Constructing and Comparing User Mobility Profiles for Location-based Services

Xihui Chen*
Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg

Jun Pang
Computer Science and Communications, University of Luxembourg

Ran Xue
University of Luxembourg & Shandong University

ABSTRACT

With the increasing availability of location-acquisition technologies, we have better access to collections of large spatio-temporal datasets. This brings new opportunities to location-based services (LBS), especially when knowledge of users' movement behaviour (i.e., mobility profiles) can be extracted from such datasets. For instance, in social networks, friends can be recommended according to similarity scores between user mobility profiles.

In this paper, we propose a new approach to construct users' mobility profiles and calculate the mobility similarities between users. We model mobility profiles as traces of places that users frequently visit and use frequent sequential pattern mining technologies to extract them. To compare users' mobility profiles, we first discuss the weakness of a similarity measurement in the literature and then propose our new measurement. We evaluate our work using a real-life dataset published by Microsoft Research Asia and the experimental results show that our approach outperforms the existing works on different aspects.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: General—*User profiles and alert services protection*; H.2.8 [**Database Management**]: Data Applications—*Data mining, Spatial databases and GIS*

General Terms

Algorithms, measurement

Keywords

Mobility profiles, pattern mining, similarity

1. INTRODUCTION

In the past decades, there has been a large increase on the popularity and diversity of devices capable of location-acquisition. With this increase, a new type of virtual communities – location-based social networks (LBSN) have emerged, e.g., Foursquare and Bikely. In LBSNs people can share their outdoor activities with friends, monitor travel distance and duration, or even upload photos tagging a route. By leveraging user similarity in terms of interest revealed by their whereabouts, LBSNs can provide better services. For instance, the most basic service friend recommendation can be easily enhanced or improved by ranking users according to such similarities. Furthermore, once user mobility profiles are available, we can recommend not only friends but also new places from the ones that similar users are interested in.

Due to the popularity of recommendation services, mobility profile construction and comparison have been attracting a lot of attention in the literature. One common interpretation of mobility profiles is users' regular moving behaviour in terms of *space* and *time*. More specifically, space refers to the places frequently visited by users and time indicates the typical transition time between two consecutive places. For example, a student in Luxembourg takes 10 minutes every day to transfer from the central train station to Hamilius, the central bus stop from which he spends another 15 minutes to get to the campus Kirchberg. This daily routine can be described as one of the student's regular movements:

$$Central\ Train\ Station \xrightarrow{10\ min} Hamilius \xrightarrow{15\ min} Kirchberg.$$

This interpretation leads to many mobility profile construction methods and mobility similarity measurements.

Related work. We classify the related works in the literature into two groups – one on mobility profile construction and the other on user similarity computation for recommender systems.

Giannotti et al. [5] introduce the concept of trajectory patterns to represent a set of users' trajectories including the same sequence of places referred to as regions of interest (RoI), with similar transition time. They reduce the problem of trajectory pattern mining to the typical *frequent sequential pattern* (FSP) problem [1]. Many algorithms have been proposed for FSP among which *PrefixSpan* [8] is one of the most efficient and widely used algorithms. Giannotti et al. [4] extend *PrefixSpan* to mine sequences with typical temporal annotations (TAS). Trajectory patterns are defined as an extension of TASs in [5]. The elements in a pattern are no longer events but RoIs that a user often visits. RoIs are detected by merging dense spatial cells, which are contained by many trajectories. By transforming GPS points into RoIs, trajectory pattern mining is reduced to TAS mining. Through experiments, we find that the RoIs generated by this method [4] cannot be used as a precise representation of users' meaningful areas due to their large area.

Zheng et al. propose and implement a personalised friend and location recommender system called *GeoLife* [14]. GPS points are grouped into *stay points* which stand for the places where users hang out and spend a certain amount of time. A density-based al-

*Supported by the National Research Fund, Luxembourg (SECLOC 794361).

SAC'13 March 18-22, 2013, Coimbra, Portugal.

gorithm is then used to hierarchically cluster the stay points into RoIs. Once all uses' trajectories are transformed into sequences of RoIs, the *longest common subsequence* (LCS) is extracted for any pair of users and used to measure their similarities. Xiao et al. [9] propose a similar approach but make use of the semantics of places. They model a GPS trajectory with a sequence of semantic locations, such as museums and restaurants. In this way, the semantic meanings of RoIs are considered. However, both of the two approaches [14, 9] work on trajectories, which may contain some places rarely visited. These places will enlarge the area of ROIs as outliers during the clustering process. Ying et al. also propose an approach to recommend friends based on users' semantic trajectories but on the level of trajectory patterns [11]. They use *PrefixSpan* to mine frequent semantic trajectory patterns and define a measurement called *maximal semantic trajectory pattern similarity* (MSTP-similarity) in order to compute the similarity between two users. However, Ying et al. ignore transition time and their measurement encounters a problem when comparing two identical uses (see Sect. 4).

Our contributions. In this paper, we propose a new approach to construct user mobility profiles and calculate the similarity scores between users based on their mobility profiles. First, we improve the mobility profile construction procedure by Giannotti et al. [5] to find more precise meaningful places (RoIs) of users. The use of trajectory patterns in our approach allows us to focus on users' regular movements. Second, we show that the user similarity measurement proposed by Ying et al. [9] is incorrect and we define a new measurement. Since we also consider transition time, our measurement is more accurate to compute user similarities.

We make use of a real-life dataset published by Microsoft Research Asia in our experiments. The results show that our profile construction process outperforms the existing works in the literature and our new similarity measurement is quite effective.

2. PRELIMINARIES

We refer to a GPS point as a location on the earth and denote it by $(\ell at, \ell ng)$ indicating the latitude and longitude. A region represents an area and can be considered as a set of GPS points. We use a region of interest (RoI) to denote a meaningful region for a user where he has performed an activity. For the student in Luxembourg in Sect. 1, the central train station and Hamilius are two RoIs. GPS trajectories are paths that moving objects follow through space in certain time periods. They record users' outdoor movements by logging time-stamped geographic points.

DEFINITION 1 (GPS TRAJECTORY). *A GPS trajectory is a sequence of chronologically ordered spatio-temporal points, i.e., $(p_0, \ldots, p_n)$ where $p_i = \langle \ell at_i, \ell ng_i, t_i \rangle$ $(0 \leq i \leq n)$ with t_i as a time point and $(\ell at_i, \ell ng_i)$ as a GPS point.*

A stay point stands for a geographic region, where a user stays over a time threshold θ_t and within a distance threshold θ_d [7]. Let $dis(p, p')$ be the Euclidean distance between two points p and p'. The definition of stay points can be formulated as follows:

DEFINITION 2 (STAY POINT). *A stay point s of a given trajectory $T = (p_0, \ldots, p_n)$ corresponds to a subsequence T' of T. If $T' = (p_j, \ldots, p_{j+m})$ where $\forall 0 < x \leq m$, $dis(p_j, p_{j+x}) \leq \theta_d$, $dis(p_j, p_{j+m+1}) > \theta_d$ and $t_{j+m} - t_j \geq \theta_t$, then we have $s = (\ell at, \ell ng, t_a, t_e)$ where $\ell at = \frac{\sum_{x=0}^{m} \ell at_{j+x}}{m+1}$, $\ell ng = \frac{\sum_{x=0}^{m} \ell ng_{j+x}}{m+1}$ stand for the average latitude and longitude of the points in T', $t_a = t_j$ is the arriving time at s and $t_e = t_{j+m}$ is the exiting time.*

From the trajectory dataset published by Microsoft [15], we have observed that a user tends to start and end a trajectory with one of his meaningful places, e.g., home or offices. Therefore, besides the stay points representing regions, we also consider the first and the last point of a trajectory as stay points. However, if they are close to other region representative stay points and the distance is smaller than a threshold, i.e., θ_m, we merge them into one stay point, which is the middle point of the line connecting them.

A *trajectory pattern* of a user represents one of the user's regular mobility trace. It is usually denoted as sequences of RoIs with transition time annotated [5].

DEFINITION 3 (TRAJECTORY PATTERN). *A trajectory pattern (T-pattern for short) is a pair (S, A) where $S = (R_0, \ldots, R_n)$ $(n \geq 0)$ is a sequence of RoIs and $A = (\alpha_1, \ldots, \alpha_n)$ is the temporal annotation of the sequence. It can be represented as $(S, A) = R_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} R_n$.*

If a user sequentially travels all the RoIs of a T-pattern in a trajectory and spends similar time to transfer between regions, then we say this pattern is spatio-temporally contained in this trajectory.

DEFINITION 4 (SPATIO-TEMPORAL CONTAINMENT). *Given a trajectory T, time tolerance τ and a T-pattern $(S, A) = R_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} R_n$, we say that (S, A) is spatio-temporally contained in T (denoted by $(S, A) \preceq_\tau T$) if and only if there exists a subsequence of T, i.e., $T' = (\langle x'_0, y'_0, t'_0 \rangle, \ldots, \langle x'_n, y'_n, t'_n \rangle)$ such that:*

$$\forall 0 \leq i \leq n, \langle x'_i, y'_i \rangle \in R_i \text{ and } |\alpha_i - \alpha'_i| \leq \tau \text{ where } \alpha'_i = t'_i - t'_{i-1}.$$

When T-pattern (S, A) is spatio-temporally contained in a trajectory, we say that the T-pattern has an occurrence. A T-pattern usually has multiple occurrences in a spatio-temporal dataset. We use *support value* ($support_\tau^{\mathcal{T}}(S, A)$) to represent the percentage of the trajectories containing (S, A) in dataset $\mathcal{T}$ when the time interval tolerance is set to τ. If the support value of a T-pattern is larger than a given *minimum support*, then we call the pattern a *frequent T-pattern*.

The problem of *trajectory pattern mining* is how to find frequent T-patterns in a given spatio-temporal dataset. The result is a set of T-patterns, called *frequent pattern set*.

DEFINITION 5 (FREQUENT PATTERN SET). *For a set of trajectories $\mathcal{T}$, time tolerance τ and a minimum support value σ, the (τ, σ)-frequent pattern set of $\mathcal{T}$ is*

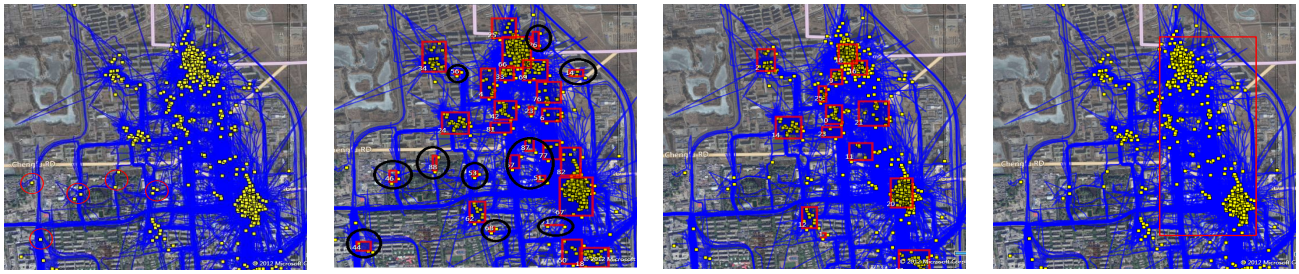
$$PS_{\tau,\sigma}^{\mathcal{T}} = \{(S, A) \,|\, support_\tau^{\mathcal{T}}(S, A) \geq \sigma\}.$$

A mobility profile describes a user's regular movement, i.e., the traces of places that the user often visits. Such traces have a natural correspondence with frequent T-patterns when the places are interpreted as RoIs. Moreover, as a user's movement is represented by a collection of trajectories, we can model his mobility profile by the frequent pattern set of his trajectories and use trajectory pattern mining techniques to construct it. Let $\mathcal{T}_u$ be the trajectories taken by user u in a dataset $\mathcal{T}$. We call $PS_{\tau,\sigma}^{\mathcal{T}_u}$ the mobility profile of u. In the following discussion, we use PS^u to denote u's mobility profile for short by assuming τ and σ have been defined and $\mathcal{T}_u$ is clear from the context. With the same rule applied, the support value $support_\tau^{\mathcal{T}_u}(S, A)$ is denoted by $support^u(S, A)$ instead.

3. CONSTRUCTING MOBILITY PROFILES

Given user u's trajectories $\mathcal{T}_u$, we construct his mobility profile through the following four sequential steps:

1. Compute the stay points of each trajectory in $\mathcal{T}_u$ using stay point detection & merging algorithm;



(a) Trajectories & stay points. (b) RoIs with outliers. (c) RoIs without outliers. (d) RoIs by the T-pattern miner.

Figure 1: An example of RoI construction.

2. Remove the noisy stay points and apply a hierarchical clustering algorithm on remaining stay points to generate RoIs;
3. Transform the GPS trajectories in $\mathcal{T}_u$ into RoI trajectories using the RoIs computed at step 2;
4. Use the trajectory pattern miner [5] to compute frequent trajectory patterns from the RoI trajectories obtained at step 3.

The first two steps are about constructing RoIs. Fig. 1 shows an example of the RoI construction for a user. We use the blue lines to depict the user's trajectories. The extraction of stay points eliminates the points collected during transition between places and enables us to only focus on users' meaningful places. Fig. 1a displays the extracted stay points in yellow dots. Since GPS trajectories can be different even if they are collected from an identical route, the stay points vary from trajectory to trajectory. However, from Fig. 1a, we can observe that the stay points in an RoI are usually close to each other. Thus we can apply clustering algorithms to automatically detect nearby stay points. In this paper, we use the minimum rectangular area which covers a cluster of stay points to represent an RoI. Another observation is that there exist outlying stay points which users visit occasionally (see the points in red circles in Fig. 1a). Such points degrade the quality of generated RoIs, e.g., enlarging area or computing infrequent places [6]. We introduce LOF (Local Outlier Factor) [3] to measure the extent of each stay point to which it is isolated from others. Based on the results, we discard a certain percentage (called *deletion percentage*) of the points with the largest LOF values. Fig. 1b and 1c show the RoIs generated by the hierarchical clustering algorithm with and without outlying points, respectively. It is clear that if the outliers are not removed, a number of small regions are computed and they only have a few points inside. Such regions should not be considered as RoIs where users usually visit. After removing the outliers, we can see that the RoIs have a relative large number of stay points inside and have smaller area compared to the ones in Fig. 1b.

The last two steps focus on mining the frequent pattern set. With the stay points computed in the step 1, we first transform each trajectory into a sequence of stay points. Subsequently we transform this stay point trajectory into an RoI trajectory by replacing any stay point with the RoI where it lies in. In the end, we give the RoI trajectories to the trajectory mining tool [4] and compute the T-patterns that satisfy given minimum support and time tolerance.

With regards to the RoI construction, there exist other methods in the literature. Zheng et al. [14] use a density-based clustering algorithm OPTICS [2] to compute RoIs from stay points but without removing outliers. We have illustrated the shortcoming of this method by Fig. 1b and 1c. In the trajectory pattern miner, Giannotti et al. [5] also implement an RoI construction algorithm. Space is divided into a grid, each cell of which is assigned a density value according to the number of GPS trajectories passing through. Afterwards, a region growing procedure starts from dense cells by merging nearby dense cells. The procedure continues until the average density of the region is below a threshold. We do not use this method because: (1) the density measures the frequency of a user passing by a cell but not staying in the cell; (2) the popularity of an RoI is determined only by density and stay time is ignored. Therefore, the generated RoIs tend to have large areas, particularly when users own large numbers of fine-grained trajectories. Fig. 1d shows the RoIs computed by the tool, which covers almost the whole area. In the following we present an example of mobility profiles.

EXAMPLE 1. *Suppose after trajectory transformation (i.e., step 3), user u has three RoI trajectories:*

$$T_1: \ A \xrightarrow{1} B \xrightarrow{4} D \xrightarrow{2} C \qquad T_2: \ A \xrightarrow{2} B \xrightarrow{5} E \xrightarrow{1} C$$
$$T_3: \ A \xrightarrow{3} B \xrightarrow{7} F \xrightarrow{1} C.$$

For the sake of being concise, we label the transition time between RoIs explicitly. Assume the minimum support $\sigma = 0.5$ and time tolerance $\tau = 2$.

We find that a sequence of RoIs may correspond to infinitely many T-patterns. For instance, when $0 \leq \alpha \leq 4$, $A \xrightarrow{\alpha} B$ always has two occurrences and $support^u(A \xrightarrow{\alpha} B) = \frac{2}{3} > 0.5$. For these T-patterns, we can use the interval $[0, 4]$ to represent the transition time between A and B, and thus $A \xrightarrow{[0,4]} B$ is the set of all T-patterns with the sequence of RoIs (A, B) and transition time between 0 and 4. Thus, in Exa. 1 the user u's mobility profile can be represented as follows:

$$PS^u = \{A, B, C, A \xrightarrow{[0,4]} B, A \xrightarrow{[6,12]} C, B \xrightarrow{[6,8]} C, A \xrightarrow{[0,4]} B \xrightarrow{[6,8]} C\}.$$

4. COMPARING MOBILITY PROFILES

In this section, we first focus on comparing mobility profiles without considering transition time and then give our method to take transition time into account.

When transition time is ignored, given a user u's mobility profile PS^u we get a simpler mobility profile (called *sequence pattern set*) $\overline{PS}^u = \{S \mid \exists (S, A) \in PS^u\}$. The support value of a sequence pattern S ($support^u(S)$) equals to the support value of any $(S, A) \in PS^u$ when τ is set to $+\infty$, i.e., $support^{\mathcal{T}_u}_{+\infty}(S, A)$. In the sequence pattern set, we have all the frequent sequences of RoIs. In Exa. 1, $\overline{PS}^u = \{A, B, C, A \rightarrow B, A \rightarrow C, B \rightarrow C, A \rightarrow B \rightarrow C\}$. We notice that some of these sequence patterns have duplicated information. For instance, in Exa. 1, if we know $A \rightarrow B$ is a sequence pattern, then A and B are also his sequence patterns.

If we compare two users' mobility profiles using the whole sequence pattern set, some behaviour will be considered more than once. Therefore, we use the *maximal pattern set* to compare users' mobility profiles which consist of only the patterns that are not contained in other patterns.

Given $P = (R_0, \ldots, R_n)$ and $Q = (R'_0, \ldots, R'_m)$, we call Q a subsequence of P (denoted by $Q \sqsubseteq P$) if there exists $j_1 < \ldots < j_m$ such that $R'_i = R_{j_i}$ $(0 \leq i \leq m)$. We define the maximal sequence pattern set as follows:

DEFINITION 6 (MAXIMAL SEQUENCE PATTERN SET). *Given user u's sequence pattern set $\overline{PS}^u$, the maximal sequence pattern set of u is*

$$M(\overline{PS}^u) = \{P \in \overline{PS}^u \,|\, \nexists P' \in \overline{PS}^u (P \sqsubseteq P')\}.$$

In the following discussion, we first give a brief introduction to a similarity measurement [11] in the literature and show that it is incorrect through examples. Afterwards, we define our own measurement and extend it to take transition time into account.

4.1 A similarity measurement by Ying et al.

Ying et al. define a user similarity measurement [11] according to semantic trajectory patterns. By 'semantics' they mean the function of the places, such as parks, schools or hospitals. For instance, the trajectory pattern of the student in Luxembourg in Sect. 1 corresponds to semantic pattern $train\ station \rightarrow bus\ stop \rightarrow school$. To compute two users' similarity, they first propose a similarity measurement between maximal semantic pattens (MSTP-similarity). Then based on pattern similarity, users' profiles which consist of all maximal semantic patterns are compared.

Although maximal semantic patterns are defined on the level of location semantics, they have the same form as sequence patterns syntactically. So we can apply it on maximal sequence patterns as well. Given two maximal sequence patterns, the argument is that the more similar they are, the longer common part they share. We use *longest common sequences* (LCS) to represent the longest common part. For example, sequence patterns $P = A \rightarrow E \rightarrow B \rightarrow H \rightarrow D$ and $Q = E \rightarrow A \rightarrow B \rightarrow D$ have two longest common sequences $E \rightarrow B \rightarrow D$ and $A \rightarrow B \rightarrow D$. They form the set of the longest common sequences of P and Q, denoted by $lcs(P, Q)$. Let $lenLCS(P, Q)$ be the length of the longest common sequences in $lcs(P, Q)$ and $len(P)$ be the length of P. According to the weighted average trajectory pattern similarity in [11], the similarity between P and Q is calculated as follows:

$$sim(P, Q) = \frac{2 \cdot lenLCS(P, Q)}{len(P) + len(Q)}.$$

In the previous example, with $lenLCS(P, Q) = 3$ and $len(P) = len(Q) = 4$, $sim(P, Q) = \frac{2 \cdot 3}{4+4} = 0.75$.

The similarity between users is computed based on MSTP-similarity. The idea is to compute the weighted average of all possible similarities between maximal patterns. In this paper, we use support values of patterns to construct the weighting function. Given two users u and u', the similarity between them is calculated as follows:

$$sim(u, u') = \frac{\sum_{P_i \in M(\overline{PS}^u)} \sum_{Q_j \in M(\overline{PS}^{u'})} w(P_i, Q_j) \cdot sim(P_i, Q_j)}{\sum_{P_i \in M(\overline{PS}^u)} \sum_{Q_j \in M(\overline{PS}^{u'})} w(P_i, Q_j)}$$

where $w(P_i, Q_j) = \frac{support^u(P_i) + support^{u'}(Q_j)}{2}$.

We find that the similarity between users calculated by this measurement is counter-intuitive and inconsistent with common sense in certain cases. We illustrate the inconsistency through the following example.

EXAMPLE 2. *Given three sequence patterns $P_1 = A \rightarrow B$, $P_2 = C \rightarrow D$ and $P_3 = E \rightarrow F$ and four users u, u_1, u_2 and u_3, we want to calculate the similarity of u to other three users. The user u has the same maximal sequence pattern set as u_3, which is $\{P_1, P_2, P_3\}$ while the maximal sequence pattern sets of u_1 and u_2 are $\{P_1\}$ and $\{P_1, P_2\}$, respectively. The pattern similarity between any two patterns is shown in Tab. 1. For the sake of simplicity, we assume patterns have the same support value 0.2.*

Table 1: Example of similarity computation with Ying et al.'s method.

		$M(\overline{PS}^{u_1})$	$M(\overline{PS}^{u_2})$		$M(\overline{PS}^{u_3})$		
		P_1	P_1	P_2	P_1	P_2	P_3
$M(\overline{PS}^u)$	P_1	**1**	**1**	0	**1**	0	0
	P_2	0	0	**1**	0	**1**	0
	P_3	0	0	0	0	0	**1**

We see that u shares one common pattern with u_1, two with u_2 and three with u_3. So intuitively, the similarity of u to u_1 should be the smallest and the similarity between u and u_3 should be 1 as they are identical. However, with the measurement, we get

$$sim(u, u_1) = \frac{0.2}{0.2 \times 3} = 0.33; \quad sim(u, u_2) = \frac{0.2 \times 2}{0.2 \times 6} = 0.33;$$
$$sim(u, u_3) = \frac{0.2 \times 3}{0.2 \times 9} = 0.33.$$

The results say that u is the same similar to the three users, which is clearly not what we expect.

4.2 Our method

From Exa. 2, we learn that the weighted average of pattern similarities is not the proper measurement for user similarity. Our idea is to only consider the most similar pattern of each maximal sequence pattern instead of all the maximal patterns of another user.

Given two users u and u', we use function $\psi_{u,u'} : M(\overline{PS}^u) \rightarrow M(\overline{PS}^{u'})$ to map a maximal pattern of u to the most similar maximal pattern in $M(\overline{PS}^{u'})$. Specifically, for each $P_i \in M(\overline{PS}^u)$,

$$\psi_{u,u'}(P_i) = \underset{Q_j \in M(\overline{PS}^{u'})}{\arg\max} \; sim(P_i, Q_j) \cdot w(P_i, Q_j).$$

Then for each user, we compute his *relative similarity* to the other one. The relative similarity of u to u' is

$$sim(u \,|\, u') = \frac{\sum_{P_i \in M(\overline{PS}^u)} sim(P_i, \psi_{u,u'}(P_i)) \cdot w(P_i, \psi_{u,u'}(P_i))}{\sum_{P_i \in M(\overline{PS}^u)} w(P_i, \psi_{u,u'}(P_i))}.$$

Similarly, we can also compute the relative similarity of u' to u, i.e., $sim(u' \mid u)$. As the relation of similarity is symmetric, we take the average of the two relative similarity as the similarity score between u and u':

$$sim(u, u') = \frac{sim(u \,|\, u') + sim(u' \,|\, u)}{2}.$$

EXAMPLE 3. *Suppose we have the same users as in Exa. 2. We take u_2 as an example to show the calculation process. First, for each pattern of u, we find the corresponding pattern of u_2 with the maximal similarity score, i.e., $\psi_{u,u_2}(P_1) = P_1$, $\psi_{u,u_2}(P_2) = P_2$ $\psi_{u,u_2}(P_3) = P_1/P_2$. So $sim(u \mid u_2) = \frac{0.2+0.2+0}{3 \times 0.2} = 0.67$.*

With the same process, we obtain $sim(u_2 \mid u) = \frac{0.2+0.2}{0.2+0.2} = 1$. *So* $sim(u, u_2) = 0.83$. *Tab. 2 lists calculated similarities. The similarity of* u *with* u_1 *is 0.67 which is the smallest and* u *has the largest similarity to* u_3. *The results are consistent with what we would expect.*

Table 2: Example of similarity computation with our method.

u_i	u_1	u_2	u_3
$sim(u \mid u_i)$	0.33	0.67	1
$sim(u_i \mid u)$	1	1	1
$sim(u, u_i)$	0.67	0.83	1

We can see that our method can clearly distinguish the similarity degrees of u to the three users. More importantly, for identical users, our method always gives a degree of 1.

4.3 Adding transition time

In this section, we take transition time into account in user similarity measurement. The argument is that if two users are similar, in addition to longer common sequences of RoIs, transition time between consecutive RoIs should also be close. Our idea is to update the similarity measurement between maximal patterns by taking comparison of transition time into account. The more similar the given users are, the longer common sequences they share and the closer the transition times on common sequences are.

Suppose that we have two maximal patterns $P \in M(\overline{PS}^u)$ and $Q \in M(\overline{PS}^{u'})$, and one of their longest common sequence $S = (R_0, \ldots, R_n)$ ($S \in lcs(P, Q)$). For any two consecutive RoIs R_{i-1} and R_i ($0 < i \leq n$), the typical transition time of user u between them is the union of all transition time appearing in a T-pattern with S in the user's profile. Let $tranT_S^u(i)$ be the union, then we have $tranT_S^u(i) = \{\alpha_i \mid \exists (S, A) \in PS^u$ s.t. $A = (\alpha_1, \ldots, \alpha_n)\}$. In the same way, we can obtain the corresponding set of transition time of user u', i.e., $tranT_S^{u'}(i)$.

Recall that we can use intervals to represent transition time in Exa. 1. Thus $tranT_S^u(i)$ can be represented as the union of intervals, e.g., $[x_1, y_1] \cup \ldots \cup [x_k, y_k]$. Then we can compute the overlapping transition time of the users and all the occurring transition time by calculating the intersection and union of $tranT_S^u(i)$ and $tranT_S^{u'}(i)$. Suppose $tranT_S^{u'}(i) \cup tranT_S^u(i) = [x_1, y_1] \cup \ldots \cup [x_k, y_k]$ and $tranT_S^{u'}(i) \cap tranT_S^u(i) = [x'_1, y'_1] \cup \ldots \cup [x'_m, y'_m]$. Then we can calculate $ot_S^{u,u'}(i)$, the ratio of overlapping time from R_{i-1} to R_i between u and u', by $\frac{\sum_{1 \leq i \leq m} y'_i - x'_i}{\sum_{1 \leq i \leq k} y_i - x_i}$.

We use the average of all transition time similarities in all longest common sequences to measure the transition time similarity between two maximal patterns, called *time-overlap-fraction*.

DEFINITION 7 (TIME-OVERLAP-FRACTION). *Let* P *and* Q *be two maximal sequence patterns of* u *and* u', *respectively. Then the time-overlap-fraction of* P *and* Q, *denoted by* $tof(P, Q)$ *can be calculated as:*

$$tof(P, Q) = \frac{\sum_{S \in lcs(P,Q)} \sum_{i=1}^{len(S)-1} ot_S^{u,u'}(i)}{|\, lcs(P, Q)\,| \cdot (lenLCS(P, Q) - 1)}.$$

The similarity of P and Q can thus be calculated as follows:

$$sim(P, Q) = \frac{2 \cdot lenLCS(P, Q)}{len(P) + len(Q)} \cdot tof(P, Q).$$

5. EXPERIMENTS

We use the GPS trajectory dataset collected in *Geolife* project of Microsoft Research Aisa [13] to evaluate our work.

5.1 Setting

The dataset. The Geolife dataset consists of 17,621 trajectories from 178 users in a period of over four years (from April 2007 to October 2011). The trajectories cover a total length of 1,251,654 km and a total duration of 48,203 hours. Moreover, the GPS positions are collected with a high frequency. Over 90% of the positions are recorded less than every 5 seconds with a distance less than 10 meters from their previous positions. The trajectories also reflect a diverse collection of users' outdoor movements, not restricted to only daily activities. Almost all trajectories are located in Beijing (China) although the GPS positions are distributed in over 30 cities.

The trajectory dataset does not provide users' personal information such as gender or affiliation due to privacy protection. So we have no access to the ground truth about the similarity between users. Although Zheng et al. construct the volunteers' similarity which works as the ground truth in [7], we cannot obtain and use it because of the legal rules about publishing data in Microsoft [12]. In order to validate our similarity measurement, we choose two users who have a large number of trajectories and divide them into new users. One is user 126, who is divided into two users (i.e., 126, 126^*) while the other one is user 151, divided into three users (i.e., 151 ,151^*,$151^\#$). Intuitively, the new users should preserve the original users' behaviour and thus have a higher degree of similarity. In addition, we choose another five users from the rest of volunteers. In this way, we have a testing dataset with 10 users and 55 different pair of users for similarity computation. In our experiments, a user's movement in a day forms a trajectory. For the ten chosen users, each has about 300 trajectories on average containing over 1,250 stay points.

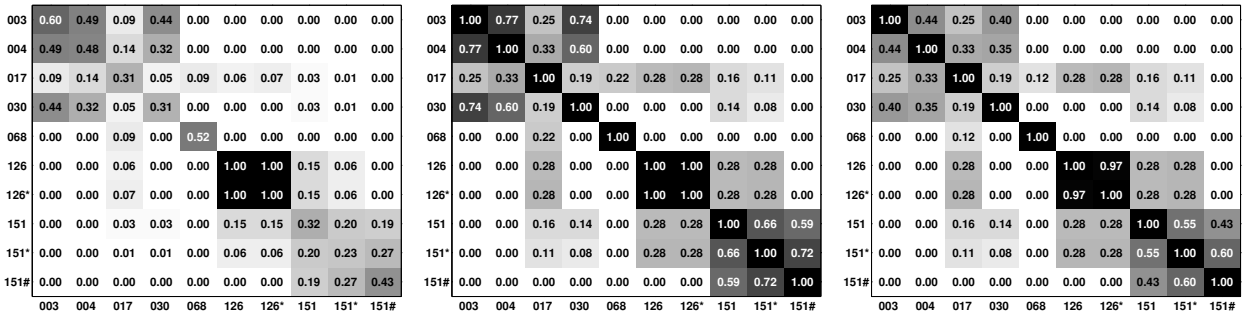
Implementation. We use the bottom-up (also called *agglomerative*) hierarchical clustering algorithm to cluster stay points. Compared to other clustering algorithms, it allows us to customise the termination condition using the shortest distance between clusters and does not need to fix the number of clusters beforehand like k-means. The clustering process stops once the shortest distance between any two clusters is larger than a threshold, i.e., δ. This parameter also determines the longest diagonal of generated RoIs.

To compare two users, we first merge their trajectories and construct their common RoIs. Then we transform each user's trajectories using these RoIs and generate his mobility profile.

All related parameters need to be fixed before performing our evaluation. The principle of setting their values is to enforce a good quality of T-patterns. Due to the page limit, we refer readers to [10] for the parameter settings.

5.2 Experimental results

In Fig. 2 we show the similarity between any pair of the 10 users calculated by three similarity measurements. We use different grey levels to distinguish the similarity scores between two users. The darker the cell is, the more similar the corresponding two users are. Fig. 2a and 2b show the user similarities computed by the measurement of Ying et al. [11] (MSTP) and our measurement without transition time (MTP), respectively. The diagonal cells correspond to the similarity of a user to himself which is expected to be one. However, from Fig. 2a it is clear that MSTP fails to capture this except for the users who have only one maximal sequence patterns, i.e., user 126 and 126^*. (These two users originate from one volunteer, their maximal sequence patterns are the same. This leads to a similarity score of 1.00 between them.)



(a) User similarities computed by MSTP. (b) User similarities computed by MTP. (c) User similarities computed by MTP+TOF.

Figure 2: User similarities by three methods.

Recall that user 151, 151^* and $151^\#$ are derived from the same volunteer. They should have large common mobility patterns in their mobility profiles and thus have high similarity scores. However, measured by MSTP, the similarity scores are only about 0.20. On the contrary, our method MTP successfully finds the similarity of these three users. The average similarity score is over 0.65.

For the users who do not share any common sequences, both of the two measurements give zero indicating their dissimilarity. From the above analysis, we can conclude that our measurement can give a better similarity comparison between users' mobility profiles, especially when they have multiple maximal sequence patterns.

Fig. 2c shows the similarity between users when we add time overlapping fraction into our measurements (MTP+TOF). Compared to the values in Fig. 2b, we find that the similarity values between users in general become smaller. That is because the difference between transition time discounts the similarities. Even for users 126 and 126^* who have the same maximal sequence pattern, their similarity decreases from 1.0 to 0.97. We can also see that transition time does help identify similar users. For example, by MTP, the similarity between users 003 and 004 is 0.77 which is larger than the similarity between users 151^* and $151^\#$ (0.72) even they are derived from the same volunteer. With transition time considered, the former similarity decreases to 0.44 while the later still remains over 0.60, mainly because users 003 and 004 do not have similar transition time. Therefore, considering transition time leads us to a more accurate evaluation of user similarity.

6. CONCLUSION

In this paper, we have accomplished two tasks. First, we propose a new method to construct users' mobility patterns. Compared to the existing methods in the literature, our method can detect more accurate RoIs for users. This also ensures the precision of the subsequent user similarity computation. Second, we showed that the user similarity measurement proposed by Ying et al. is incorrect in some cases and we defined a new measurement to fix the problem. As transition time between RoIs are also part of users' mobility patterns, we further took them into account in our user similarity measurement. We validated our work by experiments on a dataset of real-life trajectories. The results show that our measurement and user profile construction are effective.

For future work, we will apply our similarity measurement to location privacy analysis. A high similarity between a given set of anonymised trajectories and a user's mobility profile indicates a high probability for the user to be the owner of the trajectories. It is also interesting to analyse users' similarity according to their trajectory logs posted on social networks.

7. REFERENCES

[1] AGRAWAL, R., AND SRIKANT, R. Mining sequential patterns. In *Proc. ICDE* (1995), IEEE CS, pp. 3 –14.

[2] ANKERST, M., BREUNIG, M. M., KRIEGEL, H.-P., AND SANDER, J. OPTICS: Ordering points to identify the clustering structure. In *Proc. SIGMOD* (1999), ACM Press, pp. 49–60.

[3] BREUNIG, M., KRIEGEL, H.-P., NG, R. T., AND SANDER, J. LOF: Identifying density-based local outliers. In *Proc. SIGMOD* (2000), ACM Press, pp. 93–104.

[4] GIANNOTTI, F., NANNI, M., PEDRESCHI, D., AND PINELLI, F. Mining sequences with temporal annotations. In *Proc. SAC* (2006), ACM Press, pp. 593–597.

[5] GIANNOTTI, F., NANNI, M., PEDRESCHI, D., PINELLI, F., AND AXIAK, M. Trajectory pattern mining. In *Proc. SIGKDD* (2007), ACM Press, pp. 330–339.

[6] JAIN, A. K., AND DUBES, R. C. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[7] LI, Q., ZHENG, Y., XIE, X., CHEN, Y., LIU, W., AND MA, W.-Y. Mining user similarity based on location history. In *Proc. SIGSPATIAL* (2008), ACM Press, pp. 34–43.

[8] PEI, J., HAN, J., MORTAZAVI-ASL, B., WANG, J., PINTO, H., CHEN, Q., DAYAL, U., AND HSU, M.-C. Mining sequential patterns by pattern-growth: the PrefixSpan approach. *IEEE TKDE*, 16 (2004), 1424 – 1440.

[9] XIAO, X., ZHENG, Y., LUO, Q., AND XIE, X. Finding similar users using category-based location history. In *Proc. SIGSPATIAL* (2010), ACM Press, pp. 442–445.

[10] XUE, R. Constructing and comparing user mobility profiles for location-based services. Master's thesis, University of Luxembourg, 2012.

[11] YING, J.-C., LU, H.-C., LEE, W.-C., WENG, T.-C., AND TSENG, S. Mining user similarity from semantic trajectories. In *Proc. SIGSPATIAL* (2010), ACM Press, pp. 19–26.

[12] ZHENG, Y. Personal communication, 2012.

[13] ZHENG, Y., WANG, L., ZHANG, R., XIE, X., AND MA, W.-Y. GeoLife: Managing and understanding your past life over maps. In *Proc. MDM* (2008), IEEE CS, pp. 211–212.

[14] ZHENG, Y., ZHANG, L., MA, Z., XIE, X., AND MA, W.-Y. Recommending friends and locations based on individual location history. *ACM TWEB*, 1 (2011), 1–44.

[15] ZHENG, Y., ZHANG, L., XIE, X., AND MA, W.-Y. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. WWW* (2009), ACM Press, pp. 791–800.