Constructing and Comparing User Mobility Profiles

Xihui Chen, Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
Jun Pang[1], Faculty of Science, Technology and Communication, University of Luxembourg &
Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
Ran Xue, Faculty of Science, Technology and Communication, University of Luxembourg

Nowadays, the accumulation of people's whereabouts due to location-based applications has made it possible to construct their mobility profiles. This access to users' mobility profiles subsequently brings benefits back to location-based applications. For instance, in on-line social networks, friends can be recommended not only based on the similarity between their registered information, e.g., hobbies and professions but also referring to the similarity between their mobility profiles.

In this paper, we propose a new approach to construct and compare users' mobility profiles. First, we improve and apply *frequent sequential pattern mining* technologies to extract the sequences of places that a user frequently visits and use them to model his mobility profile. Second, we present a new method to calculate the similarity between two users using their mobility profiles. More specifically, we identify the weaknesses of a similarity metric in the literature, and propose a new one which not only fixes the weaknesses but also provides more precise and effective similarity estimation. Third, we consider the semantics of spatio-temporal information contained in user mobility profiles and add them into the calculation of user similarity. It enables us to measure users' similarity from different perspectives. Two specific types of semantics are explored in this paper– *location semantics* and *temporal semantics*. Last, we validate our approach by applying it to two real-life datasets collected by Microsoft Research Asia and Yonsei University, respectively. The results show that our approach outperforms the existing works from several aspects.

Categories and Subject Descriptors: H.3.4 [**Systems and Software**]: General—*User profiles and alert services protection*

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: Mobility profiles, pattern mining, spatio-temporal trajectories, recommendation systems, similarity measurement

1. INTRODUCTION

Nowadays, most mobile devices are equipped with location-acquiring chips. This technological progress has not only popularised location-based services (LBSs) but also made it possible to collect users' detailed movements. This availability of users' whereabouts subsequently leads to the construction of their *mobility profiles*, which capture their regular patterns during motion, for instance, the places a user usually visits after work. Many applications can benefit from the access to mobility profiles since they carry valuable information about the users' everyday life such as hobbies and professions, even if sometimes this may breach users' privacy. For instance, frequent visits to book stores indicate the user is probably a fan of reading. On-line social networks are one of the potential beneficiaries. The friend recommendation service, a basic service in social networks, can be upgraded by considering the similarity between the mobility profiles of friend candidates and those of the targeted users. The quality of service is thus improved as more comprehensive information has been taken into account.

[1]To whom correspondence should be addressed. Email: jun.pang@uni.lu.

Xihui Chen is supported by the National Research Fund, Luxembourg (SECLOC 794361).
Authors' address: University of Luxembourg, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg.

DOI : http://dx.doi.org/10.1145/0000000.0000000

Extracting users' movement patterns have attracted numerous research efforts and different models have been proposed in the literature [Monreale et al. 2009; Song et al. 2010; Rhee et al. 2011]. In the setting of computing users' similarity with respect to their interests, we need to consider the places where they usually visit since such repetitive behaviour well reflect users' hobbies. This requirement leads us to use the popular model which describes users' movement patterns as their regular moving behaviour in terms of *space* and *time* called *trajectory patterns* [Monreale et al. 2009]. In particular, space refers to the places frequently visited by users and time indicates the typical transition time between two successive places. For example, a student in Luxembourg usually takes 10 minutes to transfer from the central train station (Gare de Luxembourg) to Hamilius, the central bus stop from which he spends another 15 minutes to get to the university campus Kirchberg. This daily routine can be described as one of the student's regular movements:

$$Gare\ de\ Luxembourg \xrightarrow{10\ min} Hamilius \xrightarrow{15\ min} Kirchberg.$$

The spatio-temporal information contained in users' whereabouts has certain semantics as well. For instance, the functionalities of a location can be interpreted as one type of its semantics, e.g., cinema or university. When location semantics are taken into account, users' regular movements can be transformed into sequences of *location semantics*. For example, the sequence of the above geographical routine can be represented as

$$train\ station \xrightarrow{10\ min} bus\ stop \xrightarrow{15\ min} university.$$

The semantics contained in trajectory patterns allows us to analyse users' mobility behaviour from various perspectives which may fit some services better than others. For instance, the fact that a user regularly goes to cinemas is sufficient to decide whether to recommend films to him, and the recommendation does not rely on whether he goes to exactly the same cinema every time.

Our contributions. In this paper, we propose a new approach to precisely construct user *mobility profiles* and calculate the similarities between users based on their mobility profiles. First, we improve the trajectory pattern mining procedure proposed by Giannotti et al. [Giannotti et al. 2007] by developing a new algorithm to mine users' regions of interest (RoIs). Second, we propose a new similarity metric by extending the one by Ying et al. [Ying et al. 2010]. Specifically, we identify that the metric violates a basic principle in similarity computation and we also find that it cannot provide a precise evaluation of users' similarity in some cases. Our new metric not only fixes these weaknesses but also takes into account the transition time between RoIs. Third, we propose new methods to explore two types of semantics in the comparison of users' mobility profiles – *location semantics* and *temporal semantics* . In the end, we exploit two real-life datasets and conduct extensive experiments to validate our work. The results show that our profile construction procedure outperforms the existing works in the literature and our new similarity measurement is rather effective. This paper is an extended version of the earlier work published in the Proceedings of 28th ACM Symposium on Applied Computing (SAC'13) [Chen et al. 2013].

Structure of the paper. In Section 2, we investigate the state-of-the-art related to our work. Some basic notions are defined in Section 3. In Section 4, we present our method which constructs user mobility profiles through trajectory pattern mining. We discuss the problems with the user similarity measurement defined in [Ying et al. 2010] and present our new metric in Section 5. Location and temporal semantics are discussed in Section 6. We show the effectiveness of our methods by experiments on real-life datasets in Section 7. The paper is concluded with future work in Section 8.

2. RELATED WORK

Due to the development of information technologies, various user behaviour logs have been collected, e.g., visited webs and mobile phone usage. So far, discovering similar users with respect to their repetitive patterns have been well studied for many types of logs. For instance, Ma et al. [Ma et al. 2012] develop an approach to compare users based on their patterns with regard to phone

usage. In this paper, we follow their methodology and identify similar users by comparing their mobility profiles mined from their spatio-temporal mobility records. In the following, we classify the related work in the literature into two groups, focusing on mobility profile construction and user similarity computation, respectively.

Mobility profile construction. Zheng et al. propose and implement a personalised friend and location recommender system called *GeoLife* [Zheng et al. 2011]. 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 algorithm is then applied to hierarchically cluster the stay points into areas referred to as *regions of interest* (RoI). Zheng et al. then transform a user's trajectory into a sequence of RoIs and take it as his mobility profile. This method does not consider users' repetitive movements in trajectories and thus the constructed profiles do not rigorously capture the regularity concealed in users' trajectories.

Giannotti et al. [Giannotti et al. 2007] introduce the concept of trajectory patterns to represent a set of users' trajectories which contains the same sequence of RoIs with similar transition time. In this way, mobility profiles become more concise compared to the ones constructed by the method of Zheng et al. [Zheng et al. 2011]. Giannotti et al. reduce the problem of trajectory pattern mining to the typical *frequent sequential pattern* (FSP) problem [Agrawal and Srikant 1995], which has many algorithms proposed. Among them *PrefixSpan* [Pei et al. 2004] is one of the most efficient and widely used. Giannotti et al. [Giannotti et al. 2006] extend *PrefixSpan* to mine sequences with typical temporal annotations (TAS). Trajectory patterns are then defined as an extension of TASs in [Giannotti et al. 2007]. The elements in a pattern are no longer events but RoIs that a user often visits. The whole area is partitioned into a grid of cells and RoIs are detected by merging the dense ones, which are passed through by a certain number of trajectories.

Through experiments (see Section 4), we find that the RoIs generated by the above two methods [Giannotti et al. 2007] cannot be used as a precise representation of users' meaningful hanging out regions due to their large area. In this paper, we combine the idea of the two works. More specifically, we use trajectory patterns to concisely model users' mobility profiles and improve the idea of clustering stay points to identify more precise RoIs.

Mobility profile comparison. Given the profiles of two users of the form of RoI sequences, Zheng et al. [Zheng et al. 2011] first compute their longest common sequences. Then user similarity is calculated by combining the lengths of the common sequences and the popularity of the RoIs contained in them. Xiao et al. [Xiao et al. 2010] propose a similar approach but make use of the semantics of locations. They transform a trajectory into a semantic trajectory by mapping each location to its location semantics. However, both of the two approaches [Zheng et al. 2011; Xiao et al. 2010] work directly on trajectories, which may contain some places rarely visited. These places will affect the accuracy of the calculated similarity,

Ying et al. propose an approach to recommend friends based on users' semantic trajectories but based on trajectory patterns [Ying et al. 2010]. They use *PrefixSpan* to mine frequent semantic trajectory patterns and define a metric called *maximal semantic trajectory pattern similarity* (MSTP-similarity) to estimate the similarity between two users. However, Ying et al. do not take into account transition time, and more specifically, their metric has a flaw when comparing two identical users (see Section 5 for detailed discussions).

With respect to the semantics contained in trajectories, the work of Ying et al. [Ying et al. 2010] can be improved from at least two aspects. First, the fact that an RoI may correspond to multiple semantics has been taken into account but they ignore the different likelihoods among the semantics to be the real purpose during users' visits to the RoI. Second, Ying et al. [Ying et al. 2010] only consider location semantics of users' mobility records and other types of semantics are also effective in understanding users' mobility. Ma et al. [Ma et al. 2012] take into account multiple types of semantics when normalising users' context logs. For example, 8:00am is replaced by 'morning' and the game Angry Birds is substituted for 'action game'. In this paper, we consider both location semantics and temporal semantics of user mobility profiles.

3. PRELIMINARIES

In this section, we present briefly the backgrounds about trajectory pattern mining and give the definitions of basic concepts. We suppose that users' locations are calculated based on the global positioning system (GPS) due to its popularity and high precision. Our approach can also be extended to other positioning systems which may differ in coordinate formats and location accuracy.

A *GPS point* is referred as a location on the earth and can be denoted by its coordinate, e.g., $(\ell at, \ell on)$ indicating the latitude and longitude, respectively. A *region* is an area and can be interpreted as a set of adjacent GPS points. A *region of interest* (RoI) is a meaningful region where users perform an activity. For the student in Luxembourg in Section 1, the central train station and Hamilius are two of his RoIs. *GPS trajectories* record users' outdoor movements and can be seen as paths that users follow through space in certain periods. They can be defined as a sequence of time-stamped GPS points as the following:

Definition 3.1 (*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 $(\ell at_i, \ell on_i)$ as a GPS point.

Given a trajectory of a user, we can extract the regions where the user lingered according to the amount of time spent in these regions. In [Li et al. 2008], such regions are represented by *stay points*. Intuitively, a stay point stands for a region where a user stays over a time threshold θ_t and his maximum distance to the entering point is bounded by a distance threshold θ_d. Let $dis(p, q)$ be the Euclidean distance between two points p and q. The definition of stay points can then be formulated as follows:

Definition 3.2 (*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

$$\forall 0 < x \leq m,\ dis(p_j, p_{j+x}) \leq \theta_d \ \wedge\ dis(p_j, p_{j+m+1}) > \theta_d \ \wedge\ t_{j+m} - t_j \geq \theta_t,$$

then we have $s = (\ell at, \ell on, t)$ where

$$\ell at = \frac{\sum_{x=0}^{m} \ell at_{j+x}}{m+1}, \quad \ell on = \frac{\sum_{x=0}^{m} \ell on_{j+x}}{m+1}, \text{ and } t = t_j.$$

From the trajectory dataset published by Microsoft [Zheng et al. 2009], we find that the places where a user usually starts and ends a trajectory are also meaningful to them, e.g., home or offices. Therefore, besides the stay points captured by the above definition, we also consider the first and the last point of a trajectory as stay points. However, if such a stay point is close to the adjacent stay point within a distance, i.e., θ_m, we merge them into a single point which is the middle of the straight line between them.

As we mentioned in Section 1, a user's *trajectory pattern* represents one of his regular movement traces. In [Giannotti et al. 2007], it is denoted by a sequence of RoIs annotated by typical transition time between consecutive RoIs.

Definition 3.3 (*Trajectory pattern*). A trajectory pattern (T-pattern for short) is a pair $\langle S, A \rangle$ 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 also be represented as $R_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} R_n$.

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. In other words, the pattern has an occurrence in the trajectory.

Definition 3.4 (*Spatio-temporal containment*). Given a trajectory T, time tolerance τ and a T-pattern $\langle S, A \rangle = R_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} R_n$, we say that $\langle S, A \rangle$ is spatio-temporally contained in T (denoted by $\langle S, A \rangle \preceq_\tau T$) 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:

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

In a spatio-temporal dataset, a T-pattern usually has multiple occurrences. In a dataset $\mathcal{T}$, the frequency of the occurrences of $\langle S, A\rangle$ is quantified by the percentage of the trajectories containing it which is called its *support value*. When the time interval tolerance is set to τ, we denote the support value of $\langle S, A\rangle$ in $\mathcal{T}$ as $sup^{\tau}_{\mathcal{T}}(S, A)$. If the support value of a T-pattern is larger than a given *minimum support*, then the pattern is called a *frequent T-pattern*.

The problem of *trajectory pattern mining* can be formulated as finding all the frequent T-patterns in a given spatio-temporal dataset. The result is called *frequent pattern set*.

Definition 3.5 (*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

$$PS^{\tau,\sigma}_{\mathcal{T}} = \{\langle S, A\rangle \mid sup^{\tau}_{\mathcal{T}}(S, A) \geq \sigma\}.$$

4. CONSTRUCTING MOBILITY PROFILES

We begin with an intuitive example explaining what should be captured by users' mobility profiles.

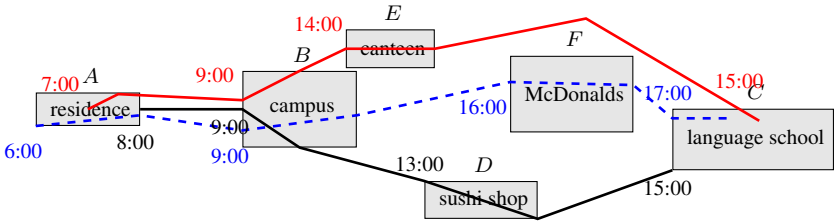


Fig. 1: Three trajectories of a user.

Example 4.1. In Figure 1, we plot three daily trajectories of a college student and annotate them with three distinctive colours. The time stamps of the first GPS points in all PoIs are also labelled on the trajectories. From these trajectories, we can learn that the student visits the same three places (i.e., residence, campus, and language school) every day – he always leaves for the campus from his residence and has language classes after school, although he might pass through other three different places (i.e., canteen, McDonalds, and sushi shop) before arriving at the language school.

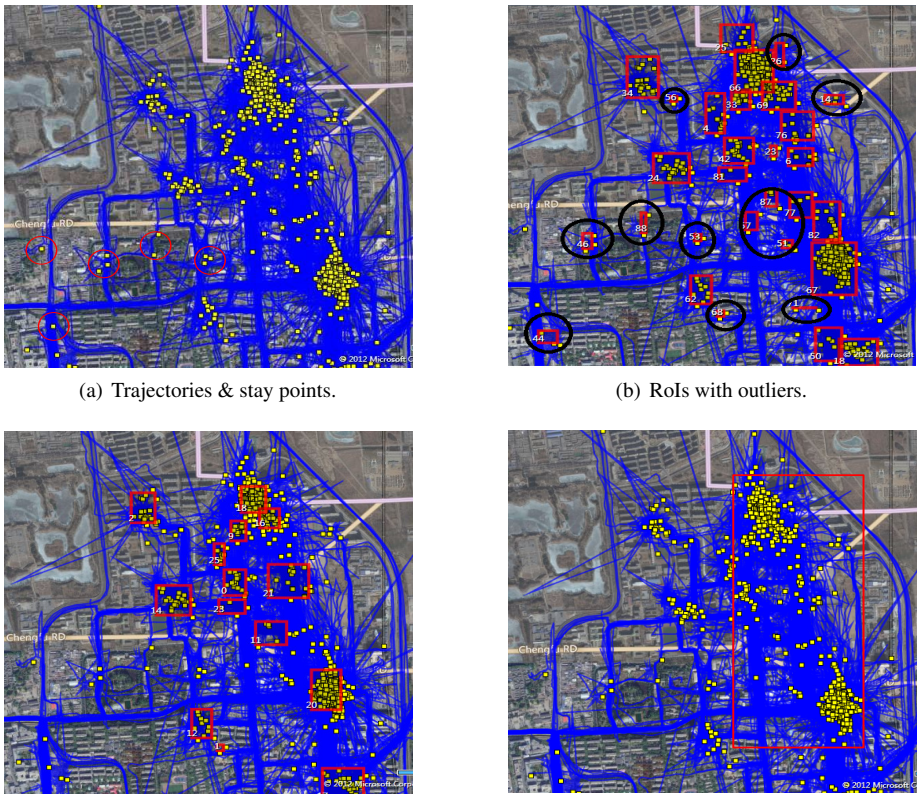
The mobility profile of this student should capture the repetitive movements mentioned in the above example among his residence, the campus and the language school. In fact, these places can be interpreted as the PoIs of the student. With this interpretation, any of such repetitive movements has a natural correspondence to a frequent T-pattern. Therefore, in this paper, we make use of T-patterns to model users' mobility profiles. If $\mathcal{T}_u$ is the set of trajectories of user u, then we can denote the mobility profile of u, i.e., $\mathcal{P}_u$ as the pair $\langle PS^{\tau,\sigma}_{\mathcal{T}_u}, sup^{\tau}_{\mathcal{T}_u}\rangle$. In the following discussions, we use PS_u to stand for $PS^{\tau,\sigma}_{\mathcal{T}_u}$ for short by assuming that τ and σ are implicitly defined and $\mathcal{T}_u$ is clear from the context. Similarly, we can write $sup^{\tau}_{\mathcal{T}_u}(S, A)$ as $sup_u(S, A)$.

With our model of mobility profiles, constructing a user's mobility profile is then reduced to mining frequent trajectory patterns which has been studied in [Giannotti et al. 2007]. In general, the procedure consists of two sequential steps – *RoI extraction* and *trajectory pattern calculation*. Going back to Example 4.1, we have to first identify all the student's PoIs labelled by grey rectangles before proceeding to extract his frequent T-patterns.

As the input of the latter step, the extracted RoIs have a significant impact on the quality of trajectory patterns. Missing some RoIs will hide some patterns while abundant RoIs will generate dummy patterns which do not belong to the user. Furthermore, the accuracy of RoIs should also be

considered important. Intuitively, an RoI is accurate if it does not cover any extra area besides the real place in which the user performs an activity. Although in practice it is hard to extract all the accurate RoIs, we should achieve a balance between the size of RoIs and the frequency of visits to them. By referring to the methods in the literature [Giannotti et al. 2007; Uddin et al. 2011; Zheng et al. 2011] and considering the setting of calculating user similarity, we propose a new extraction by improving the one based on clustering stay points. Given a 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;
(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 [Giannotti et al. 2007] to compute frequent trajectory patterns from the RoI trajectories obtained at step (3).



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

Fig. 2: An example of RoI construction.

The first two steps are about constructing RoIs. Figure 2 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 allows us to focus on users' meaningful places. Figure 2(a) 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 Figure 2(a), 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 Figure 2(a)). Such points degrade the quality of generated RoIs, e.g., enlarging area or computing infrequent places [Jain and Dubes 1988]. We introduce LOF (Local Outlier Factor) [Breunig et al. 2000] 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. Figure 2(b) and Figure 2(c) 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 Figure 2(b).

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 [Giannotti et al. 2006] and compute the T-patterns that satisfy given minimum support and time tolerance.

With respect to the RoI construction, there exist other methods in the literature. Zheng et al. [Zheng et al. 2011] use a density-based clustering algorithm OPTICS [Ankerst et al. 1999] to compute RoIs from stay points but without removing outliers. We have illustrated the shortcoming of this method by Figure 2(b) and Figure 2(c). In the trajectory pattern miner, Giannotti et al. [Giannotti et al. 2007] 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. Figure 2(d) shows the RoIs computed by the tool, which covers almost the whole area. Uddin et al. [Uddin et al. 2011] propose a different method which makes use of the heuristic that users tend to move in a low speed in RoIs. An RoI should contain a sufficiently large number of trajectory segments with speed in the predefined ranges. In this paper, we adopt the methodology based on stay points. In the following we present an example of mobility profiles.

Example 4.2. Consider the trajectories of the student in Example 4.1. Suppose his identity is assigned as u. If we represent his PoIs with the letters labelled above the corresponding rectangles in Figure 1, then after trajectory transformation (i.e., step (3)), user u has the following three RoI trajectories:

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

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, for any α such that $0 \leq \alpha \leq 4$, $A \xrightarrow{\alpha} B$ always has at least two occurrences and $sup_u(A \xrightarrow{\alpha} B) \geq \frac{2}{3} > 0.5$. When $\alpha = 2.5$, $A \xrightarrow{2.5} B$ is spatio-temporally contained in all the three trajectories. This is because the differences between α and the transition time in T_1, T_2 and T_3 are $2.5 - 1 = 1.5$, $2.5 - 2 = 0.5$ and $3 - 2.5 = 0.5$, which are all smaller than 2. For these T-patterns, we use $[\alpha_1, \alpha_2]$ to represent the transition time interval between α_1 and α_2. Thus $A \xrightarrow{[0,4]} B$ represents the set of all T-patterns with

the sequence of RoIs (A, B) and transition time between 0 and 4. We will apply the same notation in the rest of our paper. In Example 4.2, the user u's mobility profile can be represented as follows:

$$PS_u = \{A, B, C, A \xrightarrow{[0,4]} B, A \xrightarrow{[6,10]} C, B \xrightarrow{[4,8]} C, A \xrightarrow{[0,4]} B \xrightarrow{[4,8]} C\}.$$

5. COMPARING MOBILITY PROFILES

In this section, we address the comparison of two users' mobility profiles. Recall that a T-pattern consists of two components – a sequence of RoIs and a sequence of typical transition times. A comprehensive similarity evaluation between mobility profiles should take into account the information contained in both of them. We adopt a methodology that divides the comparison into two steps. Two users' similarity is first calculated based on only the sequences of RoIs and then transition time is integrated into the calculation.

When transition time is not considered, the user u's mobility profile $\mathcal{P}_u$ has a simpler form $\overline{\mathcal{P}}_u = \langle \overline{PS}_u, \overline{sup}_u \rangle$ where $\overline{PS}_u = \{S \,|\, \exists \langle S, A \rangle \in PS_u\}$ (called the *sequence pattern set*) and $\overline{sup}_u$ returns the support value of any sequence pattern in $\overline{PS}_u$. Note that for each $S \in \overline{PS}_u$, $\overline{sup}_u(S)$ is actually equivalent to the support value of any $\langle S, A \rangle \in PS_u$ when τ is set to $+\infty$, i.e., $sup_{\mathcal{T}_u}^{+\infty}(S, A)$. For instance, in Example 4.2 $\overline{PS}_u = \{A, B, C, A \to B, A \to C, B \to C, A \to B \to C\}$ and $\overline{sup}_u(A \to B \to C) = 1.0$.

The sequence patterns in a pattern set contain duplicated information. For instance, for the user in Example 4.2, if we know that $A \to B$ is one of his sequence patterns, then A and B also belong to his sequence pattern set. If we compare two users' mobility profiles using the original sequence pattern sets, some behaviour will be used more than once. Therefore, *maximal pattern sets* are introduced to replace users' sequence pattern sets in user comparison. Intuitively a maximum pattern is a sequence pattern which is not a *subsequence* of any 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 $R'_i = R_{j_i}$ $(0 \leq i \leq m)$. Formally, the maximal sequence pattern set of $\overline{PS}_u$ can be defined as follows:

Definition 5.1 (*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 rest of the section, we first present our similarity metric based on two users' maximal pattern sets and then proceed to add transition time into the similarity metric.

5.1. Similarity based on maximal patterns sets

We develop our similarity metric by extending the metric proposed by Ying et al. [Ying et al. 2010]. Specifically, we identify a flaw of this metric in which it violates a basic principle that should hold in all similarity metrics. That is, the largest similarity value should be calculated when a user is compared to himself. We also find that the metric cannot capture users' similarity precisely in some cases. Our new metric not only fixes the flaw but also provides more precise similarity assessment. Before presenting the weaknesses and our fixes, we start with a brief description of the metric of Ying et al. [Ying et al. 2010].

A similarity metric in [Ying et al. 2010]. Ying et al. define a user similarity metric [Ying et al. 2010] on mobility profiles composed by semantic trajectory patterns. By 'semantics' they mean location semantics, i.e., functionalities of places such as parks, schools or hospitals. Although the metric is defined based on maximal semantic pattern sets, its idea can be applied to maximal sequence pattern sets as well, which is shown in the following.

The similarity calculation consists of two steps. In the first step, each pair of sequence patterns from the given two maximal patterns sets respectively are compared and the result is called the

pattern similarity between them. In the second step, the calculated pattern similarity values are combined in a specific way as the final similarity .

The similarity between two maximal sequence patterns is calculated based on the intuition that the more similar they are, the longer common part they share. As their longest common part, the *longest common sequences* (LCS) are used. 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 LCSs of P and Q, denoted by $lcs(P, Q)$. Let $lenLCS(P, Q)$ be the length of the LCSs in $lcs(P, Q)$ and $len(P)$ be the length of P. According to the weighted average trajectory pattern similarity defined in [Ying et al. 2010], 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, since $lenLCS(P, Q) = 3$, $len(P) = 5$ and $len(Q) = 4$, we have $sim(P, Q) = \frac{2 \cdot 3}{5+4} = 0.67$.

The similarity between maximal pattern sets is computed by combining the pattern similarity values. Ying et al. make use of the weighted average. They introduce a weight function $w(P, Q)$ to incorporate the importance of each pair of maximal patterns, e.g., $P \in \overline{PS}_u$ and $Q \in \overline{PS}_{u'}$, into the user similarity. The similarity between users u and u', i.e., $sim(u, u')$, is calculated as follows:

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

The weight function can be defined in different ways according to the various requirements of applications for user similarity. For instance, a user may be considered to be more similar to another as long as they share more common movements in some applications, while other applications may require two similar users to share more behaviours which are only possessed by them. This has been addressed in [Ying et al. 2010] and for the sake of simplicity, in this paper, we adopt the first interpretation. Thus,

$$w(P, Q) = \frac{sup_u(P) + sup_{u'}(Q)}{2}.$$

Weaknesses of the metric in [Ying et al. 2010]. We find in some cases that the user similarity calculated by the above metric is counter-intuitive and inconsistent with common sense. In particular, it fails to satisfy a basic principle in similarity assessment, i.e., the similarity value between two identical subjects should be the maximum – 1.0 in the setting of user similarity calculation. We illustrate the weaknesses of the metric through the following example.

Example 5.2. 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 the 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 Table I. For the sake of simplicity, we assume that all patterns have the same support value 0.2.

As 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. Furthermore, the similarity between u and u_3 should be 1.0 as they are identical. However, according to the metric in [Ying et al. 2010], we have

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

Table I: 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**

In the above results, user u has the same similarity to all the other three users. This shows that the metric cannot distinguish the different similarity values of user u to the other users. In fact, the similarity values are smaller than they should be. Since the similarity between u and u_3 is calculated as $0.33 < 1.0$ and they have exactly the same patterns, the metric fails to output the maximum similarity to identical users.

In fact, it is straightforward to prove that the metric by Ying et al. will output the maximum similarity value, i.e., 1.0, if and only if two users share the same maximum pattern set of size one.

Our similarity metric. From Example 5.2, we learn that the weighted average of pattern similarities is not the correct combination of pattern similarities for user similarity calculation. In the following, we present our fix to the metric. Our main idea is to exploit the intuition that if user u is similar to user u', then any pattern of u will correspond to a similar pattern of user u'.

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 $P_i \in M(\overline{PS}_u)$,

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

Then for each user, we compute his *relative similarity* to the other one. The relative similarity of u to u', denoted by $sim(u \,|\, u')$, is calculated as the average weighted value of all the patten similarity values of the identified most similar pattern pairs:

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

Similarly, we can also compute the relative similarity of u' to u, i.e., $sim(u' \,|\, u)$. As the relation of similarity should be symmetric, we calculate the average of the two relative similarity values as the similarity between users u and u':

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

We illustrate the similarity calculation using our metric through the following example.

Example 5.3. Suppose we have the same users as in Example 5.2. 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., $\psi_{u,u_2}(P_1) = P_1$, $\psi_{u,u_2}(P_2) = P_2$ $\psi_{u,u_2}(P_3) = P_1$(or P_2). So we have $sim(u \,|\, u_2) = \frac{0.2+0.2+0}{3\times 0.2} = 0.67$. With the same process, we can obtain $sim(u_2 \,|\, u) = \frac{0.2+0.2}{0.2+0.2} = 1.0$. So the similarity between u and u_2 is $sim(u, u_2) = 0.83$. Table II lists the calculated similarity values. The similarity of u with u_1 is 0.67 which is the smallest and u has the largest similarity, i.e., 1.0 to u_3.

We can see that our method can clearly distinguish the similarity degrees of u to the other three users, which is consistent with what we shall expect. More importantly, for identical users, our metric always guarantees the maximum similarity value 1.0 to be calculated.

Table II: 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

5.2. Adding transition time

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

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 is $S = (R_0, \ldots, R_n)$ ($S \in lcs(P, Q)$). 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 $tranT^u_S(i)$ be the union, then we have $tranT^u_S(i) = \{\alpha_i \mid \exists (S, A) \in PS_u \text{ s.t. } A = (\alpha_1, \ldots, \alpha_n)\}$. In the same way, we can obtain the corresponding set of transition time of user u', i.e., $tranT^{u'}_S(i)$.

Recall that we can use intervals to represent transition time in Example 4.2. Thus $tranT^u_S(i)$ 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 $tranT^u_S(i)$ and $tranT^{u'}_S(i)$. Suppose $tranT^{u'}_S(i) \cup tranT^u_S(i) = [x_1, y_1] \cup \ldots \cup [x_k, y_k]$ and $tranT^{u'}_S(i) \cap tranT^u_S(i) = [x'_1, y'_1] \cup \ldots \cup [x'_m, y'_m]$. Then we can calculate $ot^{u,u'}_S(i)$, 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 5.4 (*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:

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

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

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

Note that since we use the average of transition time overlapping in all longest common sequences between two maximal patterns as a discount factor for pattern similarity, in general the calculated user similarity will decrease.

6. ADDING SEMANTICS

In the above discussion, user mobility profiles are described in terms of regular geographical movements. User similarity based on them may fit the applications such as finding potential partners in car pooling. But for other types of services, it may not be sufficiently effective. For instance, in social networks friends are recommended mainly based on shared interests or hobbies. If two users live in different cities but are both fond of films, then their mobility profiles will have no common patterns and their common interest will not be discovered. However, when places are labelled by

their functionalities, e.g., cinema, we will be able to detect their similarity. The functionalities of a place indicate its semantics, called *location semantics*. In practice, it is also necessary to study the interpretation of the timing information of users' movements, which we call *temporal semantics*. This is because some applications value users' behaviour in a specific time period more than others. For instance, the friends in a circle of photographing usually have similar patterns on public holidays when they pursue their hobby. In the following discussions, we present methods to calculate the similarity between two users, by taking into account location semantics and temporal semantics, respectively.

6.1. Location semantics

In this section, we give a method to find similar users considering the *location semantics* of their regular movements. Our main idea is to modify the pattern similarity metric by enriching the RoIs in sequence patterns with their location semantics.

The problem of annotating RoIs with their location semantics can be formulated as labelling an RoI with appropriate location semantic tags, e.g., shop or bar. The set of semantic tags varies between applications. For instance, tagging a place as 'entertainment' is sufficient for some applications but not for others which need to learn whether it is a cinema or a theatre. In practice, a place may usually be associated with multiple semantic tags. For instance, a place labelled by 'café' can also be associated with 'bar'. Annotation of semantic tags has been discussed extensively in the literature, especially in recent years with the booming LBSNs [Yan et al. 2011; Ye et al. 2011]. In this paper, we adopt the methodology of Ye et al. [Ye et al. 2011] which calculates a probability distribution over the set of semantic tags for a place. The probability of a tag represents the likelihood that the place is labelled by the tag. The probabilities can then be used to rank the semantic tags and the ranking indicates which is the most dominant tag.

Let $\mathcal{AL} = \{\mu_1, \ldots, \mu_n\}$ be an ordered set of location semantic tags that can be assigned to an RoI. Given an RoI R, we use tag_R to denote the location semantic tag of R. Moreover, $Pr_R(\mu_i)$ is used to represent the probability that tag_R is μ_i, and $\sum_{\mu \in \mathcal{AL}} Pr_R(\mu) = 1$. Thus we have a vector of probabilities for R, i.e., $v_R = \langle p_1, \ldots p_n \rangle$ where $p_i = Pr_R(\mu_i)$. In the following discussion, we call v_R the *location-semantics vector* of R and use $v_R(i)$ to denote its ith element, i.e., p_i.

We say that two RoIs R and R' are the same according to their location semantics when they are labelled by the same tag, i.e., $tag(R) = tag(R')$. Due to the uncertainty about $tag(R)$ and $tag(R')$, we cannot definitely determine the equivalence between them. Instead, we make use of their location semantic vectors and take two RoIs as location semantically similar (*LS-similar* for short) if the distance between them is below a threshold. Since a location semantic vector corresponds to a distribution, we make use of the notion of *relative entropy* to define our distance metric between two semantic vectors. Strictly speaking, relative entropy is not a metric as it is asymmetric. Therefore, we first introduce our location-semantics vector distance.

Let $dist_{RE}(v_R \parallel v_{R'})$ be the relative entropy from $v_{R'}$ to v_R and it is formally defined as $dist_{RE}(v_R \parallel v_{R'}) = \sum_{i=1}^{n} v_R(i) \cdot \log \frac{v_R(i)}{v_{R'}(i)}$. Then the distance between v_R and $v_{R'}$ can be expressed as

$$dist_v(v_R, v_{R'}) = \frac{dist_{RE}(v_R \| v_{R'}) + dist_{RE}(v_{R'} \| v_R)}{2}.$$

If $dist_v(v_R, v_{R'}) \leq \delta$ where δ is a pre-defined threshold of vector distance, then R and R' are LS-similar. LS-similarity can be extended to sequence patterns. Intuitively, given two sequence patterns with the same length, if any pair of the aligned RoIs are LS-similar, then we call these two patterns *LS-similar*.

Definition 6.1 (*LS-similar sequence patterns*). Let P and Q be two sequence patterns. We say that they are LS-similar, denoted by $P \approx_{LS} Q$, if

$$len(P) = len(Q) \ \wedge \ \forall_{i \leq len(P)} dist_v(v_{P(i)}, v_{Q(i)}) \leq \delta$$

where δ is the distance threshold.

We can calculate the similarity between two sequence patterns with respect to location semantics following the same method discussed in Section 5. However, instead of longest common sequences, we make use of *longest LS-similar sequences*.

For a sequence pattern P, if there exists a subsequence $P' \sqsubseteq P$ and $Q' \sqsubseteq Q$ where $P' \approx_{LS} Q'$ then we call P' a *LS-similar subsequence* of P with respect to Q. The set of longest LS-similar subsequences of P with respect to Q is denoted as $lss(P \,|\, Q)$. Although $lss(P \,|\, Q) \neq lss(Q \,|\, P)$, the length of a sequence in $lss(P \mid Q)$ is the same as that of any sequence in $lss(Q \mid P)$. We use $lenLSS(P, Q)$ to denote the length of longest LS-similar subsequences between P and Q.

Given two maximal sequence patterns P and Q of users u and u', the similarity between P and Q with respect to location semantics is defined as follows:

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

The calculation of user similarity with respect to location semantics is the same as the definition in Section 5 after substituting $sim(P, Q)$ with $lsSim(P, Q)$. We use the following example to show the calculation of the user similarity with respect to location semantics.

Example 6.2. Let $M(\overline{PS}_u) = \{A \to C \to E \to F, C \to F \to E\}$, $M(\overline{PS}_{u'}) = \{B \to D \to F, D \to E\}$ and $\mathcal{AL} = \{\mathsf{residence}, \mathsf{restaurant}, \mathsf{bar}\}$. In addition, we learn that the RoIs A and B are two zones of residence. The RoIs C and D are the areas where restaurants are intensively located while E and F are the camps of bars. The location semantic vectors of the RoIs are given as follows:

$$\begin{aligned} v_A =& \langle 0.7, 0.3, 0.0\rangle; \quad v_C = \langle 0.2, 0.7, 0.1\rangle; \quad v_E = \langle 0.1, 0.1, 0.8\rangle; \\ v_B =& \langle 0.8, 0.1, 0.1\rangle; \quad v_D = \langle 0.1, 0.7, 0.2\rangle; \quad v_F = \langle 0.2, 0.2, 0.6\rangle. \end{aligned}$$

The support values of the patterns are all set to 0.2 and δ is set to 0.5 for the sake of simplicity.

Consider $C \to F \to E$ and $D \to E$. As $dist_v(C, D) = 0.07$ and $dist_v(E, F) = 0.10$ which are both smaller than δ, the set of longest LS-similar sequences of $C \to F \to E$ with respect to $D \to E$ is $\{C \to E, C \to F\}$. Meanwhile, $lss(D \to E \,|\, C \to F \to E) = \{D \to E\}$. Thus we have $lenLSS(C \to F \to E, D \to E) = 2$. Then the similarity between $C \to F \to E$ and $D \to E$ can be calculated as follows:

$$lsSim(C \to F \to E, D \to E) = \frac{2 \cdot 2}{2 + 3} = 0.8.$$

The location semantic similarities between the other patterns are summarised in Table III. We also list the original similarity values between sequence patterns for comparison.

Table III: Example of similarity computation w.r.t. location semantics.

	$sim(P, Q)$		$lsSim(P, Q)$	
	$B \to D \to F$	$D \to E$	$B \to D \to F$	$D \to E$
$A \to C \to E \to F$	0.29	0.33	0.86	0.67
$C \to F \to E$	0.33	0.40	0.67	0.80

Subsequently, we have

$$lsSim(u \,|\, u') = \frac{0.86 \cdot 0.2 + 0.80 \cdot 0.2}{0.2 + 0.2} = 0.83, \quad lsSim(u' \,|\, u) = \frac{0.86 \cdot 0.2 + 0.80 \cdot 0.2}{0.2 + 0.2} = 0.83.$$

Thus $lsSim(u, u') = 0.83$ while $sim(u, u') = 0.37$.

It is clear that u and u' should be very similar when considering location semantics, as their mobility patterns indicate that they both favour two sequence patterns with the same location semantics: home $\rightarrow$ restaurant $\rightarrow$ bar and restaurant $\rightarrow$ bar. From this example, we can see that our method can effectively capture the similarity between users with respect to location semantics.

Note that in our method we consider two RoIs semantically the same as long as the distance between their location-semantics vectors is below a threshold (δ). One way to refine our method is to incorporate such distance into the calculation of the similarity between two patterns.

6.2. Temporal semantics

Temporal semantics refers to an interpretation of the occurring time of events, e.g., *daytime/night* and *weekday/weekend*. There are two reasons to consider temporal semantics when comparing user mobility profiles. First, in practice some applications may value more users' behaviour in a specific period of time. Taking weekday/weekend as an example, trajectory patterns on weekends are regarded as more important for recommending friends. This is because users' behaviours on weekends are more likely to reflect users' real activities of interest than weekdays when they are restricted by their professions. Second, some mobility patterns that occurs only in certain periods may remain hidden from the mobility profiles constructed from the whole trajectory dataset. The daily trajectories on weekends only take up $\frac{2}{7}$ of all the daily trajectories. If the minimum support value is set to 0.4, then no mobility patterns that are only contained in the weekends trajectories will be extracted.

In this section, we propose a method to calculate user similarity with respect to temporal semantics. Our main idea is to first annotate users' trajectories with their temporal semantics and then construct user mobility profiles based on their trajectories with the same temporal semantics. In this way, we can compute the similarity between users using their mobility profiles corresponding to different temporal semantics. At last, we provide a flexible way to combine such similarities and obtain a customised similarity measurement.

Similar to location semantics, annotating trajectories with temporal semantics can also be formulated as labelling each trajectory with a temporal semantic tag. We use $\mathcal{AT}$ to denote the set of temporal semantic tags, which is determined by applications as well. Since in this paper our purpose is to study users' regular daily movement patterns, we take a user's movements in each day as a trajectory and study the temporal semantics of daily trajectories.

In fact, a semantic tag corresponds to a set of time points. Any point in the set can be labelled by the tag. We use $timeSet(\mu)$ to denote the set of time points of the temporal semantic tag μ. For any two tags μ and μ' in $\mathcal{AT}$, their time point sets are exclusive, i.e., $timeSet(\mu) \cap timeSet(\mu') = \emptyset$. For example, $timeSet(\mathsf{weekend})$ contains all time points on March 23rd, 2013 which is a Saturday but no time points on the previous day. Given a trajectory $T = (\langle \ell at_1, \ell on_1, t_1 \rangle, \ldots, \langle \ell at_k, \ell on_k, t_k \rangle)$, if for any t_i ($1 \leq i \leq k$), $t_i \in timeSet(\mu)$, then T is annotated by μ, i.e., $tag(T) = \mu$.

Recall that $\mathcal{T}_u$ is the set of trajectories of user u. Based on the temporal semantic tags of trajectories, we can divide $\mathcal{T}_u$ into disjoint subsets of trajectories with the same tags. That is, for any $\mu \in \mathcal{AT}$ we have $\mathcal{T}_u^\mu = \{T \in \mathcal{T}_u \mid tag(T) = \mu\}$. The process of user mobility profile construction can be applied on each subset, i.e., $\mathcal{T}_u^\mu$. In this way, we obtain a mobility profile for the user with respect to a temporal semantic tag μ. We use $\mathcal{P}_u^\mu = \langle PS_u^\mu, sup_u^\mu \rangle$ to denote the mobility profile of the semantic tag μ.

Given two users u and u', we can calculate their similarity based on the mobility profiles of each temporal semantic tag. Then for each $\mu \in \mathcal{AT}$, we learn a similarity value, denoted by $sim_\mu(u, u')$. This similarity indicates the similarity between the users on their movements in the period $timeSet(\mu)$. For instance, $sim_{\mathsf{weekend}}(u, u')$ represents the similarity between u and u' of their movements on weekends.

As we have mentioned before, in practice applications usually have various opinions on users' movements in different time periods. This can be captured by assigning different weights to users' similarity values of each temporal semantic tag. The weighted average of the similarity values can thus be calculated and taken as the overall final similarity. Let w_i be the corresponding weight assigned to $sim_{\mu_i}(u, u')$ by an application. Then the similarity between u and u' with respect to

temporal semantics can be calculated as follows, with $\sum_{\mu_i \in \mathcal{AT}} w_i = 1$:

$$tsSim(u', u) = \sum_{\mu_i \in \mathcal{AT}} w_i \cdot sim_{\mu_i}(u, u').$$

Our method is flexible and can be tuned according to the different requirements of applications. Moreover, due to the patterns that used to be ignored by the original construction method, the calculated similarity becomes more accurate.

7. EXPERIMENTS

In this section, we explore two real-life GPS trajectory datasets to validate our work. One is collected in the *Geolife* project of Microsoft Research Asia [Zheng et al. 2008] while the other is collected and published by Yonsei University in Korea. Since the datasets are collected independently in two different scenarios, if the experimental results of these two datasets can lead to similar observations, then we can learn that our methods are general and effetive in practice.

Although mobility datasets can be synthesised by tools such as SUMO [Behrisch et al. 2011] and the moving object generator [Brinkhoff 2002], we decide to perform the experiments on real-life datasets. This is because we can test the robustness of our methods against the unpredictable factors which can influence users' trajectory collection in practical settings. For instance, when carrying mobility devices, users usually have different preference in exposing their locations such as frequency. Sometimes, the devices may be switched off due to privacy concerns. Such factors cannot be easily captured by synthesised datasets. In addition, we also want to validate our hypotheses made in this paper. For example, we take into account temporal semantics as we expect that users' mobility behaviour is sensitive to time periods. We cannot achieve this goal with synthesised datasets. Furthermore, the datasets we explore also have a good quality in terms of the numbers and length of the collected trajectories.

7.1. Experimental setting

Before presenting the detailed experimental results, we first give a brief description of the datasets and the parameter settings used in our experiments.

The datasets. We make use of two datasets which we call Geolife and Yonsei, respectively. The Geolife dataset consists of 17,621 trajectories from 182 users in a period of over five years (from April 2007 to August 2012). Each trajectory corresponds to a user's movement in one day. The trajectories cover a total length of about 1,250,000 km and a total duration of more than 48,000 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 the previous positions. The trajectories also reflect a diverse collection of users' outdoor movements, not only restricted to the ones related to their jobs. Almost all trajectories are located in Beijing (China) although the GPS positions are distributed in over 30 cities. After projecting the trajectories to the map, we find that the volunteers tend to have similar background since they share a common area with the highest density of visits, which is the assembling place of IT companies. This indicates a large chance that the volunteers may have similar interests to each other.

The Yonsei dataset is collected by Yonsei University in Seoul, South Korea [Chon et al. 2011]. It consists of 1,865 daily trajectories from 12 users, which cover a total length of 32,626 km. Although users movements locate in different cities or even countries, we focus on their local movements in Seoul. The Yonsei dataset is different from the GeoLife dataset as its trajectories are stored in the form of stay points. According to the description of the dataset, all the volunteers are students in Yonsei University, whose similar backgrounds also indicate similar mobility patterns between them.

Due to privacy issues, both of the trajectory datasets do not provide users' personal information in their dissemination such as gender or affiliation. Thus 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 [Li et al. 2008], we cannot obtain and use it because of the legal policy of

Microsoft [Zheng 2012] for data publishing. Therefore, in order to validate our similarity metrics, we have to find a different way without using the ground truth. We choose some users with larger numbers of trajectories and split them into new users. Intuitively, these new users should preserve the original users' behaviour and thus have high degrees of similarity between each other. If our metrics are effective, then they should be able to capture such high similarities. In the following, we briefly describe the construction of new users in the two datasets. In the GeoLife dataset, we choose two users – user 153 and user 163. Note that we keep the identities of users used in both datasets in case of possible further evaluation by readers. User 153 has 1,245 trajectories while user 163 owns 537 trajectories. We construct three new users (i.e., 153, 153^* and $153^\#$) from user 153 and two (i.e., 163 and 163^*) from user 163 by evenly splitting the sets of trajectories. In the Yonsei dataset, we also select two users – user 08 and user 12 and split each of them into another two users, i.e., 08^*, $08^\#$, 12^* and $12^\#$. In addition, from each dataset, we select some other users so that we can obtain two testing datasets consisting of 10 users each.

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 (e.g., k-means). The clustering process stops once the shortest distance between any two clusters is larger than a threshold, i.e., β. This parameter also determines the longest diagonal of generated RoIs.

We merge all users' trajectories of stay points and based on them we compute the set of RoIs. Then we transform each user's trajectories using the RoIs and construct the mobility profiles. In this way, we manage to guarantee that the trajectory patterns of all users are described with the same alphabet and comparable with each other. To compare two users, we just give their mobility profiles as input to our comparing algorithm.

To evaluate our methods handling semantics, we make some simplifications in the experiments. However, they are reasonable for the purpose of evaluating the effectiveness of our methods. First, for location semantics, as our purpose is to show the changes on the similarity measurements, it is sufficient to synthesise the location semantic distribution assigned to each RoI. Another reason for this is that there is no public precise information about the geological positions of semantic places, such as restaurants and bars. As the types of location semantics can be categorised in various ways and the inclusive relations between the categories can form a hierarchical structure. In practice, the location semantic tags that are of interest differ between applications in terms of not only their names but also levels. Therefore, we do not fix the types of location semantics in our experiments. Second, with regard to temporal semantics, we take a simple set of semantic tags – $\{\mathsf{weekday}, \mathsf{weekend}\}$ as an example. As a trajectory in the GeoLife and Yonsei dataset captures a user's daily movement, we label each trajectory with a weekday or weekend tag based on its date.

Parameter setting. All related parameters need to be fixed before performing our evaluation. The principle of setting their values is to enforce T-patterns to have good qualities, e.g, in terms of lengths and support values. We skip the selection process of the values and only list the fixed values for some important parameters in Table IV. Note that we use $\%dp$ to denote the deletion percentage of outliers.

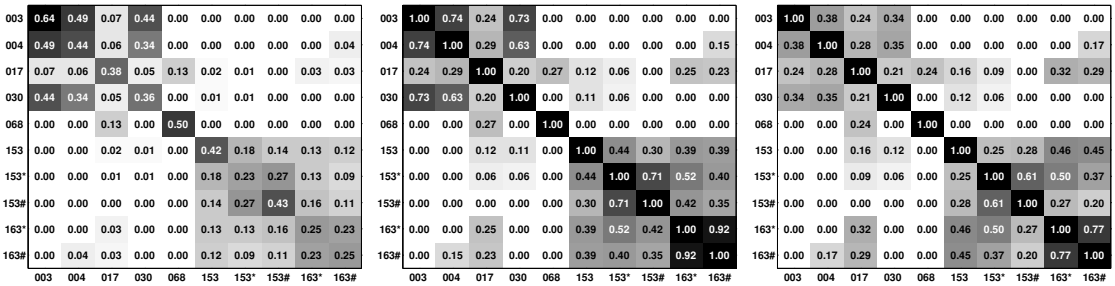
Table IV: Values of the parameters.

parameter	value	parameter	value
θ_d	200m	σ	10%
θ_t	30min	β	100m
θ_m	200m	$\%dp$	20%
τ	7200s		

7.2. Experimental results

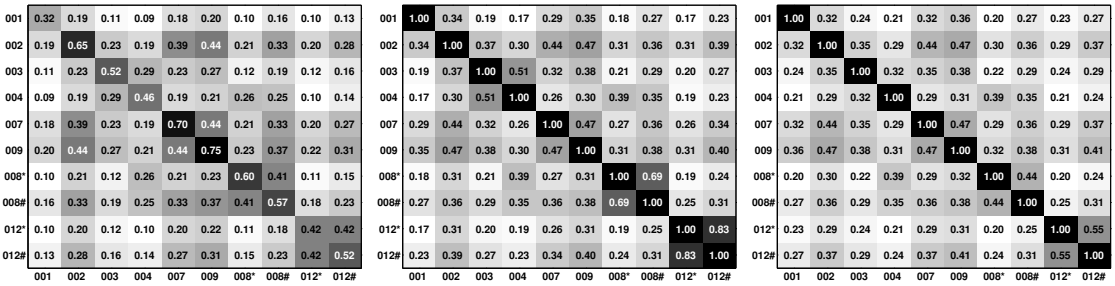
In this section, we show and analyse the experimental results. We divide our experiments into three phases so as to extensively validate our work. In the first phase, we construct users' mobility profiles and test our similarity metric on users' mobility profiles. In the last two phases, we test the effectiveness of our methods for handling location semantics and temporal semantics, respectively.

User mobility similarities. In Figure 3 and Figure 4, we show the mobility similarity values between each pair of our chosen users in the datasets Geolife and Yonsei, respectively. For the sake of comparison, given a pair of users, we calculate their similarity using three different similarity metrics. We use different grey levels to distinguish the similarity values. Specifically, a darker cell indicates the corresponding two users are more similar.



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

Fig. 3: User similarities by three methods (GeoLife).

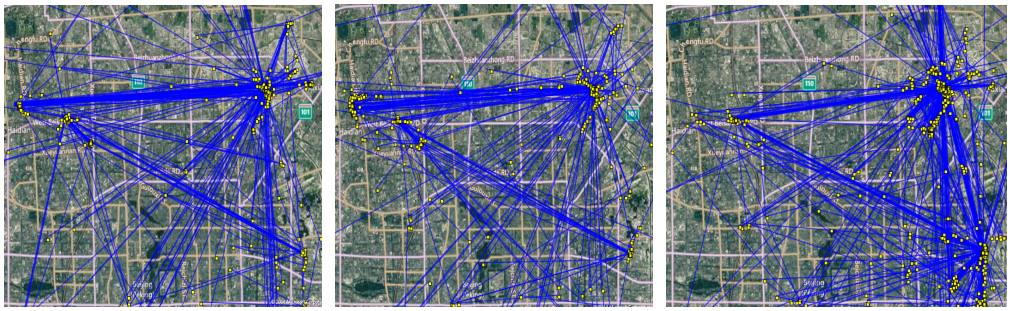


(a) User similarities by MSTP. (b) User similarities by MTP. (c) User similarities by MTP+TOF.

Fig. 4: User similarities by three methods (Yonsei).

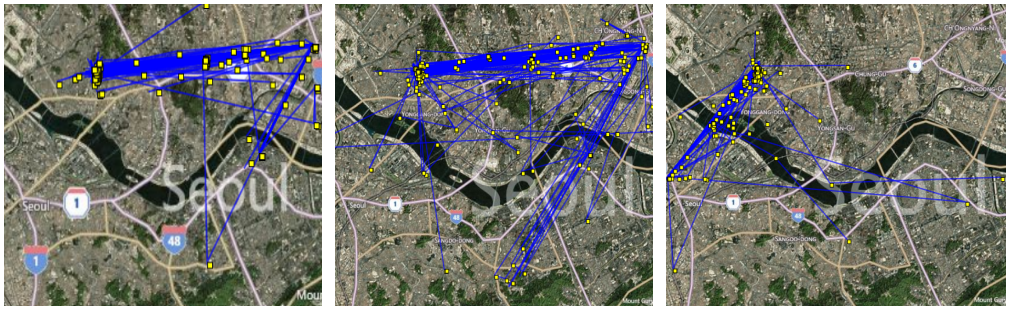
Figure 3(a) and Figure 4(a) show the user similarity values computed by the metric of Ying et al. [Ying et al. 2010] (MSTP) while Figure 3(b) and Figure 4(b) present the results given by our metric without transition time (MTP). We have two main observations with regard to these two metrics. First, the diagonal cells correspond to the similarity of a user to himself which is expected to be 1.0. However, from Figure 3(a) and Figure 4(a) it is clear that MSTP fails to satisfy this basic principle. Second, our metric MTP can give a more precise evaluation of user mobility similarities. We validate this observation for the two exclusive types of users according to whether they are derived from the same users. For the users who are derived from the same volunteers, they should preserve the mobility patterns of the original volunteers and thus share more common trajectory patterns. In

other words, they will have larger similarity values. Our metric can successfully verify this argument while the metric MSTP cannot clearly distinguish the similarity values between these users from those between the other users. For instance, in Figure 5(a) and 5(b), we depict the trajectories of user 163^* and $163^\#$ on the map. We can see that the trajectories annotated by blue curves scatter in a very similar pattern, which indicates a large degree of similarity. However, the similarity value between 163^* and $163^\#$ by the metric MSTP is only 0.23. On the contrary, our metric MTP gives 0.92. This observation also holds among the users derived from 153. In the Yonsei dataset we have the same observation. The similarity value between 012^* and $012^\#$ increases from 0.42 to 0.83 (see Figure 6(a) and 6(b)).



(a) 163^* (b) $163^\#$ (c) 153^*

Fig. 5: The trajectories of 163^*, $163^\#$, and 153^* in Geolife.



(a) 12^* (b) $12^\#$ (c) 009

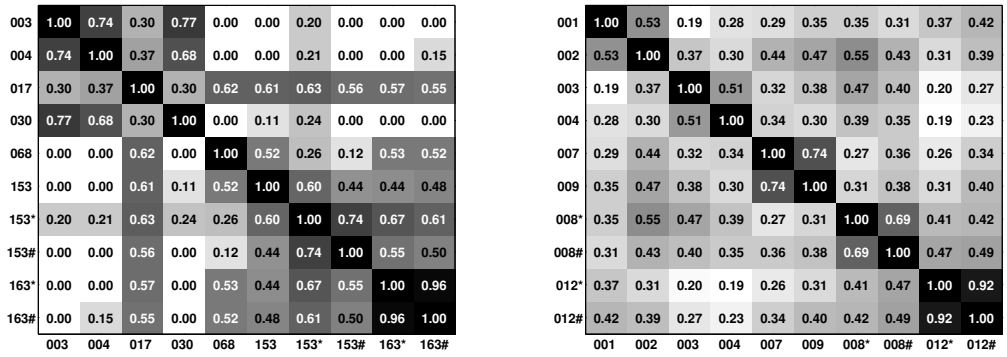
Fig. 6: The trajectories of 12^*, $12^\#$, and 009 in Yonsei.

In addition, our metric can derive a more precise similarity between users who are not from the same volunteers. This can be observed on both of the two datasets. From Figure 5, we can observe that in the Geolife dataset user 153^* have similar trajectories to 163^* and $163^\#$ while the metric MSTP just gives 0.13 (for users 153^* and 163^*) and 0.09 (for users 153^* and $163^\#$), which are apparently not compatible with this observation. Instead, our new metric increases these two similarities to 0.52 and 0.40, respectively. Considering the different support values possessed by user 153^* who has more movements on the right side of the city, our evaluation is more accurate. For the Yonsei dataset (see Figure 6), it is clear that user 009 only has one small common area with 012^* and $012^\#$ and he mainly travels in another side of the city. Considering the common movements they share, we should assign a much higher similarity to users 012^* and $012^\#$ such that we can

clearly distinguish them from user 009. However, the metric MSTP calculates 0.42 for users 012^* and $012^\#$ which is not much different from the similarity values when 12^* and $12^\#$ are compared to 009, i.e., 0.22 and 0.31, respectively. When our metric MTP is used, the similarity between 12^* and $12^\#$ increases to 0.83 while the other two similarity values only increase slightly to 0.31 and 0.40. The difference among the values (i.e., 0.83, 0.31 and 0.40) can capture the distinctiveness of 009 from 12^* and $12^\#$ more accurately.

Figure 3(c) and Figure 4(c) show the similarity between users when we add time overlapping fraction into our metric (MTP+TOF). Compared to the values in Figure 3(b) and Figure 4(b), we find that the similarity values between users decrease in general. That is because the difference between transition time discounts the similarities. We can also see that transition time does help identify similar users. For example, for the Geolife dataset, by MTP the similarity between users 003 and 004 is 0.74 which is larger than the similarity between users 153^* and $153^\#$ (0.71) even they are constructed from the same volunteer. With transition time added, the former similarity decreases to 0.38 while the later is 0.61. This is 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. Note that in our following experiments, we take MTP+TOF as the default metric unless we explicitly specify the metric used.

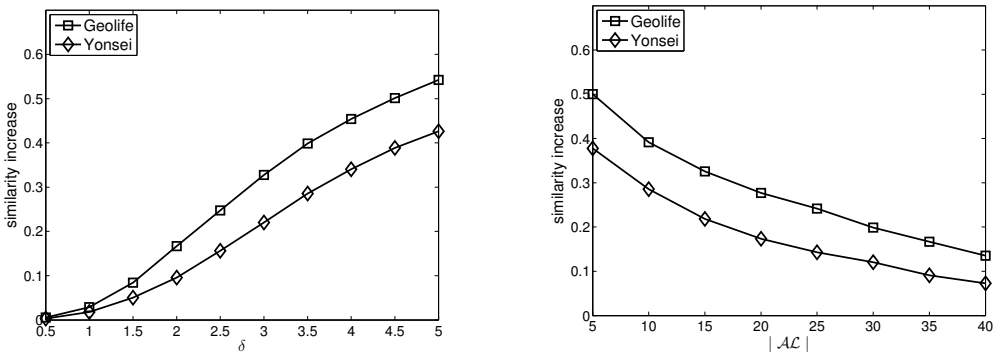
Adding location semantics. We proceed to illustrate the impacts of location semantics on user similarity calculation. Recall that in order to determine LS-similar patterns, we should set the minimum distance δ allowed between two LS-similar RoIs. We also mentioned in Section 6.1 that the number of location semantic tags is also not fixed. It is determined by applications according to different scenarios. We start with discussing the changes that occur after location semantics are considered and then proceed to show the influences of the values of the parameters – δ and the number of location semantics tags, i.e., $|\mathcal{AL}|$.



(a) Geolife. (b) Yonsei

Fig. 7: User similarity w.r.t. location semantics.

Figure 7 shows the similarity values between users considering location semantics, when δ is set to 2.0 and 10 semantic tags are chosen. Note with a focus on the impact of location semantics, transition time is not taken in account. By comparing it with Figure 3(b) and Figure 4(b), we can observe two differences. First, some pairs of users that are totally different (with similarity value of 0) become similar to some extent. For instance, the similarity value of users $153^\#$ and 017 in the Geolife dataset is 0.56 indicating a high degree of similarity when taking into account location semantics. Second, the similarity values of users which are not zero get increased as well. Take users 007 and 009 in the Yonsei dataset as an example. Their similarity grows from 0.47 to 0.74 with location semantics.



(a) Similarity increase with δ.

(b) Similarity increase with the number of tags.

Fig. 8: User similarity increase v.s. δ and $\mid \mathcal{AL} \mid$.

The increases of users' similarity values are influenced by the threshold δ and the number of location semantic tags. We depict their impacts in Figure 8. Each number on the curves is an average of 100 different location semantic distributions. Figure 8(a) shows the average increase of the similarity values of all pairs of users when different distance thresholds are used. For both datasets, the increase gets larger as δ grows. When δ is set to 3.0, the increase has been over 0.2. On one hand, this is because a larger δ results in more LS-similar patterns. Two patterns which are completely different from the perspective of geography may become LS-similar to each other. On the other hand, a larger threshold can also make the longest common sequences of two patterns longer and thus lead to larger pattern similarity subsequently. However, the scale of increase differs between the two datasets. The Geolife dataset always has larger increases for the same threshold. In Figure 8(b), we show the changes of user similarity increase along with the number of location semantic tags when the threshold is set to 3.5. Our two testing datasets share the same tendency, i.e., the increase will get smaller when more semantic tags are used. Intuitively, this is because a finer categorisation of the functionalities of locations makes it harder for two places to share common functionalities. Therefore, in order to ensure an accurate evaluation of user similarity considering location semantics, we should assign an appropriate value to the threshold δ based on the number of location semantic tags.

Adding temporal semantics. As we mentioned, we use $\{\mathsf{weekday}, \mathsf{weekend}\}$ as the set of temporal semantic tags $\mathcal{AT}$ and take it as an example to illustrate the impacts of temporal semantics on user similarity comparison. We start with evaluating our observation that users tend to have different mobility profiles in real life. Then we compare the similarity between our 10 chosen sample users in terms of their movements on weekdays and weekends, respectively.

Given a user u, after constructing his mobility profiles on weekends and weekdays, we have three profiles in total available. Specifically, they are constructed based on the whole trajectory dataset, the trajectories on weekdays and the trajectories on weekends respectively, i.e., $\mathcal{P}_u$, $\mathcal{P}_u^{\mathsf{weekday}}$ and $\mathcal{P}_u^{\mathsf{weekend}}$. In order to check their difference, we compute the similarity values between them using our metric. In this way, we can learn their difference by the intuition that the smaller a similarity value is, the more different the mobility profiles are from each other. Figure 9 shows the similarity values calculated between the three profiles for the 10 chosen users. We have two main observations from Figure 9. The first is that users' movements on weekdays contribute more to their mobility profiles based on all the trajectories. In general, the similarity between $\mathcal{P}_u$ and $\mathcal{P}_u^{\mathsf{weekday}}$ is significantly larger than that between $\mathcal{P}_u$ and $\mathcal{P}_u^{\mathsf{weekend}}$ (about 0.24 on average). This is because 5 days in a week are labelled as weekdays compared to 2 days labelled as weekends. As a result, there are more trajectories on weekdays and thus a larger number of mobility patterns are generated from week-

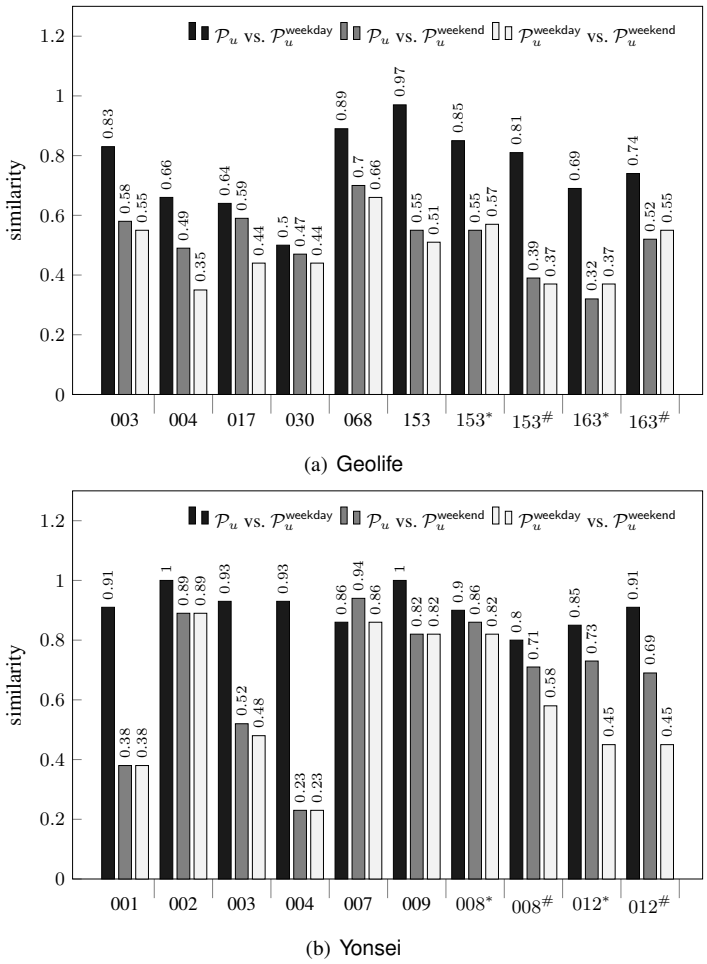
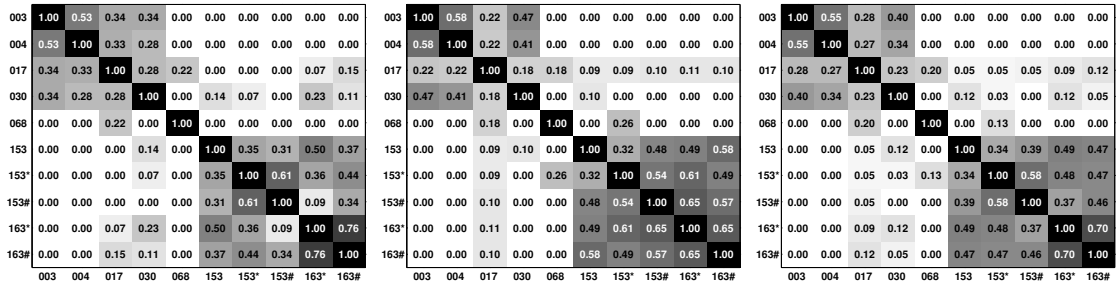


Fig. 9: The comparison of $\mathcal{P}_u$, $\mathcal{P}_u^{\text{weekday}}$ and $\mathcal{P}_u^{\text{weekend}}$.

days than weekends. The second observation is that users do have significantly different mobility patterns on weekends and weekdays. The average similarity values between $\mathcal{P}_u^{\text{weekday}}$ and $\mathcal{P}_u^{\text{weekend}}$ are only 0.48 and 0.58 in the Geolife dataset and the Yonsei dataset, respectively.

From the above discussion, we can infer that the similarity values between the same pair of users will be different between time periods. We take the Geolife dataset as an example to validate this inference. Figure 11(a) and Figure 11(b) summarise the mobility similarity scores between users on weekdays and weekends, respectively. By comparing them, we can see that some mobility patterns on weekends are ignored in the mobility profiles generated from the whole trajectory dataset (e.g., $\mathcal{P}_u$). User 068 and 153^* are completely different based on their mobility profiles computed from all trajectories and those on weekdays (see Figure 3(c) and Figure 11(a)). However, their similarity increases to 0.26 according to their movements on weekends (see Figure 11(b)). This is mainly because the number of the supporting trajectories of the patterns on weekends is not large enough to result in a support value larger than the minimum support in the original dataset. Figure 11(c) shows the weighted average similarity between users when the similarity values on weekdays and weekends are assigned the same weight, i.e., 0.5. We can find that the similarity values are different from the ones without temporal semantics considered. Furthermore, by adjusting

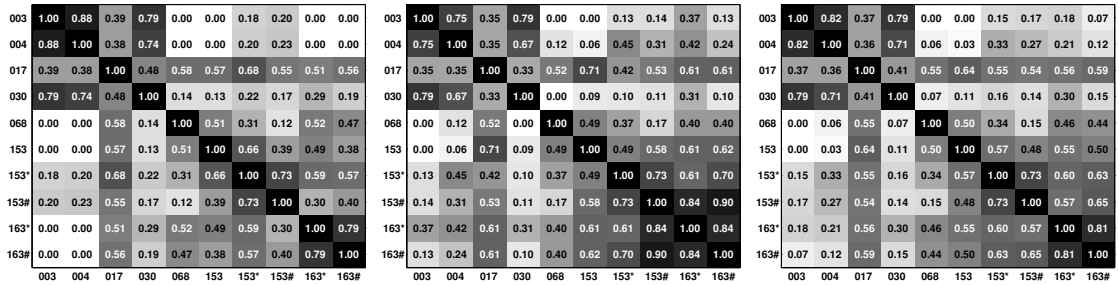
the weight distribution, our method is flexible enough to meet the various requirements of practical applications.



(a) User similarity on weekdays. (b) User similarity on weekends. (c) Average user similarity.

Fig. 10: User similarity w.r.t. temporal semantics (Geolife).

Adding location and temporal semantics. From the above discussion, we have studied the effectiveness of our similarity metrics and the changes of users' similarities when location semantics and temporal semantics are added separately. In Figure 11, we show the similarity between users when both location and temporal semantics are taken into account. Compared to Figure 10, we also have the observation that user similarity increases when location semantics are considered for each temporal semantic tag. However, there is an additional observation that users become more similar to each other on weekends when location semantics are added. On average, users' similarity value increases by 0.20 on weekends compared with 0.16 on weekdays. This can be explained by the fact that users tend to perform similar activities when they are out of work.



(a) User similarity on weekdays. (b) User similarity on weekends. (c) Average user similarity.

Fig. 11: User similarity w.r.t. temporal semantics and location semantics (Geolife).

8. CONCLUSION

In this paper, we have accomplished three tasks. First, we proposed 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 flawed in some cases and we defined a new measurement to fix the problems. As transition time between RoIs is also part of users' mobility patterns, we further took it into account in our user similarity measurement. Third, we proposed a method to integrate two types of semantics into user similarity comparison, which are locations semantics and temporal semantics. Among

them, temporal semantics are usually ignored in the related literature. We validated our work by experiments on two datasets of real-life trajectories. The results show that our measurement and user profile construction are effective and efficient.

For future work, we will apply our similarity measurement for 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, such as check-ins, posted on social networks.

REFERENCES

R. Agrawal and R. Srikant. 1995. Mining sequential patterns. In *Proc. 11th International Conference on Data Engineering (ICDE)*. IEEE CS, 3–14.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. 1999. OPTICS: Ordering points to identify the clustering structure. In *Proc. 20th ACM SIGMOD International Conference on Management of Data (SIGMOD)*. ACM Press, 49–60.

M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. 2011. SUMO - Simulation of Urban MObility: An overview. In *Proc. 3rd Conference on Advances in System Simulation (SIMUL)*. 63–68.

M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. 2000. LOF: Identifying density-based local outliers. In *Proc. 21st ACM SIGMOD International Conference on Management of Data (SIGMOD)*. ACM Press, 93–104.

T. Brinkhoff. 2002. A framework for generating network-based moving objects. *GeoInformatica* 6, 2 (2002), 153–180.

X. Chen, J. Pang, and R. Xue. 2013. Constructing and comparing user mobility profiles for location-based services. In *Proc. ACM Symposium on Applied Computing (SAC)*. ACM Press, 264–269.

Y. Chon, E. Talipov, H. Shin, and H. Cha. 2011. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proc. 9th International Conference on Embedded Networked Sensor Systems (SenSys)*. ACM Press, 82–95.

F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. 2006. Mining sequences with temporal annotations. In *Proc. 21st ACM Symposium on Applied Computing (SAC)*. ACM Press, 593–597.

F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, and M. Axiak. 2007. Trajectory pattern mining. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM Press, 330–339.

A. K. Jain and R. C. Dubes. 1988. *Algorithms for clustering data*. Prentice-Hall, Inc.

Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. 2008. Mining user similarity based on location history. In *Proc. 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (GIS)*. ACM Press, 34–43.

H. Ma, H. Cao, Q. Yang, E. Chen, and J. Tian. 2012. A habit mining approach for discovering similar mobile users. In *Proc. 21st World Wide Web Conference (WWW)*. ACM Press, 231–240.

A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. 2009. WhereNext: a location predictor on trajectory pattern mining. In *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 637–646.

J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. 2004. Mining sequential patterns by pattern-growth: the PrefixSpan approach. *IEEE Transactions on Knowledge and Data Engineering* 16, 11 (2004), 1424–1440.

I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong. 2011. On the levy-walk nature of human mobility. *IEEE/ACM Transaction on Networking* 19, 3 (2011), 630–643.

C. Song, T. Koren, P. Wang, and A.-L. Barabási. 2010. Modelling the scaling properties of human mobility. *Nature Physics* 6 (2010), 818–823.

M. Reaz Uddin, C. V. Ravishankar, and V. J. Tsotras. 2011. Finding regions of interest from trajectory data. In *Proc. 12th IEEE International Conference on Mobile Data Management (MDM)*. IEEE CS, 39–48.

X. Xiao, Y. Zheng, Q. Luo, and X. Xie. 2010. Finding similar users using category-based location history. In *Proc. 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*. ACM Press, 442–445.

Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. 2011. SeMiTri: A framework for semantic annotation of heterogeneous trajectories. In *Proc. 14th International Conference on Extending Database Technology (EDBT)*. ACM Press, 259–270.

M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM Press, 520–528.

J.-C. Ying, H.-C. Lu, W.-C. Lee, T.-C. Weng, and S. Tseng. 2010. Mining user similarity from semantic trajectories. In *Proc. 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN)*. ACM Press, 19–26.

Y. Zheng. 2012. Personal communication. (2012).

Y. Zheng, L. Wang, R. Zhang, X. Xie, and W.-Y. Ma. 2008. GeoLife: Managing and understanding your past Life over maps. (2008).

Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. 2011. Recommending friends and locations based on individual location history. *ACM Transactions on the Web* 5, 1 (2011), 1–44.

Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. 18th International Conference on World Wide Web (WWW)*. ACM Press, 791–800.