-day activities, users can first define reference places and trajectories by interacting with the map, and then assign these elements to each time spans of a day. Users also input the background knowledge (e.g. reference places type, user profile, etc.) by filling the forms. We finally collect over 1500 1-day activities from over 100 registered users, who live in different areas of China.

6.1. Routine activity mining evaluation

This experiment tried to evaluate the effectiveness of the routine activity mining algorithm. Firstly, we tried to verify our time-based clustering algorithm for visit point extraction. The algorithm has totally found 321 visit points from the real dataset. For the incorrect results made by the algorithm, we distinguish them as either *false negative* (i.e. the algorithm reports

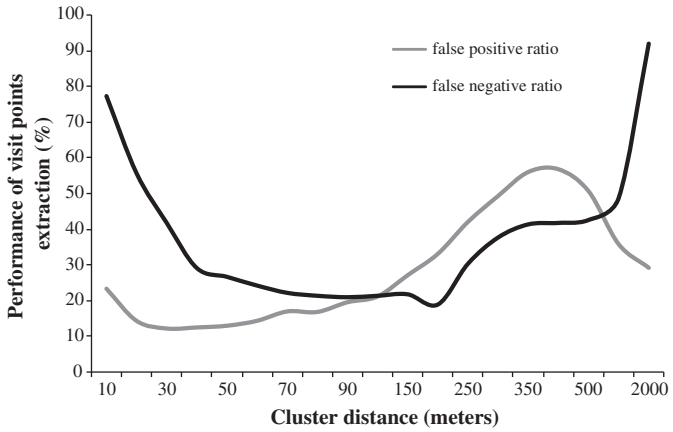

Fig. 10. The influence of cluster distance parameter on the algorithm performance of the existing time-based clustering.

the user is moving while he/she is actually visiting a place) or *false positive* (i.e. the algorithm reports the user is visiting a place while he/she is actually moving).

We compared our work with the existing time-based clustering algorithm [22] (we refer it as *ETC*), and tuned the parameters in order to reduce both false negative and false positive. The major difference is that ETC does not have the parameter $\delta_{tolerated\_distance}$, and it simply ignores the cluster whose time duration is not long enough. We fixed the parameter δ_{time} = 300 s based on the experiences gained in [22], and monitored the performance of ETC by adjusting the parameter $\delta_{cluster\_distance}$. Fig. 10 shows the influence of parameter $\delta_{cluster\_distance}$ on the *false positive ratio* (FPR, i.e. the ratio of the number of false positive to the number of extracted visit points) and the *false negative ratio* (FNR, i.e. the ratio of the number of false negative to the number of logged visit points) of the algorithm. It can be found from the figure that the FNR is significantly sensitive to $\delta_{cluster\_distance}$. Setting $\delta_{cluster\_distance}$ too low (lower than 40 m) or too high (higher than 200 m) will both result in high FNR. This is because setting $\delta_{cluster\_distance}$ too low will cause the extracted clusters too small to represent a visit point, and setting $\delta_{cluster\_distance}$ too high will cause multiple visit points to be merged into one cluster. Besides, we cannot simultaneously minimize FPR and FNR for ETC, so we configure $\delta_{cluster\_distance}$ to minimize Eq. (11) ($\delta_{cluster\_distance}$ = 50 m, and F = 0.175), which considers both FPR and FNR like *F-measure* used to evaluate the information retrieval systems.

$$F = \frac{2 \times \mathrm{FPR} \times \mathrm{FNR}}{\mathrm{FPR} + \mathrm{FNR}} \tag{11}$$

Our time-based clustering algorithm adopts a tolerated distance parameter $\delta_{tolerated\_distance}$ to alleviate the entrance and exit deviation problems of GPS sampling, and this means that our algorithm tends not to miss visit points even when $\delta_{cluster\_distance}$ is small. Therefore, we can fix $\delta_{cluster\_distance}$ to minimize the FPR and tune $\delta_{tolerated\_distance}$ to reduce the FNR. As shown in Fig. 11, we fix $\delta_{cluster\_distance}$ = 30 m to minimize the FPR (FPR = 12.27%), and the FNR is significantly decreased as increasing $\delta_{tolerated\_distance}$. However, setting $\delta_{tolerated\_distance}$ too high will cause the FPR increasing significantly, so we find an optimal $\delta_{tolerated\_distance}$ value to minimize Eq. (11) ($\delta_{tolerated\_distance}$ = 300 m, and F = 0.062). Based on the analysis, the visit point extraction performance of our time-based clustering algorithm could be greatly improved in comparison with the existing time-based clustering algorithm by appropriately setting the parameters.

Secondly, we evaluated the reference place clustering algorithm following the evaluation approach proposed in [23]. The algorithm has totally found 93 reference places from the real dataset. Reference places extracted by the algorithm are called *Discovered*, and reference places logged by participants are called *Logged*. Logged places that have not been extracted are called *Missed*, while extracted places that have not been logged are called *False*. Reference places that are both extracted and logged are further divided into *Correct* (i.e. a single place logged by participants has been extracted as a single place), *Merged* (i.e. two or more different places logged by participants have been extracted as a single place) and *Divided* (i.e. a single place logged by participants has been extracted as multiple places). To evaluate the algorithm performance, we define the following metrics (# stands for "number of"), where *FR*, *MsR*, *MeR*, *DR*, *P* and *R* denote False Rate, Missed Rate, Merged Rate, Divided Rate, Precision and Recall, respectively.

$$FR = \frac{\#\mathrm{False}}{\#\mathrm{Discovered}}, \quad \mathrm{MsR} = \frac{\#\mathrm{Missed}}{\#\mathrm{Logged}}, \quad \mathrm{MeR} = \frac{\#\mathrm{Merged}}{\#\mathrm{Discovered}} \tag{12}$$

$$DR = \frac{\#\mathrm{Divided}}{\#\mathrm{Logged}}, \quad P = \frac{\#\mathrm{Correct}}{\#\mathrm{Discovered}}, \quad R = \frac{\#\mathrm{Correct}}{\#\mathrm{Logged}} \tag{13}$$

Since the visit points can be grouped into reference places based on different kinds of clustering techniques, we compared three kinds of clustering algorithms, i.e. *Merging* (directly merging a new discovered visit point to a visit point set if the distance of them is less than a threshold $\delta_{merge\_distance}$, as proposed in [22]), *K-Means* (a variant of *K*-Means clustering algorithm

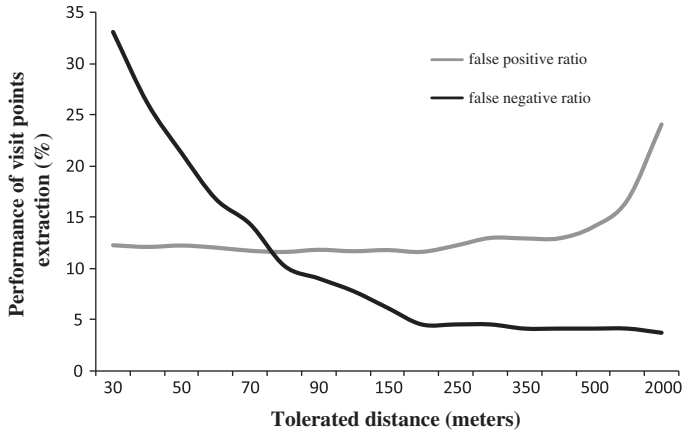

Fig. 11. The influence of tolerated distance parameter on algorithm performance of our time-based clustering ($\delta_{cluster\_distance}$ = 30 m).

as proposed in [1]) and *DBSCAN* (a variant of DBSCAN clustering algorithm as proposed in [44]). The parameters of the three algorithms were tuned to output the best performance ($\delta_{merge\_distance}$ of Merging is set to 100 m, *radius* of *K*-Means is set to 100 m, *Eps* of DBSCAN is set to 50 m). Table 1 shows the results obtained from the three algorithms. From the table, it can be found that *Merging* algorithm has the lowest performance because it creates reference places without a clustering procedure. As compared to *K-Means*, *DBSCAN* algorithm has a higher *MeR* value because it merges the overlapping visit points together to a great extent. However, DBSCAN algorithm has the lowest *DR* value because it can detect reference place with arbitrary shape, which can accommodate all its visit points, and thus achieves the best overall performance.

Thirdly, we tried to evaluate the 1-day activity clustering algorithm for routine activities mining. As the clustering algorithm terminates when the Dunn index value exhibits a significant decrease, we propose *DCR* (Dunn index Change Ratio) as Eq. (14) to quantize the change of Dunn index value. We set a change ratio threshold parameter δ_{DCR} (the algorithm terminates when $DCR > \delta_{DCR}$), and monitor the 1-day activity clustering quality by adjusting δ_{DCR}. For each user, the clustering quality is calculated based on *CNDR* (Cluster Number Difference Rate, as Eq. (15)), which reflects the relative difference between the number of the true routine activities (specified by the experiment participants and the registered users) and the number of the extracted routine activities. A low *CNDR* value indicates a better clustering result. Fig. 12 shows the influence of parameter δ_{DCR} on the average *CNDR* of all the users. It can be found from the figure that setting δ_{DCR} too low ($\delta_{DCR} < 0.3$) will significantly degenerate the clustering quality ($CNDR > 1.0$). Setting δ_{DCR} too high will not degenerate the clustering quality greatly, because users usually have limited number of routine activities (so the value of *CNDR* will not become too large even if only one cluster is found for each user). The clustering algorithm achieves the best performance ($CNDR \approx 0.2$) when setting δ_{DCR} around 0.7.

$$DCR = \frac{\text{Last Dunn index} - \text{Current Dunn index}}{\text{Last Dunn index}} \tag{14}$$

$$CNDR = \frac{|\text{Obtained cluster number} - \text{True cluster number}|}{\text{True cluster number}} \tag{15}$$

6.2. User similarity calculation evaluation

The routine activities capture users' long-term activity regularities. Since users with similar profile usually have similar routine activities, the routine activities can reflect the user's profile to a great extent. To demonstrate it, we exploit the ability of routine activity based user similarity to discriminate users with different profiles, which are specifically represented by the users' occupations in the experiment. Since the artificial dataset contains 1-day activities of a great number of users with different occupations (there are generally three kinds of occupation, i.e. the graduated student, the undergraduate student and the company employee), we evaluated our user similarity measure approach based on the artificial dataset.

Table 1
Reference place clustering performance of different algorithms.

	FR	*MsR*	*MeR*	*DR*	*P*	*R*
Merging	0.105	0.053	0.116	0.213	0.526	0.667
K-Means	0.093	0.053	0.093	0.173	0.747	0.747
DBSCAN	0.088	0.053	0.163	0.053	0.738	0.787

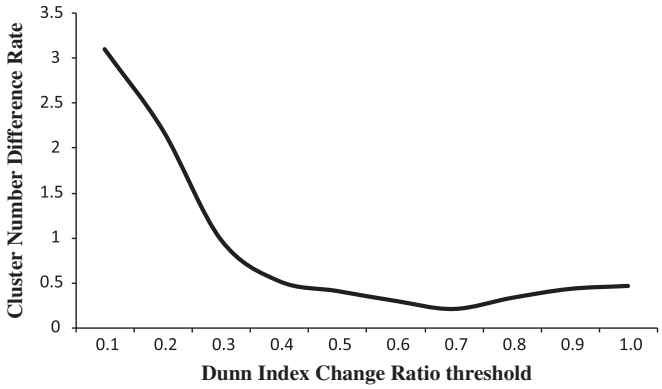

Fig. 12. The influence of Dunn index change ratio parameter on the quality of 1-day activity clustering.

In the experiment, we first extracted routine activities from the 1-day activities of all the registered users, and calculated the similarity between each user pair based on their routine activities. Then we used *K-Means* clustering algorithm to partition all the registered users into three groups based on their mutual similarities. To measure the performance of the clustering, for each kind of occupation, we found its *dominant cluster* (i.e. the cluster possessing most of the users with the corresponding occupation), and adopted the *intra-dominant rate* (i.e. the ratio of the number of users with the corresponding occupation in the dominant cluster to the total number of users in the dominant cluster) and the *inter-dominate rate* (i.e. the ratio of the number of users with the corresponding occupation in the dominant cluster to the total number of users actually with the corresponding occupation) to evaluate how well the users are partitioned according to their occupations based on our user similarity measurement (the higher of the values of the two metrics are, the better the performance is).

The intra-dominant rate (*IaDR*) and the inter-dominant rate (*IrDR*) of user clustering based on their similarity are shown in Fig. 13. The high intra-dominant rate (88.4% on average) and inter-dominant rate (87.1% on average) demonstrate that users' profiles can be well discriminated based on our similarity measurement, and this also means that users' long-term activity regularities can be well captured by their routine activities. Based on the analysis of the experiment results, we found that most of the clustering errors were caused by the phenomena that even the users who have the same occupations may do similar activities (represented by staying at places with similar personal semantic meanings in this paper) during different time spans. For example, we found that an undergraduate student usually went for class in the morning, while another one usually went to the classroom in the afternoon. Another error source is that users with different occupations may share similar routine activities. For example, we found that one of the graduated student and a company employee stayed at places with similar personal semantic meanings (i.e. lab for the graduated student and office for the company employee) during the same time spans.

We also verified the necessity of discovering OMS (as mentioned in Section 5) for the routine activity similarity calculation. Based on the same experimental setting, we calculated the similarity between each pair of users without discovering the OMS (i.e. taking into account all pairs of reference places when calculating the similarity of routine activities), and conducted the user clustering experiment again. As shown in Fig. 13, both IaDR (72.3% on average) and IrDR (77.5% on average) are significantly decreased. This result means that our user similarity measurement can more appropriately reflect users' long-term activity similarity by using the OMS.

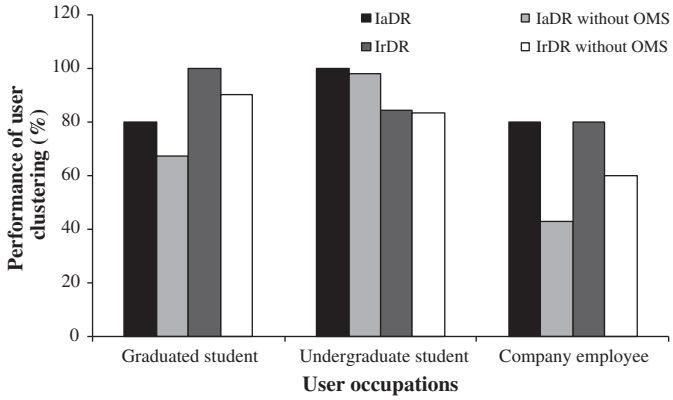

Fig. 13. The performance of user clustering according to their occupations based on the similarity measurement.

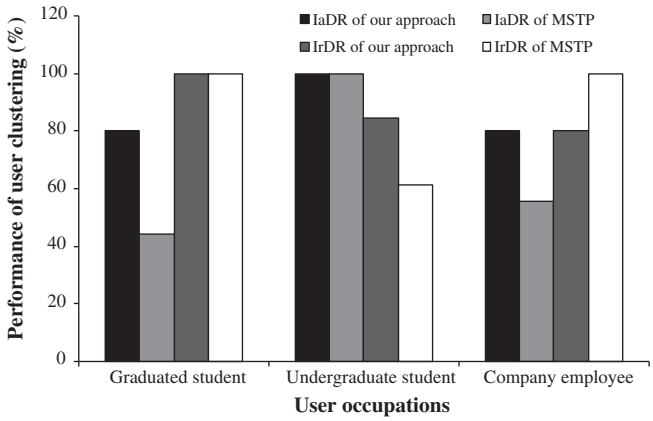

Fig. 14. The performance of user clustering based on different user similarity measurement approaches.

6.3. Comparative experiment

In this section, we compared the capacity of discriminating users with different profiles (i.e. the capacity of capturing users' long-term activity regularities) of our user similarity measurement with that of the existing work [40] (we refer it as MSTP) based on the artificial dataset. To implement the MSTP approach, we first transformed each registered user's 1-day activities to a set of semantic trajectories based on the reference place type input by the users, e.g. the semantic trajectory of the 1-day activity shown in Fig. 2(a) is "home → on the way → lab → on the way → home". Second, we performed the sequential pattern mining algorithm PrefixSpan [32] on each user's semantic trajectory dataset to extract the frequent semantic trajectory patterns. Finally, we calculated user similarity by exploiting the similarity of their maximal semantic trajectory patterns based on the longest common sequences.

Taking the same experimental settings as described in Section 6.2, we used clustering techniques to partition the registered users according to their occupations based respectively on our approach (i.e. routine activity based similarity) and the MSTP approach (i.e. semantic trajectory based similarity). Fig. 14 compares the user clustering performance of the two different user similarity measure approaches. It can be found from the figure that the MSTP approach has satisfactory performance (IaDR is 71.3% on average, IrDR is 83.9% on average) on user clustering, although still lower than that of our approach. However, the MSTP approach was performed under ideal conditions (i.e. the personal meanings of the reference places are given in advance) in the experiment. Since the accurate personal meanings of the reference places are almost impossible to be obtained, the ability to discriminate users with different profiles of the MSTP approach will be much weaker in practice, whereas our approach can capture the users' long-term activity similarity without a priori knowledge of the reference places.

7. Conclusions and future work

In this paper, we propose an approach to measure user similarity for LBSNs based on GPS trajectory mining. The most important novelty of our user similarity measure approach is that it can capture the similarity of users' long-term activity regularities. To achieve this goal, we propose a framework to extract the routine activities from users' daily GPS trajectories, and calculate the similarity score between users based on their routine activities. The experimental results drawn from both real user trajectories dataset and artificial user activities datasets validate our approach which has the ability to measure users' long-term activity similarity. This kind of context awareness can favor the "computational social science" research by exploiting the activity regularities of individuals and activity similarities of groups of users based on their trajectories.

We envision that future mobile and location-based applications will be highly personalized, and deeply incorporated into each user's daily lives. Understanding users' long-term activity regularities and their long-term activity similarity can help these applications to provide more intelligent and personalized services, e.g. activity plan taking into account users' activity regularities, friends and information recommendation based on users' long-term similarity, etc.

While our approach exploits the GPS trajectories, it is important to extend the routine activity mining framework to make it compatible with other indoor locating infrastructure, and apply our approach to both indoor and outdoor environments. Another important issue is to mine user similarity with sparse and incomplete trajectory data. We consider these as promising future works.

Acknowledgements

This work was supported by the Ministry of Industry and Information Technology of China (No. 2010ZX01042-002-003-001), the Natural Science Foundation of China (Nos. 61202282 and 60703040), the Zhejiang Provincial Natural Science

Foundation of China (No. LY12F02046), Science and Technology Department of Zhejiang Province (Nos. 2007C13019 and 2011C13042).

References

[1] D. Ashbrook, T. Starner, Using GPS to learn significant locations and predict movement across multiple users, Personal and Ubiquitous Computing 7 (5) (2003) 275–286.
[2] A. Bellogín, I. Cantador, P. Castells, A comparative study of heterogeneous item recommendations in social systems, Information Sciences 221 (1) (2013) 142–169.
[3] U. Blanke, B. Schiele, Daily routine recognition through activity spotting, in: Proceedings of International Symposium on Location and Context Awareness, 2009, pp. 192–206.
[4] J. Bobadilla, A. Hernando, F. Ortega, A. Gutiérrez, Collaborative filtering based on significances, Information Sciences 185 (1) (2012) 1–17.
[5] V. Bogorny, B. Kuijpers, L.O. Alvares, ST-DMQL: a semantic trajectory data mining query language, International Journal of Geographical Information Science 23 (10) (2009) 1245–1276.
[6] L. Chen, M.Q. Lv, G.C. Chen, A system for destination and future route prediction based on trajectory mining, Pervasive and Mobile Computing 6 (6) (2010) 657–676.
[7] L. Chen, M.Q. Lv, Q. Ye, G.C. Chen, J. Woodward, A personal route prediction system based on trajectory data mining, Information Sciences 181 (7) (2011) 1264–1284.
[8] C.Y. Chow, J. Bao, M.F. Mokbel, Towards location-based social networking services, in: Proceedings of the ACM SIGSPATIAL International Workshop on Location Based, Social Networks, 2010, pp. 31–38.
[9] P. De Meo, A. Nocera, G. Terracina, D. Ursino, Recommendation of similar users, resources and social networks in a social internetworking scenario, Information Sciences 181 (7) (2011) 1285–1305.
[10] J.C. Dunn, Well separated clusters and optimal fuzzy partitions, Cybernetics and Systems 4 (1) (1974) 95–104.
[11] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, Personal and Ubiquitous Computing 10 (4) (2006) 255–268.
[12] K. Farrahi, D. Gatica-Perez, Daily routine classification from mobile phone data, in: Proceedings of International Workshop on Machine Learning for Multimodal, Interaction, 2008, pp. 173–184.
[13] K. Farrahi, D. Gatica-Perez, Discovering routines from large-scale human locations using probabilistic topic models, ACM Transactions on Intelligent Systems and Technology 2 (1) (2011).
[14] S.P. Ferrando, E. Onaindia, Context-aware multi-agent planning in intelligent environments, Information Sciences (2012).
[15] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, R. Trasarti, Unveiling the complexity of human mobility by querying and mining massive trajectory data, The VLDB Journal 20 (5) (2011) 695–719.
[16] M.C. González, C.A. Hidalgo, A.L. Barabási, Understanding individual human mobility patterns, Nature 453 (7196) (2008) 779–782.
[17] I. Guy, M. Jacovi, A. Perer, I. Ronen, E. Uziel, Same places, same things, same people? Mining user similarity on social media, in: Proceedings of the ACM Conference on Computer Supported Cooperative, Work, 2010, pp. 41–50.
[18] T. Horozov, N. Narasimhan, V. Vasudevan, Using location for personalized POI recommendations in mobile environments, in: Proceedings of the International Symposium on Applications on Internet, 2006, pp. 124–129.
[19] W.J. Hsu, D. Dutta, A. Helmy, Mining behavioral groups in large wireless LANs, in: Proceedings of ACM International Conference on Mobile Computing and Networking, 2007, pp. 338–341.
[20] C.C. Hung, W.C. Peng, W.C. Lee, Clustering and aggregating clues of trajectories for mining trajectory patterns and routes, The VLDB Journal (2011).
[21] H. Jeung, M.L. Yiu, X. Zhou, C.S. Jensen, Path prediction and predictive range querying in road network databases, The VLDB Journal 19 (4) (2010) 585–602.
[22] J.H. Kang, W. Welbourne, B. Stewart, G. Borriello, Extracting places from traces of locations, ACM SIGMOBILE Mobile Computing and Communications Review 9 (3) (2005) 58–68.
[23] D.H. Kim, J. Hightower, R. Govindan, D. Estrin, Discovering semantically meaningful places from pervasive RF-beacons, in: Proceedings of International Conference on Ubiquitous, Computing, 2009, pp. 21–30.
[24] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.L. Barabási, D. Brewer, Computational social science, Science 323 (5915) (2009) 721–723.
[25] A.J.T. Lee, Y.A. Chen, W.C. Ip, Mining frequent trajectory patterns in spatial-temporal databases, Information Sciences 179 (13) (2009) 2218–2231.
[26] M.J. Lee, C.W. Chung, A user similarity calculation based on the location for social network services, in: Proceedings of the International Conference on Database Systems for Advanced Applications, 2011, pp. 38–52.
[27] Z. Li, J. Han, B. Ding, R. Kays, P. Nye, Mining periodic behaviors of object movements for animal and biological sustainability studies, Data Mining and Knowledge Discovery 24 (2) (2011) 355–386.
[28] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, W.Y. Ma, Mining user similarity based on location history, in: Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic, Information Systems, 2008.
[29] C. Lin, B. Jin, Z. Long, H. Chen, On context-aware distributed event dissemination, Personal and Ubiquitous Computing 15 (3) (2011) 305–314.
[30] E.H.C. Lu, V.S. Tseng, P.S. Yu, Mining cluster-based temporal mobile sequential patterns in location-based service environments, IEEE Transactions on Knowledge and Data Engineering 23 (6) (2011) 914–927.
[31] R. Monclar, A. Tecla, J. Oliveira, J.M. Souza, MEK: using spatial–temporal information to improve social networks and knowledge dissemination, Information Sciences 179 (15) (2009) 2524–2537.
[32] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.C. Hsu, Mining sequential patterns by pattern-growth: the PrefixSpan approach, IEEE Transactions on Knowledge and Data Engineering 16 (11) (2004) 1424–1440.
[33] D. Quercia, N. Lathia, F. Calabrese, G.D. Lorenzo, J. Crowcroft, Recommending social events from mobile phone location data, in: Proceedings of the IEEE International Conference on Data Mining, 2010, pp. 971–976.
[34] M. Ros, M.P. Cuéllar, M. Delgado, A. Vila, Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows, Information Sciences (2011).
[35] Y. Takeuchi, M. Sugimoto, CityVoyager: an outdoor recommendation system based on user location history, in: Proceedings of the International Conference on Ubiquitous Intelligence and, Computing, 2006, pp. 625–636.
[36] G.S. Thakur, A. Helmy, W.J. Hsu, Similarity analysis and modeling in mobile societies: The missing link, in: Proceedings of the ACM Workshop on Challenged, Networks, 2010, pp. 13–20.
[37] M. Wang, X.S. Hua, R. Hong, J. Tang, G.J. Qi, Y. Song, Unified video annotation via multi-graph learning, IEEE Transactions on Circuits and Systems for Video Technology 19 (5) (2009) 733–746.
[38] M. Wang, X.S. Hua, J. Tang, R. Hong, Beyond distance measurement: constructing neighborhood similarity for video annotation, IEEE Transactions on Multimedia 11 (3) (2009) 465–476.
[39] M. Ye, P. Yin, W.C. Lee, D.L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in: Proceedings of the ACM SIGIR International Conference on Research and Development in, Information Retrieval, 2011, pp. 325–334.
[40] J.J.C. Ying, E.H.C. Lu, W.C. Lee, T.C. Weng, V.S. Tseng, Mining user similarity from semantic trajectories, in: Proceedings of ACM SIGSPATIAL International Workshop on Location Based, Social Networks, 2010, pp. 19–26.
[41] X. Zhao, J. Yuan, R. Hong, M. Wang, Z. Li, T. Chua, On video recommendation over social network, in: Proceedings of the International Conference on Advances in Multimedia Modeling, 2012, pp. 149–160.

[42] Y. Zheng, L. Zhang, Z. Ma, X. Xie, W.Y. Ma, Recommending friends and locations based on individual location history, ACM Transaction on the Web 5 (1) (2011) 1–44.
[43] V.W. Zheng, Y. Zheng, X. Xie, Q. Yang, Towards mobile intelligence: learning from GPS history data for collaborative recommendation, Artificial Intelligence 184–185 (2012) 17–37.
[44] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, L. Terveen, Discovering personally meaningful places: an interactive clustering approach, ACM Transactions on Information Systems 25 (3) (2007).
[45] Y. Zhu, R.Y. Shtykh, Q. Jin, A human-centric framework for context-aware flowable services in cloud computing environments, Information Sciences (2012).