Faculty of Science, Technology and Communication

New User Similarity Measures Based on Mobility Profiles

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information and Computer Sciences

Author:
Ruipeng LU

Supervisor:
Prof. Sjouke Mauw

Reviewer:
Prof. Yves Le Traon

Advisor:
Dr. Jun Pang
Xihui Chen

September 2013

Acknowledgements

I wish to express my gratitude towards my supervisor SJOUKE MAUW. He gave me the opportunity of attending the one-year exchange program in the University of Luxembourg and experiencing a memorable and colorful life in Luxembourg, and warmly took me into the SaToSS group. He taught me how to approach a daunting research task, and offered support and encouragement to me.

I am indebted to my advisor JUN PANG. He guided me on the right path of my research and were always patient with my questions during our discussions. He is expert at enlightening me as to specific research problems. His clear views, sharp thoughts and gentle nature made our discussions both thought-stimulating and delightful.

I wish to thank XIHUI CHEN, my advisor and deep in my heart, my brother. He helped me to overcome research dilemmas and get rid of research confusions. His great patience enabled me to always seek advice in time. He taught me how to write a thesis well with generous suggestions and support. He also taught me priceless life philosophy and gave me selfless help for my life here.

I wish to thank everybody who helped me during the writing of this thesis and my stay in Luxembourg.

Last but not least, thank my parents for everything.

Contents

Abstract

Social networks has become an essential part of many people's everyday lives. A lot of social networks have deployed location-based services with which friends can share favorite places or outdoor activities by virtue of the increasingly ubiquitous devices that are able to acquire locations accurately. Location-based services leads social networks to a new approach of user recommendation based on the similarity of users' mobility profiles which are their frequent movement behavior extracted from their movement history. This can be done by measuring the extent of proximity between users' movement trajectories, or semantically by measuring the extent of closeness between the functionalities of places often visited by users.

In this thesis, we propose multiple desired principles that user similarity measures based on mobility profiles should follow, and identify the defects of the existing user similarity measures in the literature and fail to follow the principles, and further propose three novel similarity measures, one without semantics and the other two with semantics, that avoid those defects and follow the principles. The experimental results on real datasets corroborate that our new similarity measures outperform the existing ones. We also develop the MinUS software tool that coalesces the mobility profile construction process and the comparison of user similarity using the existing or our new similarity measures.

Chapter 1

Introduction

1.1 Background

In recent years, mobile devices equipped with positioning chips have become very popular, e.g., smart phones. Especially, due to the free access to navigation systems such as GPS (Global Positioning System), people can obtain their real-time positions with a high precision. This leads to a new type of social networks – *geosocial networks* (GSN) such as Bikely [3], Foursquare [6]. What is unique in GSNs is that people can attach their locations to their messages. For example, photos and videos can be tagged with the shooting place. Even the traditional social networks such as Google+ and facebook have also been upgraded to support this location sharing service. People's whereabouts shared with their friends in GSNs subsequently generate new services such as nearby friend search and place recommendation based on the visited places of their friends. With time passing by, the posted locations are accumulated and form a dataset of people's mobility histories. This results in a new opportunity for the classic friend recommendation service of GSNs as people's historical movements can significantly reveal their interests. For instance, if Alice often goes to book stores, then we can infer that she is fond of reading. To implement the new recommendation service, it is necessary to calculate the similarity between users based on their movement records.

The comparison between users' mobility has attracted a lot of research in the literature. One popular and promising method is to make use of user mobility profiles in the form of trajectory patterns [18]. A trajectory pattern is usually represented as a sequence of places which a user frequently visits and the typical transition times between two successive places. For instance, every morning Pierre, a student in Luxembourg, spends ten minutes moving from the bus stop Hamilius to the campus Kirchberg, from which in the afternoon he spends another five minutes on the way to Auchan. His daily routine can be described as a trajectory pattern:

$$Hamilius \xrightarrow{10\ min} Campus\ Kirchberg \xrightarrow{5\ min} Auchan.$$

The calculation of user similarity is reduced to the comparison between mobility profiles. The main idea is that two users are more similar once they have more common mobility patterns.

The semantics of locations are also taken into account in the construction of mobility profiles. This captures the observation that two users whose movement traces are distant from each other may also share similar interests. For instance, Bob who also likes reading lives in another city which is far from Alice's place and goes to different book stores from Alice. However, no matter where the book stores are located, they have the same

semantics and reveal the same information with regard to Bob and Alice's hobbies. The semantic mobility profiles allows the comparison between users from a different perspective.

1.2 Related work

In the literature there have been a number of works about constructing and comparing user mobility profiles.

Mobility profile construction: Giannotti et al. [18] introduce the concept of trajectory patterns which represents a set of individual trajectories all of which go through the same sequence of places named regions of interest (RoIs). Trajectory patterns are derived from the concept of temporally annotated sequences (TASs) [16] by restricting the elements of TASs to RoIs. TASs in turn are an extension of the concept of frequent sequential patterns (FSPs) by adding information about the typical transition times between their elements to sequences. The FSP concept is introduced by Agrawal et al. [13], which refers to all frequent sequences in a database of sequences D, i.e., the number of sequences in D that have a sequence as a subsequence reaches a certain percentage. Many algorithms for extracting frequent sequential patterns have been proposed, like PrefixSpan [19] and SPADE [23], among which PrefixSpan is the most efficient and widely used. PrefixSpan is extended by Giannotti et al. [17] to mine frequent trajectory patterns which compose frequent pattern sets or mobility profiles. Chen et al. [14] improve the mobility profile construction process proposed by Giannotti et al. [17] to find more precise and meaningful regions-of-interest (ROIs), by removing outliers and using a clustering procedure other than a region growing procedure.

Comparing user mobility: A personalized friend and location recommender system is proposed and implemented by Zheng et al [25]. They also collected a dataset whose source data are GPS point trajectories. GPS points are clustered into stay points which represent the sites where users stay over a period of time. Stay points are then clustered into RoIs hierarchically by a density-based algorithm. The longest common subsequences (LCSs) are extracted for a pair of users and used to measure their similarity after transforming their trajectories into sequences of RoIs. A similar method that takes semantics into account is proposed by Xiao et al. [20], in which a GPS trajectory is transformed into a sequence of the functionalities of locations, like schools and hospitals. However, both methods [25, 20] work on the level of trajectories, which might contain some places rarely visited. These places do not belong to users' typical movement behaviour and might interfere with the process of comparing user similarity. Ying et al. [22] propose a method to compare user similarity semantically but on the level of frequent patterns. They use PrefixSpan to mine frequent patterns and develop a similarity measure, called *maximal semantic trajectory pattern similarity* (MTP similarity). Maximal trajectory patterns, or maximal patterns, are those patterns that are not contained in any other frequent patterns. In the MTP similarity measure, the comparison between users is based on the comparison between maximal patterns. Chen et al. [14] improve the MTP similarity measure by remedying a defect which is that when comparing two identical users using the similarity measure the similarity value is not necessarily one, and extend it to take temporal information into account. Chen et al. [15] further extend the improved MTP similarity measure to take semantics into account.

1.3 Motivation

The concept of similarity have been studied in many domains, such as mathematics and computer science. There are three principles that a valid similarity metric should follow:

1. Two users are completely different, i.e., the similarity value is 0, if and only if they have no common patterns in their mobility profiles.

2. Given a user's behavior, he is more similar to another user compared to others if they have (i) more common behaviors; (ii) closer preference, or frequencies on the common behaviors.

3. The similarity value between two users is maximum, i.e., 1, if and only if they are exactly the same. Specifically, the mobility patterns in the two profiles are the same and have the same support values.

However, the MTP similarity measure proposed by Ying et al. and the improved MTP measure proposed by Chen et al. fail to satisfy two principles.

Both of the two measures violate the third principle. Specifically, as we mentioned, when the MTP similarity measure is used, even two identical users do not necessarily have the maximum similarity value, while the improved MTP measure ignores the impact of users' different preference, i.e., the frequencies of the patterns. In other words, as long as two users share the same maximal pattern set, then the similarity value will be 1.

Both of the measures violate the second principle. They assign a weight, which is the average support value, to each pair of mobility patterns being compared. In fact this is incorrect. This only captures the relative importance of each pair of sequence patterns to other pairs of patterns. The absolute magnitudes of the frequencies of a pair of patterns play no role in determining the similarity between two users. Thus it does not capture the extent of closeness between frequencies of common behavior.

These defects of the improved MTP measure still remain in its semantic version [15].

This thesis precisely addresses the problem of how to devise new similarity measures (with semantics) using frequent pattern sets, and thus better measure user similarity based on users' movement history in order to make the resultant similarity values as close as possible to the real extent of similarity among them. First we attempt to seek a new user similarity measure that follows the three basic principles and eradicates the defects of the (improved) MTP similarity measure. Then we attempt to extend the new similarity measure to enable it to consider semantic information, and we also attempt to develop another similarity measure which takes semantics into account from scratch.

Generally, when developing a new similarity measure we use a "theory-experiment" methodology, which means that first we give the desired principles that the similarity measure should follow, then according to the principles we devise the working process of the new similarity measure, finally we carry out experiments on real datasets to corroborate that the similarity measure is effective. We also choose the appropriate ones from a couple of candidate solutions for a specific problem by checking whether they satisfy the desired principles.

1.4 Our contributions

We propose three novel similarity measures, one without semantics and the other two with semantics, and carry out experiments on two real datasets. We also develop the MinUS (Mine User Similarity) software tool which has two major functionalities, mobility profile construction and user similarity comparison.

1. Based on the principles mentioned above, we propose a new similarity measure, called the CPS-based similarity measure, by directly comparing two users' frequent pattern sets instead of being based on the comparison of patterns. This measure eliminates the above-mentioned defects of the (improved) MTP measure.

2. The CPS-based similarity measure does not consider semantics. This is not enough, because it finds similar users who not only have similar interests, but also have close geographical trajectories. But we want to find users whose interests are similar regardless of whether their geographical trajectories are close to each other or not. To tackle the problem, we propose a method of extending the CPS-based similarity measure to make it take semantics into account using the same representation of semantic information as in the improved MTP similarity measure with semantics. The resultant CPS-based similarity measure with semantics also eliminates the above-mentioned defects of the improved MTP measure with semantics.

3. The CPS-based similarity measure with semantics removes some infrequent behavior, so we propose another new similarity measure with semantics based on the notion of Hausdorff distance that takes users' whole behavior into consideration.

4. We conducted experiments on two real datasets, Geolife [24, 7] and Yonsei [12], using our new similarity measures. The comparison between the results of our new similarity measures and those of the improved MTP similarity measure (with semantics) shows that the CPS-based similarity measure has a better performance than the improved MTP similarity measure, and the CPS-based and the Hausdorff distance-based similarity measures with semantics also have a better performance than the improved MTP similarity measure with semantics.

5. We developed the MinUS tool whose major functions are to construct user profile from source data of geographical trajectories using and to measure user similarity using the (improved) MTP similarity measure or our new similarity measures. It also implements the management of datasets.

Chapter 2

Preliminaries

In this chapter, we briefly introduce Chen et al.'s method of constructing user profile, the MTP similarity measure and its improved version (with semantics), during which we elucidate fundamental notions that will also be used in our new user similarity measures in the following chapters.

2.1 Chen et al.'s method of constructing user profile

In the user profile construction method proposed by Chen et al. [14], first we need to collect source data which are usually in the form of GPS points. A GPS point is a position on the earth's surface and here denoted by (lat, lng) which gives its latitude and longitude. Users' movement can be actually represented by a sequence of GPS points which forms a GPS trajectory.

Definition 1 (GPS trajectory)**.** *A GPS trajectory is a sequence of chronologically ordered spatio-temporal points, i.e.* $(p_1, \ldots, p_n)$ *where* $p_i = \langle lat_i, lng_i, t_i \rangle (0 \leq i \leq n)$ *with* t_i *as a time instant and* (lat_i, lng_i) *as a GPS point.*

A stay point stands for a geographic region, where a user stays over a time interval threshold θ_t and within a distance threshold θ_d. Let $dis(p, p')$ be the distance between two GPS points p and p'.

Definition 2 (Stay point)**.** *A stay point* s *of a GPS trajectory* $T = (p_1, \cdots, p_n)$ *corresponds to a subsequence* T' *of* T*. If* $T' = (p_j, \ldots, p_{j+m})$ *where* $\forall_{x=0}^{m} dis(p_j, p_{j+x}) \leq \theta_d, dis(p_j, p_{j+m+1}) > \theta_d$ *and* $t_{j+m} - t_j \geq \theta_t$*, then we have* $s = (lat, lng, t_a, t_d)$ *where* $lat = \frac{\sum_{x=0}^{m} lat_{j+x}}{m+1}, lng = \frac{\sum_{x=0}^{m} lng_{j+x}}{m+1}$ *stand for the average latitude and longitude of the points in* T'*,* $t_a = t_j$ *is the arrival time at* s *and* $t_d = t_{j+m}$ *is the departure time.*

In reality, there is a good chance that a user starts and ends a trajectory at his meaningful places, like his home or office. Thus the first point and the last point of a GPS trajectory are directly regarded as two stay points. If the two stay points are close to other stay points and the distances are below a threshold θ_m, we merge them into one stay point by replacing them with the middle point of the line segment connecting them.

Multiple stay points are likely to be close to each other and belong to one meaningful region. For example, a student might linger at several sites on a campus, then the campus will include several stay points. And one stay point only belongs to one user, so stay points cannot be directly used to compare two users' similarity. Thus next we cluster all the stay points of the users who we want to compare into regions of interest shared

by these users. An region of interest (RoI) is a geographic region where one user carried out an activity.

There might exist outlying stay points rarely visited by users. Such points degrade the quality of generated RoIs, e.g. enlarging their areas or producing improper RoIs. Thus before generating RoIs, a certain percentage (called deletion percentage) of points with the greatest LOF values are removed. LOF (local outlier factor) values measure the extent of isolation of every stay point from others.

A trajectory pattern of one user represents one frequent trace of the user, which is a sequence of geographic regions that the user often travels through. There is also temporal information related to trajectory patterns, which is transition times which the user spends on transferring between consecutive regions. Thus here a trajectory pattern is denoted by a sequence of RoIs with transition times annotated.

Definition 3 (Trajectory pattern)**.** *A trajectory pattern (T-pattern) 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* S*. It can be represented by* $(S, A) = R_0 \xrightarrow{\alpha_1} R_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} R_n$*.*

If a user sequentially goes through all the RoIs of a T-pattern in a trajectory T and spends similar transition times on transferring between RoIs, then we say that 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* $(S, A) = R_0 \xrightarrow{\alpha_1} R_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} R_n$*, 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* $T' = (\langle x'_0, y'_0, t'_0\rangle, \cdots, \langle x'_n, y'_n, t'_n\rangle)$ *of* T *such that* $\forall 0 \leq i \leq n, \langle x'_i, y'_i\rangle \in R_i$ *and* $\mid \alpha_i - \alpha'_i \mid \leq \tau$ *where* $\alpha'_i = t'_i - t'_{i-1}$*.*

When a 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 $support\ value(support^D_\tau(S, A))$ to represent the percentage of trajectories containing (S, A) in the dataset D when the time tolerance is set to τ. If the support value of a T-pattern is greater than a given minimum support value, we call the pattern a *frequent T-pattern*, or *frequent pattern* for short.

The goal of the process of trajectory pattern mining is to find the set of all frequent T-patterns, named $frequent\ pattern\ set$, in a spatio-temporal dataset.

Definition 5 (Frequent pattern set)**.** *Given 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:*

$$PS^{\mathcal{T}}_{\tau,\sigma} = \{(S, A) \mid suppport^{\mathcal{T}}_\tau(S, A) \geq \sigma\}$$

A user's mobility profile describes his regular movement behavior, i.e. the traces of places that the user often visits which exactly corresponds to frequent T-patterns when the places are interpreted as RoIs. So a user's mobility profile is modeled here as the frequent pattern set of his collection of trajectories. Let $\mathcal{T}_u$ be the trajectories of user u in a dataset $\mathcal{T}$. We call $PS^{\mathcal{T}_u}_{\tau,\sigma}$ the mobility profile of u. In the following, we use PS^u to denote u's mobility profile for short by assuming τ and σ have been defined and $\mathcal{T}$ is clear from the context. Using the same rule the support value $suppport^{\mathcal{T}_u}_\tau(S, A)$ is denoted by $suppport^u(S, A)$. And below we use the function $len()$ to acquire the length of a sequence.

2.2 The MTP similarity measure [22]

After ignoring transition times, a frequent T-pattern becomes a frequent sequence pattern, and a user u's frequent pattern set PS^u becomes a *sequence pattern set* $\overline{PS}^u = \{S|\exists(S,A) \in PS^u\}$. This measure uses users' maximal sequence pattern sets instead of their whole sequence pattern sets to avoid considering repetitive behavior. Given two sequence patterns $P = (R_0, R_1, \cdots, R_n)$ and $Q = (R'_0, R'_1, \cdots, R'_m)$, Q is a subsequence of P (denoted by $Q \sqsubseteq P$) if and only if there exists $j_1 < j_2 < \cdots < j_m$ such that $R'_i = R_{j_i} (0 \leq i \leq m)$. A maximal sequence pattern set only consists of those sequence patterns, called maximal sequence patterns or maximal patterns for short, that are not subsequences of any other patterns. This method also uses the notion of longest common sequences to represent the longest common part of two sequence patterns.

Definition 6 (Maximal sequence pattern set)**.** *Given the sequence pattern set PS of the user u, his maximal sequence pattern set is:*

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

Definition 7 (Longest common sequences)**.** *Given two sequence patterns P and Q, a sequence pattern S is a longest common sequence (LCS) of P and Q if and only if the following condition is satisfied:*

$$S \sqsubseteq P \wedge S \sqsubseteq Q \wedge len(S) \geq len(S'), \forall S'(S' \sqsubseteq P \wedge S' \sqsubseteq Q)$$

Using the notion of longest common sequences, this method further defines a way to measure the similarity between two patterns. The similarity $sim(P,Q)$ between two patterns P and Q is defined as:

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

where $lenLCS(P,Q)$ is the length of the longest common sequences of P and Q and $len(P)$ is the length of P.

Given two users u and u', this method calculates the weighted average of the similarity values of all pairs of maximal sequence patterns in their maximal sequence pattern sets as the similarity value between them.

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

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

2.3 The improved MTP similarity measure [14]

In the MTP similarity measure, the similarity value of two identical users is not always one. This defect is fixed in the improved MTP similarity measure [14] by making the following change. In the MTP similarity measure each maximal sequence pattern of a user is compared with all maximal patterns of another user, while in the improved version it is only compared with the most similar pattern of another user.

Given two users u and u', this measure uses the 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 of u'.

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

Then the measure calculates the similarity values of u and u' relative to each other. The relative similarity of u to u' is defined as:

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

The overall similarity value $sim(u, u')$ between the two users is defined as:

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

This measure also introduces a way of taking transition times into account in the procedure of calculating the similarity value of two maximal patterns.

Assume that two maximal patterns $P \in M(\overline{PS}^u)$ and $Q \in M(\overline{PS}^{u'})$ have a longest common sequence $S = (R_0, R_1, \cdots, R_n)$. For any two consecutive RoIs R_{i-1} and $R_i (0 < i \leq n)$, $tranT_S^u(i)$ is the typical transition times of the user u between them, which is the union of all transition times appearing in a frequent pattern that contains S in u's profile.

$$tranT_S^u(i) = \{\alpha_i \mid \exists (S, A) \in PS_u \text{ s.t. } A = (\alpha_1, \alpha_2, \cdots, \alpha_n)\}$$

$tranT_S^u(i)$ can be denoted by a union of time intervals, e.g. $[x_1, y_1] \cup \cdots \cup [x_t, y_t]$. Then we calculate the union of $tranT_S^u(i)$ and $tranT_S^{u'}(i)$, e.g. $tranT_S^u(i) \cup tranT_S^{u'}(i) = [x_1, y_1] \cup \cdots \cup [x_k, y_k]$, and their intersection, e.g. $tranT_S^u(i) \cap tranT_S^{u'}(i) = [x'_1, y'_1] \cup \cdots \cup [x'_m, y'_m]$. Then the ratio of overlapping transition time $ot_S^{u,u'}(i)$ from R_{i-1} to R_i between u and u' is calculated as $\frac{\sum_{i=1}^{m} y'_i - x'_i}{\sum_{i=1}^{k} y_i - x_i}$.

Definition 8 (Time-overlap-fraction)**.** *Let P and Q be two maximal frequent patterns of u and u', respectively. The time-overlap-fraction of P and Q $tof(P, Q)$ is defined as:*

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

where $lcs(P, Q)$ is the set of longest common sequences of P and Q.

The similarity value of P and Q with considering transition times is calculated as:

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

2.4 The improved MTP similarity measure with semantics [15]

The improved MTP similarity measure takes semantic information into account by enriching RoIs with *location semantics*, which is a place's functionalities. Specifically, RoIs are labeled with appropriate location semantic tags, like *school* or *hospital*. Different applications have different sets of semantic tags. Ye et al.'s method [21] is used to calculate a probability distribution over the set of semantic tags for a place. The probability

of a tag represents the likelihood of a user utilizing the functionality denoted by the tag at the place.

Let $AL = \{\mu_1, \ldots, \mu_n\}$ be an ordered set of semantic tags in an application. Given an RoI R, tag_R denotes the location semantic tag of R. Furthermore, $Pr_R(\mu_i)$ is used to represent the probability that tag_R is μ_i and $\sum_{\mu \in AL} Pr_R(\mu) = 1$. Thus there is a vector of probabilities for R, e.g. $v_R = \langle p_1, \ldots, p_n \rangle$ where $p_i = Pr_R(\mu_i)$. v_R is called the *location-semantics vector (LS-vector)* of R and its ith element is denoted by $v_R(i)$, i.e. p_i.

In the MTP similarity measure, the similarity comparison of two RoIs are based on the equality of their identities. Now this is achieved by defining a distance measure between the LS-vectors of two RoIs using the notion of *Relative Entropy*.

Definition 9 (Distance between two LS-vectors)**.** *The distance $dist(v_R, v_{R'})$ between two location-semantics vectors v_R and $v_{R'}$ is the average of the relative entropy $dist_{RE}(v_R \parallel v_{R'})$ from v_R to $v_{R'}$ and the relative entropy $dist_{RE}(v_{R'} \parallel v_R)$ from $v_{R'}$ to v_R.*

$$dist(v_R, v_{R'}) = \frac{dist_{RE}(v_R \parallel v_{R'}) + dist_{RE}(v_{R'} \parallel v_R)}{2}$$

where

$$dist_{RE}(v_R \parallel v_{R'}) = \sum_{i=1}^{n} v_R(i) \cdot log \frac{v_R(i)}{v_{R'}(i)}$$

$$dist_{RE}(v_{R'} \parallel v_R) = \sum_{i=1}^{n} v_{R'}(i) \cdot log \frac{v_{R'}(i)}{v_R(i)}$$

Definition 10 (LS-similar)**.** *Two RoIs R and R' are LS-similar if and only if $dist(v_R, v_{R'}) \leq \delta$ where δ is a given threshold.*

When seeking the LCSs of two maximal sequence patterns, the length of the LCSs will increment by 1 if the LS-vectors of two RoIs from the two patterns respectively are LS-similar rather than the two RoIs are the same. In this way, semantics is taken into account in the improved MTP similarity measure.

Chapter 3

The CPS-based similarity measure

In this chapter, we first introduce several basic principles for similarity measures to follow, and then introduce our new CPS-based similarity measure, and present experimental results to show the effectiveness of our measure.

3.1 Basic principles

We model a user's movement behavior by his mobility profile ($\mathcal{M}$), and represent his mobility profile by his sequence pattern set ($\overline{PS}$) with the support values of the sequence patterns. We commence with the introduction of two concepts on sequence pattern sets.

Definition 11 (Mobility profile containment)**.** *If user u's pattern set is a subset of u''s pattern set, and for each sequence pattern of u its support value relative to u is not greater than its support value relative to u', we say that u's mobility profile is contained in u''s mobility profile, denoted by $\mathcal{M}^u \preccurlyeq \mathcal{M}^{u'}$.*

$$(\overline{PS}^u \subseteq \overline{PS}^{u'}) \wedge (\forall P \in \overline{PS}^u support^u(P) \leq support^{u'}(P)) \Rightarrow \mathcal{M}^u \preccurlyeq \mathcal{M}^{u'}$$

We define the notion of the common pattern set which contains all common sequence patterns of two users' sequence pattern sets.

Definition 12 (Common pattern set (CPS))**.** *The common pattern set $CPS(u, u')$ of two users u and u' is the intersection of their sequence pattern sets, and the support value of any pattern in $PS^{u \lhd u'}$ is equal to its support value in u's pattern set.*

$$CPS(u, u') = \overline{PS}^u \cap \overline{PS}^{u'}$$

We use $\mathcal{M}^{u \lhd u'}$ to represent the mobility profile whose sequence pattern set is $CPS(u, u')$ and the support value of each sequence pattern is its support value in u's pattern set.

Example 1. *Suppose four users whose mobility profiles are*

$$\mathcal{M}^{u_1} = \{A(0.1), C(0.2)\};$$
$$\mathcal{M}^{u_2} = \{A(0.1), C(0.3)\};$$
$$\mathcal{M}^{u_3} = \{A(0.1), B(0.2), C(0.4)\};$$
$$\mathcal{M}^{u_4} = \{A(0.3), B(0.1), D(0.2)\}.$$

For brevity we put the support value of each pattern in the parentheses following the pattern. Then $\mathcal{M}_{u_1} \preccurlyeq \mathcal{M}_{u_2} \preccurlyeq \mathcal{M}_{u_3}$. *Furthermore,*

$$\mathcal{M}^{u_3 \lhd u_4} = \{A(0.1), B(0.2)\}; \quad \mathcal{M}^{u_4 \lhd u_3} = \{A(0.3), B(0.1)\}.$$

Next we list basic principles that we expect similarity measures based on mobility profiles to satisfy.

1. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) \geq 0$
2. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) \leq 1$
3. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) = sim(\mathcal{M}^{u'}, \mathcal{M}^u)$
4. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) = 0 \Leftrightarrow CPS(u, u') = \emptyset$
5. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) = 1 \Leftrightarrow \mathcal{M}^u = \mathcal{M}^{u'}$
6. $\mathcal{M}^u \preccurlyeq \mathcal{M}^{u'} \preccurlyeq \mathcal{M}^{u''} \Rightarrow sim(\mathcal{M}^{u''}, \mathcal{M}^{u'}) \geq sim(\mathcal{M}^{u''}, \mathcal{M}^u)$
7. $sim(\mathcal{M}^u, \mathcal{M}^{u'}) > sim(\mathcal{M}^u, \mathcal{M}^{u''})$ if

$$(1) sim(\mathcal{M}^u, \mathcal{M}^{u' \lhd u})) > sim(\mathcal{M}^u, \mathcal{M}^{u'' \lhd u})) \text{ and}$$
$$(2) sim(\mathcal{M}^{u'}, \mathcal{M}^{u' \lhd u})) > sim(\mathcal{M}^{u''}, \mathcal{M}^{u'' \lhd u}).$$

The first two principles regulate the range of the similarity value between two users. Principle 3 says that user similarity is *symmetric* and principle 4 states that two users have the minimum similarity value, i.e., 0 if and only if they have no common frequent movement behavior. Principle 5 indicates that user similarity should be maximum, i.e., 1.0, when a user is compared to himself.

The last two principles are about comparing the similarity of a user to different users. The intuition of principle 6 is that users sharing more frequent movement behaviors with a user should be more similar to him than users sharing less common behaviors. For instance, in Example 1 user u_2 is more similar to u_3 than u_1 as u_2 travels pattern C more regularly than u_1. Principle 7 says that since users sharing more movement behaviors and having less different behaviors will be more similar to a user than users sharing less movement behaviors but having more different behaviors, the similarity values calculated with a valid similarity measure should be consistent with this reasoning.

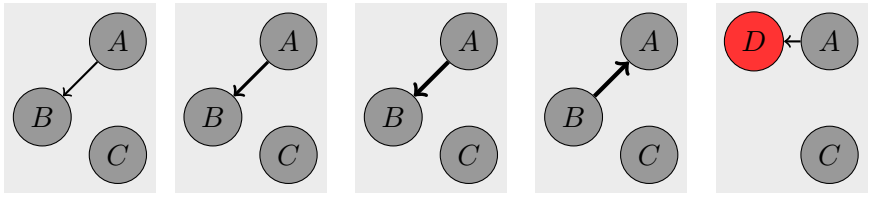
The following table summarizes which principles the MTP similarity measure and the improved version follow, respectively.

Measures	*Principle* 1	2	3	4	5	6	7
Improved MTP	√	√	√	√	×	×	×
MTP	√	√	√	√	×	×	×

We give an example to elucidate it.

Example 2. *Suppose the following five users:*

$$\mathcal{M}^{u_1} = \{A\ (0.4), B\ (0.4), C\ (0.4), A \to B\ (0.1)\};$$
$$\mathcal{M}^{u_2} = \{A\ (0.4), B\ (0.4), C\ (0.4), A \to B\ (0.2)\};$$
$$\mathcal{M}^{u_3} = \{A\ (0.4), B\ (0.4), C\ (0.4), A \to B\ (0.3)\};$$
$$\mathcal{M}^{u_4} = \{A\ (0.4), B\ (0.4), C\ (0.4), B \to A\ (0.3)\};$$
$$\mathcal{M}^{u_5} = \{A\ (0.4), C\ (0.4), D\ (0.4), A \to D\ (0.3)\}.$$



(a) u_1. (b) u_2. (c) u_3. (d) u_4. (e) u_5.

Figure 3.1: Mobility profiles in Example 2.

Figure 3.1 illustrates the mobility profiles. We use grey circles to stand for RoIs and arrows between RoIs to represent the transition direction whose thickness implies support values.

Table 3.1 shows the results given by the two similarity measures when the users are compared to each other.

	MSTP					MTP				
	u_1	u_2	u_3	u_4	u_5	u_1	u_2	u_3	u_4	u_5
u_1	0.5	0.5	0.5	0.42	0.42	1.0	1.0	1.0	0.83	0.83
u_2	0.5	0.5	0.5	0.42	0.42	1.0	1.0	1.0	0.58	0.58
u_3	0.5	0.5	0.5	0.39	0.39	1.0	1.0	1.0	0.79	0.79
u_4	0.42	0.42	0.39	0.5	0.39	0.83	0.58	0.79	1.0	0.79
u_5	0.42	0.42	0.39	0.39	0.5	0.83	0.58	0.79	0.79	1.0

Table 3.1: Pairwise user similarity in Example 2.

Obviously both measures satisfy principles 1, 2, 3 and 4. Principle 5 is violated by both measures as the similarity value between any user and himself is not 1.0, which has been pointed out by Chen et al. [14]. Principle 6 is also violated by both of them. Since $\mathcal{M}^{u_1} \preccurlyeq \mathcal{M}^{u_2} \preccurlyeq \mathcal{M}^{u_3}$ and $\mathcal{M}^{u_1} \neq \mathcal{M}^{u_2} \neq \mathcal{M}^{u_3}$, according to principle 6, we have $sim(\mathcal{M}^{u_3}, \mathcal{M}^{u_1}) < sim(\mathcal{M}^{u_3}, \mathcal{M}^{u_2})$. However, both measures produce the same similarity values for them, i.e., 0.5 and 0.1, respectively. Principle 7 does not hold for both of the measures either. Take the improved MTP measure as an example. According to its definition,

$$sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_4 \lhd u_2}) = 0.83; \quad sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_5 \lhd u_2}) = 0.86$$
$$sim(\mathcal{M}^{u_4}, \mathcal{M}^{u_4 \lhd u_2}) = 0.82; \quad sim(\mathcal{M}^{u_5}, \mathcal{M}^{u_5 \lhd u_2}) = 0.84.$$

Since $sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_5 \lhd u_2}) > sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_4 \lhd u_2})$ and furthermore $sim(\mathcal{M}^{u_5}, \mathcal{M}^{u_5 \lhd u_2}) > sim(\mathcal{M}^{u_4}, \mathcal{M}^{u_4 \lhd u_2})$, we should have $sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_5}) > sim(\mathcal{M}^{u_2}, \mathcal{M}^{u_4})$. However, the measure cannot distinguish u_2's similarity to u_4 and u_5 and outputs the same similarity value (0.58) in both cases.

Neither of the measures can give a precise measurement of similarity for all users. In the above example, from Figure 3.1 it is clear that the similarity values should decrease when comparing u_1 with the other users (from u_2 to u_5) – u_2 should be the most similar one to u_1 as they share a same set of trajectory patterns while u_5 is the least. However, they cannot distinguish this difference.

3.2 Fundamentals

The CPS-based similarity measure we propose satisfies all of the above-mentioned principles and avoids all of the defects of the (improved) MTP similarity measure. Our main idea is

1. to calculate the average percentage of the common pattern set of two users accounting for their sequence pattern sets, and

2. to calculate the similarity between the frequencies (support values) of two users' common patterns relative to the two users.

Intuitively, if a user shares more common patterns with another user and their support values are also closer, then he is more similar to this user. This idea conforms to the principles introduced in the previous section.

To measure the percentages that the common pattern set of two users accounts for either user's sequence pattern set, we associate a mobility profile with a real number, called the *magnitude*, which quantifies the amount of frequent movement behavior contained in the sequence pattern set of the mobility profile. Each sequence pattern in the pattern set contributes to the set's magnitude. The longer a frequent sequence pattern is, the harder it is for a user to have this frequent movement behavior and the more information the pattern provides about the user's behavior, and thus the greater the percentage that the pattern accounts for the user's whole movement behavior is. So patterns' lengths should carry a relatively large weight. Likewise, the greater the support value of a frequent sequence pattern is, the more frequent this movement behavior is and thus the greater the percentage that the pattern accounts for the user's whole movement behavior is. Thus we define a function f to map the mobility profile of a user to its magnitude.

$$f(\mathcal{M}^u) = \sum_{P \in \overline{PS}^u} len(P) \cdot support^u(P) \tag{3.1}$$

The percentage that the common pattern set of u and u' accounts for the sequence pattern set $\overline{PS}^u$ of u can be calculated by dividing the magnitude $f(\mathcal{M}^{u \lhd u'})$ of $\mathcal{M}^{u \lhd u'}$ by the magnitude $f(\mathcal{M}^u)$ of $\mathcal{M}^u$. Then using the notion of the geometric mean, the average percentage $AP(u, u')$ which the common pattern set accounts for $\overline{PS}^u$ and $\overline{PS}^{u'}$ is calculated as:

$$AP(u, u') = \sqrt{\frac{f(\mathcal{M}^{u \lhd u'})}{f(\mathcal{M}^u)} \cdot \frac{f(\mathcal{M}^{u' \lhd u})}{f(\mathcal{M}^{u'})}} \tag{3.2}$$

Next we explain how to measure the similarity between the two sets of support values of all common patterns. The support values of all common patterns relative to u can be regarded as a vector after they are permuted in some order. Likewise, the support values of all common patterns relative to u' can also form a vector after being permuted in the same order. Thus the problem of measuring the similarity between the two sets of support values of all common patterns is reduced to the problem of measuring the similarity of two vectors of the same length in terms of the closeness of corresponding values of the same indexes in the two vectors.

We define a function λ to map the mobility profile $\mathcal{M}^{u \lhd u'}$ to a vector by permuting the support values of the common patterns in the mobility profile in a predefined order. When the function λ is applied to $\mathcal{M}^{u' \lhd u}$ produces a vector in the same order. Provided that users u and u' have n common sequence patterns, which are $P_1, P_2, \ldots, P_n$, λ is

defined as:

$$\lambda(\mathcal{M}^{u \lhd u'}) = \langle support^u(P_1), support^u(P_2), \ldots, support^u(P_n) \rangle \tag{3.3}$$

where $CPS(u, u') = \{P_1, P_2, \ldots, P_n\}$.

We hope that the method we choose for measuring the similarity of two vectors of the same length has the following properties.

- The closer the corresponding values of the same indexes in the two vectors are, the more similar the two vectors are.
- If the differences between the corresponding values of the same indexes in a pair of vectors are the same as those in another pair of vectors, the pair of vectors in which the absolute values are greater is more similar.
- The similarity value between the two vectors is one if and only if all the corresponding values of the same indexes are the same.
- The similarity value between the two vectors is zero if and only if in each pair of corresponding values of the same index, one value is zero and the other one is positive.

We use the notion of the Bray-Curtis similarity [4] to measure the similarity between two vectors of the same length.

Definition 13 (Bray-Curtis similarity)**.** *Given two vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ of length n, the Bray-Curtis similarity between them is defined as:*

$$sim_{BC}(x, y) = 1 - \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} (x_i + y_i)}$$

Assume that the common sequence patterns of users u and u' are $P_1, P_2, \ldots, P_n$, then the similarity value between the two sets of support values of the common patterns is calculated as:

$$sim_{sup}(\lambda(\mathcal{M}^{u \lhd u'}), \lambda(\mathcal{M}^{u' \lhd u})) = 1 - \frac{\sum_{i=1}^{n} |support^u(P_i) - support^{u'}(P_i)|}{\sum_{i=1}^{n} (support^u(P_i) + support^{u'}(P_i))} \tag{3.4}$$

The similarity value between two users u and u' is defined as:

$$sim(u, u') = AP(u, u') \cdot sim_{sup}(\lambda(\mathcal{M}^{u \lhd u'}), \lambda(\mathcal{M}^{u' \lhd u})) \tag{3.5}$$

We apply our similarity measure to Example 2 and show the results in Table .

	u_1	u_2	u_3	u_4	u_5
u_1	1.0	0.96	0.93	0.76	0.50
u_2	0.96	1.0	0.97	0.71	0.47
u_3	0.93	0.97	1.0	0.67	0.44
u_4	0.76	0.71	0.67	1.0	0.44
u_5	0.50	0.47	0.44	0.44	1.0

Table 3.2: User similarity in Example 2 by our method

We can see that our measure can give more precise similarity values which reflect different extents of closeness among users. Especially, the similarity values between u_1 and the other users decrease from u_1 to u_5.

3.3 Experiments

We will present the results obtained after applying the CPS-based similarity measure to two realistic datasets Geolife and Yonsei, respectively.

The Geolife dataset is made up of 182 users collected in the Geolife project (Microsoft Research Asia) during a period of over five years (from April 2007 to August 2012). It contains 17,621 GPS trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. These trajectories have a variety of sampling rates. 91.5 percent of the trajectories are logged in a dense representation, e.g. every $1 - 5$ seconds or every $5 - 10$ meters per point. This dataset recorded a broad range of users outdoor movements, including not only regular life routines like commuting from homes to offices, but also some entertainments and sports activities, such as shopping and hiking. Although this dataset is widely distributed in over 30 cities of China and even in some cities located in the USA and Europe, the majority of the data was created in Beijing, China.

The Yonsei dataset directly contains stay point trajectories collected from commercial mobile phones over two months in Seoul, Korea. It consists of 1,865 daily trajectories from 12 users, which cover a total length of 32,626 km. It contains location information (latitude and longitude) with accuracy (error bound), Wi-Fi fingerprints (MAC address and signal strength of surrounding Wi-Fi APs), user-defined types of places (workplace, cafeteria, etc.). These trajectories were continuously recorded every 2 to 5 minutes for everyday location monitoring.

We picked seven users from the Geolife dataset, two of which have relatively larger numbers of trajectories than other users and therefore are split into more sub-users. So our Geolife testing dataset consists of 10 users, in which users 113, 151 and 160 are from one original user, and users 163, 164 are from another original user. Similarly, our Yonsei testing dataset also consists of 10 users, in which users 081 and 082 are from one original user, and users 121, 122 are from another original user. The original user from which users 081 and 082 come had very different movement behavioral patterns during the first half and second half periods, so the similarity between users 081 and 082 is not high although they come from one user.

3.3.1 The experiment on the Geolife dataset

We list our result and the result of the improved MTP similarity measure to compare them.

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.69	0.05	0.28	0.00	0.07	0.07	0.04	0.10	0.12
004	0.69	1.00	0.04	0.24	0.00	0.06	0.06	0.04	0.08	0.10
017	0.05	0.04	1.00	0.05	0.00	0.02	0.02	0.01	0.06	0.08
030	0.28	0.24	0.05	1.00	0.00	0.09	0.08	0.03	0.09	0.11
068	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
113	0.07	0.06	0.02	0.09	0.00	1.00	0.20	0.03	0.08	0.09
151	0.07	0.06	0.02	0.08	0.00	0.02	1.00	0.56	0.37	0.39
160	0.04	0.04	0.01	0.03	0.00	0.03	0.56	1.00	0.32	0.31
163	0.10	0.08	0.06	0.09	0.00	0.08	0.37	0.32	1.00	0.79
164	0.12	0.10	0.08	0.11	0.00	0.09	0.39	0.31	0.79	1.00

Table 3.3: Result of the CPS-based similarity measure

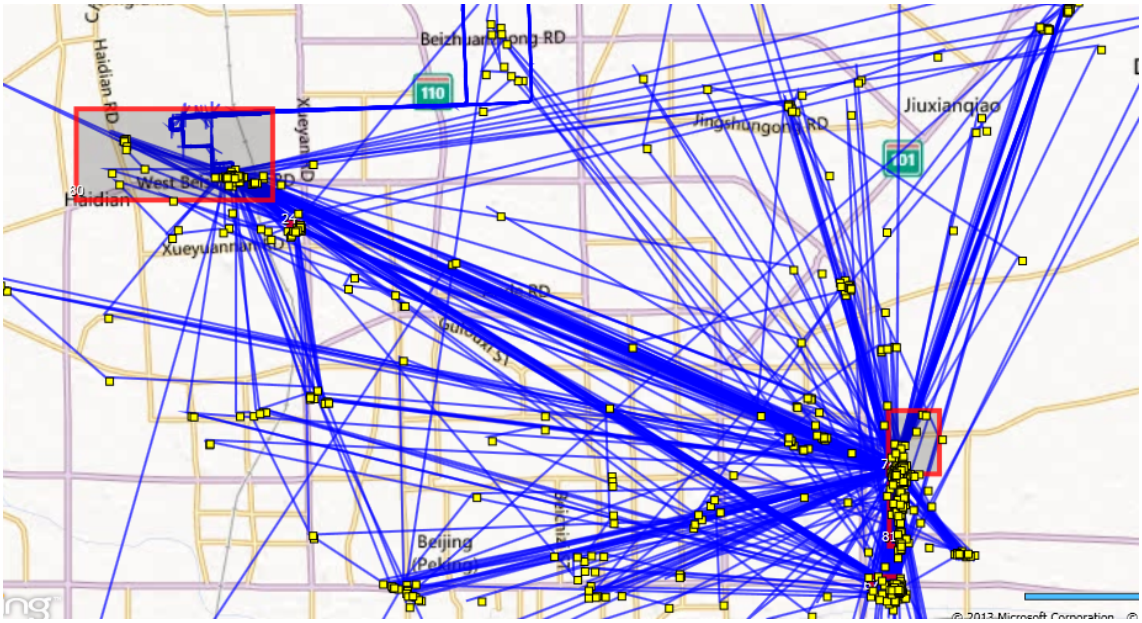
	003	004	017	030	068	113	151	160	163	164
003	1.00	0.53	0.10	0.51	0.00	0.14	0.14	0.15	0.12	0.13
004	0.53	1.00	0.17	0.36	0.00	0.23	0.24	0.24	0.21	0.23
017	0.10	0.17	1.00	0.09	0.00	0.13	0.11	0.11	0.14	0.14
030	0.51	0.36	0.09	1.00	0.00	0.27	0.23	0.10	0.12	0.12
068	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
113	0.14	0.23	0.13	0.27	0.00	1.00	0.41	0.35	0.52	0.42
151	0.14	0.24	0.11	0.23	0.00	0.41	1.00	0.74	0.36	0.48
160	0.15	0.24	0.11	0.10	0.00	0.35	0.74	1.00	0.43	0.55
163	0.12	0.21	0.14	0.12	0.00	0.52	0.36	0.43	1.00	0.89
164	0.13	0.23	0.14	0.12	0.00	0.42	0.48	0.55	0.89	1.00

Table 3.4: Result of the improved MTP similarity measure

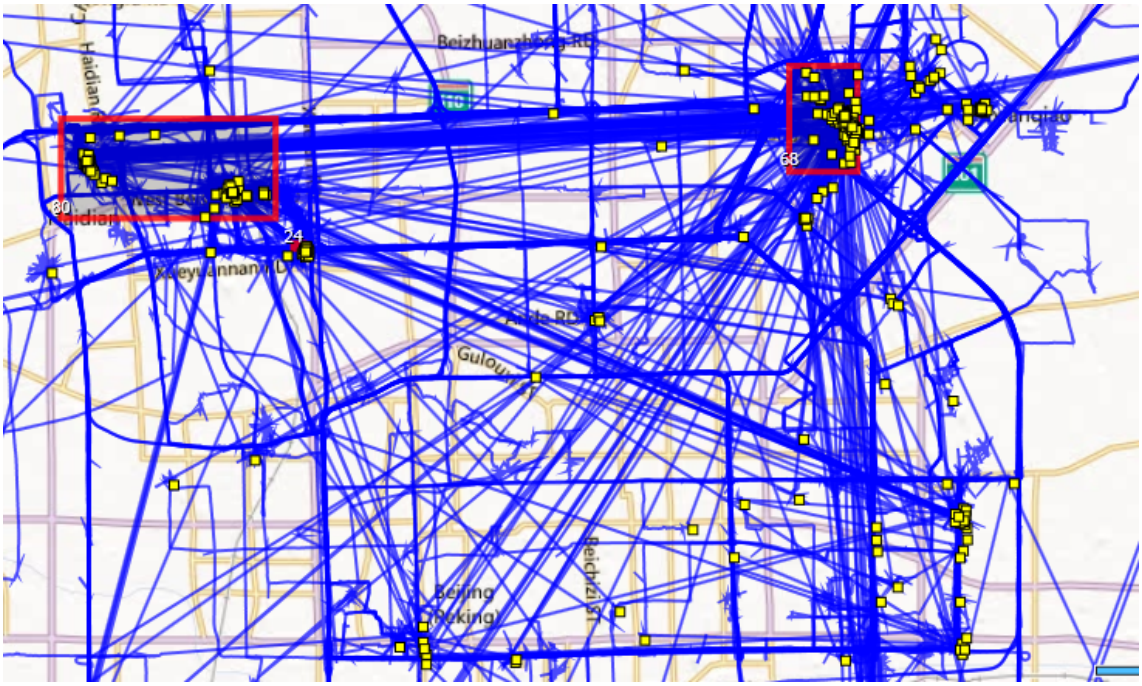
We see that in general the similarity values given by the CPS-based measure are smaller than those given by the improved MTP measure.

The results of our measure are more reasonable. We take the example of users 113 and 163 to elucidate it. We plot their trajectories on the map in Figure 3.2. RoIs are labeled by red rectangles, stay points are labeled by yellow dots, and blue lines represents stay point trajectories. The number in the lower-left corner of each red rectangle is the ID number of the RoI.

From Figure 3.2 we can see that the two users have two common RoIs (80 and 24) in the upper-left corner of the map. The vast majority of user 113's trajectories pass the two common RoIs and the three RoIs which are in the lower-right corner of the map, while most trajectories of user 163 pass the two common RoIs and the RoI (68) which is in the upper-left corner. Thus we know that the two users are very different, and the similarity value 0.08 given by our similarity measure is more reasonable than the value 0.52 given by the improved MTP measure.



(a) 113



(b) 163

Figure 3.2: Trajectories of users 113 and 163

3.3.2 The experiment on the Yonsei dataset

Below we give the results of our similarity measure on the Yonsei dataset. The results of the improved MTP similarity measure on this dataset are listed as well.

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.11	0.07	0.07	0.10	0.15	0.05	0.07	0.10	0.11
002	0.11	1.00	0.08	0.08	0.14	0.22	0.06	0.08	0.14	0.14
003	0.07	0.08	1.00	0.32	0.07	0.11	0.03	0.05	0.08	0.08
004	0.07	0.08	0.32	1.00	0.07	0.11	0.05	0.07	0.07	0.08
007	0.10	0.14	0.07	0.07	1.00	0.20	0.06	0.07	0.12	0.13
009	0.15	0.22	0.11	0.11	0.20	1.00	0.09	0.11	0.19	0.20
081	0.05	0.06	0.03	0.05	0.06	0.09	1.00	0.15	0.06	0.06
082	0.07	0.08	0.05	0.07	0.07	0.11	0.15	1.00	0.07	0.08
121	0.10	0.14	0.08	0.07	0.12	0.19	0.06	0.07	1.00	0.88
122	0.11	0.14	0.08	0.08	0.13	0.20	0.06	0.08	0.88	1.00

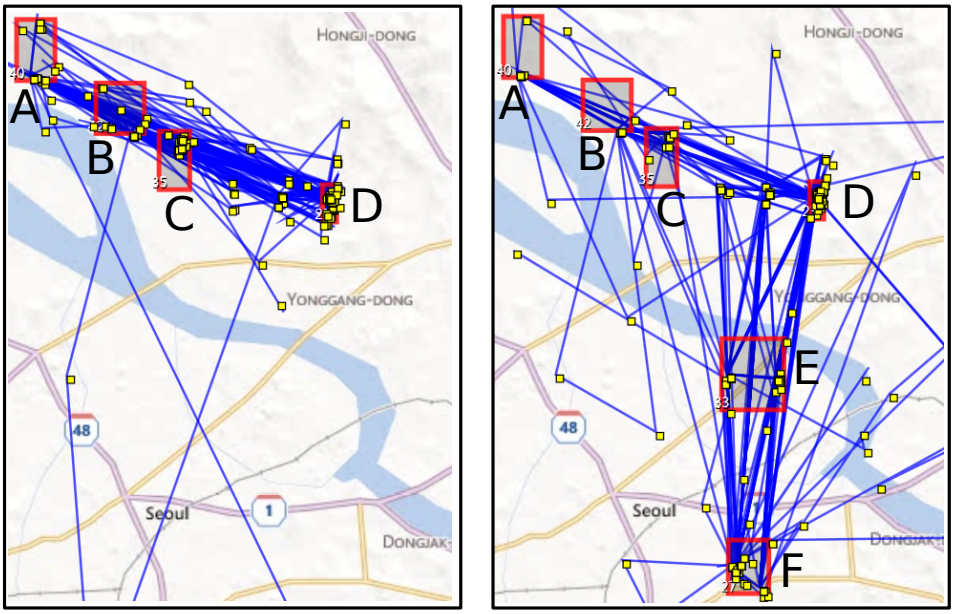
Table 3.5: Result of the CPS-based similarity measure

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.34	0.19	0.17	0.29	0.35	0.18	0.27	0.17	0.23
002	0.34	1.00	0.37	0.30	0.44	0.47	0.31	0.36	0.31	0.39
003	0.19	0.37	1.00	0.51	0.32	0.38	0.21	0.29	0.20	0.27
004	0.17	0.30	0.51	1.00	0.26	0.30	0.39	0.35	0.19	0.23
007	0.29	0.44	0.32	0.26	1.00	0.47	0.27	0.36	0.26	0.34
009	0.35	0.47	0.38	0.30	0.47	1.00	0.31	0.38	0.31	0.40
081	0.18	0.31	0.21	0.39	0.27	0.31	1.00	0.69	0.19	0.24
082	0.27	0.36	0.29	0.35	0.36	0.38	0.69	1.00	0.25	0.31
121	0.17	0.31	0.20	0.19	0.26	0.31	0.19	0.25	1.00	0.83
122	0.23	0.39	0.27	0.23	0.34	0.40	0.24	0.31	0.83	1.00

Table 3.6: Result of the improved MTP similarity measure

From the two tables we can see a trend, which is that the similarity values of all pairs of users given by our similarity measure are way smaller than those given by the improved MTP measure.

Our results are more reasonable, and we take the example of users 081 and 082 to explain the reason. We plot their trajectories on the map in the following figure. RoIs are labeled by red rectangles, stay points are labeled by yellow dots, and blue lines represents stay point trajectories. The letter near each red rectangle is the identity of the RoI.



(a) 081 (b) 082

Figure 3.3: Trajectories of users 081 and 082

From the figure, we see that user 082 has two more RoIs which do not overlap any RoI of user 081 than user 081. In addition, more than 57% trajectories of user 082 pass these two RoIs and only about 15% of his trajectories contain RoIs A, B and C, while about 78% of 08^*'s trajectories contain A, B and C. Therefore, the reasonable similarity value between $08^{\#}$ and 08^* should be around 0.20 after considering the small proportion of common patterns and the large difference between their support values. So we see that the similarity value 0.15 given by the CPS-based measure is more reasonable than the value 0.69 given by the improved MTP measure.

Chapter 4

The CPS-based similarity measure with semantics

The CPS-based similarity measure introduced in Chapter 3 measures user similarity only in terms of the extent of geographic proximity between two users' movement trajectories.

In social networks user recommendation is primarily based on users' interests. So we want similarity measures to be able to find users with similar interests no matter whether their movement trajectories are geographically close. Similar users which are found using the CPS-based similarity measure not only have similar interests, but also have geographically close movement trajectories. However, the CPS-based measure is not able to find similar users whose interests are similar, but movement trajectories are not close geographically.

For example, two users live in different cities and both of them are fond of movies and eating. Their movement trajectories will be far away from each other and their mobility profiles will not have common sequence patterns. Thus although the two users have similar interests or hobbies, they will not be similar using the CPS-based similarity measure. However, when places are tagged with their functionalities, e.g. cinema and restaurant, we will be able to detect their similarity. The functionality of a place indicates its semantics, called location semantics.

This indicates that it is necessary to take location semantics into account in the CPS-based similarity measure. In this chapter we will propose a mechanism that can incorporate semantics into the CPS-based similarity measure.

4.1 Fundamentals

The semantic information we use is in the form of LS-vectors. Recall that the notion of LS-vectors is introduced in Chapter 2. When each RoI in a frequent sequence pattern is replaced with its associated LS-vector, we get an LS-vector sequence pattern.

Definition 14 (LS-vector sequence patterns)**.** *An LS-vector sequence pattern (or LS-vector pattern for short) P_v is a sequence of LS-vectors $(v_{R_1}, v_{R_2}, \ldots, v_{R_n})(n \geq 0)$. It can be represented as $P_v = v_{R_1} \rightarrow v_{R_2} \rightarrow \cdots \rightarrow v_{R_n}$.*

Usually a user only makes use of one functionality of a place during a stay. If each RoI in a frequent sequence pattern of a user is replaced with one functionality used by the user when he visited the region, we obtain a sequence of functionalities used by the user, which is a sequence of semantic tags. Sequences of semantic tags are also a form

of representation of the user's movement behavior, and we call them semantic sequence patterns.

Definition 15 (Semantic sequence patterns)**.** *A semantic sequence pattern (or semantic pattern for short) P_s is a sequence of semantic tags $(\mu_{a_1}, \mu_{a_2}, \ldots, \mu_{a_n})(n \geq 0)$. It can be represented as $P_s = \mu_{a_1} \rightarrow \mu_{a_2} \rightarrow \cdots \rightarrow \mu_{a_n}$.*

Now we introduce in detail how to take semantics into consideration in the CPS-based similarity measure.

Step 1: After obtaining the non-repetitive support values of a user's sequence patterns in the same way as the CPS-based measure in Chapter 3, we convert each sequence pattern to an LS-vector sequence pattern by replacing each RoI with its associated LS-vector. We call the set of all the LS-vector sequence patterns of user u his *LS-vector sequence pattern set* denoted by $\overline{PS}_v^u$. We define a function $\zeta : \overline{PS}^u \rightarrow \overline{PS}_v^u$ to map a sequence pattern $P = R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_n$ to an LS-vector sequence pattern P_v.

$$P_v = \zeta(P) = v_{R_1} \rightarrow v_{R_2} \rightarrow \cdots \rightarrow v_{R_n} \tag{4.1}$$

where v_{R_i} is the LS-vector associated with the RoI R_i.

Step 2: Further we convert each LS-vector sequence pattern to a set of semantic sequence patterns. This can be done in two sub-steps.

First, all semantic tags with non-zero probabilities are drawn out and form a set from each LS-vector of an LS-vector pattern. Suppose that there are h semantic tags in total in the dataset, we define a function η to map an LS-vector $v_R = \langle p_1^R, p_2^R, \ldots, p_h^R \rangle$ to a set of semantic tags s_R.

$$s_R = \eta(v_R) = \{\mu_i | p_i^R \neq 0\}, \forall p_i^R \in v_R \tag{4.2}$$

Second, after obtaining the sets of semantic tags of all the LS-vectors in the pattern, we calculate their Cartesian product, which is a set of semantic sequence patterns, in the order that the LS-vectors appear in the pattern. This can be formulated as:

$$\prod_{i=1}^{n} s_{R_i} = \{(\mu_{a_1}, \mu_{a_2}, \ldots, \mu_{a_n}) | \mu_{a_1} \in s_{R_1} \wedge \mu_{a_2} \in s_{R_2} \wedge \ldots \wedge \mu_{a_n} \in s_{R_n}\} \tag{4.3}$$

The probability associated with each semantic tag in a semantic pattern means the likelihood of the user using the functionality represented by the semantic tag at the RoI from which the tag comes. Recall that each frequent sequence pattern corresponds to a set of trajectories that spatio-temporally contain it, so the product of the probabilities of all the semantic tags in a semantic pattern refers to the percentage of trajectories in which the user goes through the sequence of functionalities represented by the semantic pattern in the set of trajectories corresponding to the sequence pattern from which the semantic pattern comes.

So we can calculate the support value of each semantic pattern by multiplying all the probabilities included in the pattern and the support value of the original sequence pattern. The support value of a semantic pattern refers to the percentage of trajectories in which the user goes through the sequence of functionalities represented by the semantic pattern in all of the user's trajectories. The support value $support(P_s)$ of a semantic pattern $P_s = (\mu_{a_1}, \mu_{a_2}, \ldots, \mu_{a_n})$ of user u is calculated as:

$$support^u(P_s) = support^u(P) \cdot p_{a_1}^{R_1} \cdot p_{a_2}^{R_2} \cdot \ldots \cdot p_{a_n}^{R_n} \tag{4.4}$$

We give an example to clarify Step 2.

Example 3. *Assume that the ordered set of semantic tags in the dataset is* $\{hotel, restaurant, school, hospital\}$*, and a user's sequence pattern set includes one pattern* $A \rightarrow B$ *with the support value 0.5. The LS-vector of the RoI A is* $\langle 0.1, 0.9, 0.0, 0.0\rangle$*, and the LS-vector of the RoI B is* $\langle 0.0, 0.0, 0.9, 0.1\rangle$*.*

So we can obtain an LS-vector sequence pattern $\langle 0.1, 0.9, 0.0, 0.0\rangle \rightarrow \langle 0.0, 0.0, 0.9, 0.1\rangle$ *with the support value 0.5. After applying the above procedure, the four semantic sequence patterns are: (We put the probability associated with each semantic tag in the braces following the tag.)*

hotel (0.1) → *school* (0.9) *hotel* (0.1) → *hospital* (0.1)

restaurant (0.9) → *school* (0.9) *restaurant* (0.9) → *hospital* (0.1)

The support values of the above four semantic tag patterns are 0.045, 0.005, 0.405 and 0.045, respectively.

Step 3:

Note that two different LS-vector patterns can generate two equal semantic patterns. So the next task we will do is to merge equal semantic patterns generated from a user's sequence pattern set by adding up their support values so that the resultant set of semantic patterns does not contain any two equal semantic patterns. We call the set of all the semantic patterns of user u his *semantic sequence pattern set* denoted by $\overline{PS}_s^u$.

Step 4: Like the common pattern set, we define the notion of the *common semantic pattern set* which contains all common semantic sequence patterns of two users.

Definition 16 (Common semantic pattern set)**.** *The common semantic pattern set* $CPS_s(u, u')$ *of two users* u *and* u' *is the intersection of their semantic sequence patterns sets.*

$$CPS_s(u, u') = \overline{PS}_s^u \cap \overline{PS}_s^{u'}$$

We use $\mathcal{M}_s^{u \lhd u'}$ to represent the mobility profile whose semantic sequence pattern set is $CPS_s(u, u')$ and the support value of each semantic pattern is its support value in u's semantic pattern set.

The following procedure remains the same as the CPS-based similarity measure in Chapter 3, except that the sequence patterns are substituted by semantic sequence patterns. This means Equations (3.2) and (3.5) are changed.

$$AP(u, u') = \sqrt{\frac{f(\mathcal{M}_s^{u \lhd u'})}{f(\mathcal{M}_s^u)} \cdot \frac{f(\mathcal{M}_s^{u' \lhd u})}{f(\mathcal{M}_s^{u'})}} \tag{4.5}$$

$$sim(u, u') = AP(u, u') \cdot sim_{sup}(\lambda(\mathcal{M}_s^{u \lhd u'}), \lambda(\mathcal{M}_s^{u' \lhd u})) \tag{4.6}$$

The above four steps consists of the basic algorithm of the CPS-based similarity measure with semantics. However, there is a problem that we need to deal with.

The semantic patterns with relatively low support values in a user's semantic pattern set represent his infrequent or even noisy behavior. When comparing two users' similarity using this measure, we should not take infrequent behavior into account, since infrequent behavior cannot represent users' typical behavior patterns and will interfere with the comparison of user similarity. Thus we need to remove the semantic patterns with fairly low support values from users' semantic pattern sets. There are two ways that can achieve this goal.

- We set a threshold ϵ on the probabilities associated with semantic tags; that is, in Step 2 when converting LS-vector patterns to semantic patterns, we do not draw out those semantic tags with probabilities less than ϵ.

  The idea behind this action is that the fact that the support value of a semantic pattern is relatively low is attributed to the fact that the probabilities associated with some semantic tags in the pattern are relatively small. The fact that the probability associated with a semantic tag is relatively small is either because the likelihood of a user utilizing the functionality represented by the tag at the place is indeed small, or because the method of generating LS-vectors is imprecise so that the functionality which the place does not have is assigned a non-zero probability. Thus ϵ should be set to a proper value so that these relatively small probabilities will be removed.

  So this first way of removing infrequent behavior means that Equation (4.2) will be changed.

$$s_R = \eta(v_R) = \{\mu_i | p_i^R \geq \epsilon\}, \forall p_i^R \in v_R \tag{4.7}$$

  Example 4. *In Example 3, if we use 0.2 as the threshold ϵ, then we will finally obtain only one semantic pattern of length 2, which is restaurant* (0.9) $\rightarrow$ *school* (0.9) *with the support value 0.405.*

  In this first way of removing infrequent behavior, there is also a technique that can be applied to optimize the procedure of converting LS-vector sequence patterns to semantic sequence patterns. It is based on the observation that if the support value of a semantic sequence pattern is below the threshold, the support value of any longer pattern which has that pattern as a prefix must be less than the threshold as well. So after obtaining the semantic patterns of length $k(k > 0)$, for every semantic pattern of length k we can construct semantic patterns of length $k + 1$ by checking if each frequent pattern of length $k + 1$ can generate semantic patterns prefixed with the specific semantic pattern of length k. In this way we do not need to generate semantic patterns of length $k + 1$ from scratch. We give the algorithm in the following figure.

Algorithm 1: Converting LS-vector patterns to semantic patterns

Input: The LS-vector pattern set PS_v^u of user u
Output: The semantic pattern set PS_s^u of user u

```
generate semantic patterns of length 1 from frequent patterns of length 1 in PS_v^u;
foreach semantic pattern P_s of length k(k > 0) do
    foreach frequent pattern P of length k + 1 in PS_v^u do
        if P can generate semantic patterns prefixed with P_s then
            foreach non-zero probability p_i in the last LS-vector of P do
                add the concatenation of P_s and v_i to PS_s^u;
            end
        end
    end
end
return PS_s^u
```

- After performing Step 3 and thus obtaining the semantic pattern set, for the patterns of each length, we set a different threshold and remove those patterns of that

length whose non-repetitive support values are below the threshold. If there are h semantic tags and p patterns of length k in a user's frequent sequence pattern set, and the support value threshold used when extracting the user's frequent T-patterns is t, then the threshold we use for these semantic patterns of length k is calculated as:

$$\theta(k, h, p, t) = \frac{t}{h^k} \cdot p \tag{4.8}$$

When h, p and t are clear from the context, we use $\theta(k)$ to denote $\theta(k, h, p, t)$ for short. The threshold $\theta(k, h, p, t)$ is the support value of a semantic pattern of length k when the probability of each semantic tag is the average probability $\frac{1}{h}$ of a tag in an LS-vector.

So this second way of removing infrequent behavior means that after obtaining the semantic pattern set $\overline{PS}_s^u$ in Step 3, we add the following step.

$$\overline{PS}_{s,\theta}^u = \{P_s | nr\text{–}support(P_s) \geq \theta(len(P_s))\}, \forall P_s \in \overline{PS}_s^u \tag{4.9}$$

And we will use $\overline{PS}_{s,\theta}^u$ instead of $\overline{PS}_s^u$ in Step 4.

Example 5. *In Example 3, if the support value threshold used when extracting the user's frequent T-patterns is 0.1, according to Equation* (4.8) *the threshold θ for semantic patterns of length 2 is 0.00625. Thus there will be three semantic patterns of length 2, which are:*

$$restaurant\ (0.9) \rightarrow school\ (0.9)$$
$$restaurant\ (0.9) \rightarrow hospital\ (0.1)$$
$$hotel\ (0.1) \rightarrow school\ (0.9)$$

4.2 Experiments

We conducted experiments on the same two datasets as those used in Chapter 3. There are nine semantic tags in total and the probabilities in each RoI's LS-vector are randomly generated and normally distributed. This is achieved by randomly choosing nine points on the normal distribution curve in the range [0,5] of the horizontal axis and then normalizing their values of vertical axis by dividing each value by the sum of all these values.

The experiments are conducted on a desktop computer with Intel Pentium Dual-core CPU of 1.73GHz, 2GB memory and Windows XP Professional SP3. In the following table we give the running times using the two ways of removing infrequent behavior. The running time on either dataset using the first way of removing infrequent behavior is the average of the remaining running times after removing three longest and three shortest running times from ten tests on the dataset. The running time on either dataset using the second way of removing infrequent behavior is the average of the remaining running times after removing one longest and one shortest running time from five tests on the dataset.

	The first way	The second way
Geolife	0.33s	97.54s
Yonsei	0.31s	79.45s

From the table we see that the algorithm using the first way of removing infrequent behavior runs much faster than that using the second way.

Below we present our results and the results of the improved MTP similarity measure with semantics.

4.2.1 The experiment on the Geolife dataset

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.54	0.16	0.58	0.22	0.12	0.25	0.25	0.20	0.27
004	0.54	1.00	0.13	0.34	0.22	0.09	0.19	0.21	0.13	0.23
017	0.16	0.13	1.00	0.14	0.11	0.21	0.23	0.15	0.17	0.14
030	0.58	0.34	0.14	1.00	0.21	0.16	0.22	0.22	0.16	0.20
068	0.22	0.22	0.11	0.21	1.00	0.24	0.24	0.13	0.11	0.22
113	0.12	0.09	0.21	0.16	0.24	1.00	0.44	0.15	0.10	0.13
151	0.25	0.19	0.23	0.22	0.24	0.44	1.00	0.56	0.57	0.39
160	0.25	0.21	0.15	0.22	0.13	0.15	0.56	1.00	0.47	0.32
163	0.20	0.13	0.17	0.16	0.11	0.10	0.57	0.47	1.00	0.52
164	0.27	0.23	0.14	0.20	0.22	0.13	0.39	0.32	0.52	1.00

Table 4.1: Result using the first way of removing infrequent behavior and $\epsilon = \frac{1}{9}$

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.50	0.20	0.52	0.23	0.18	0.23	0.25	0.19	0.23
004	0.50	1.00	0.12	0.27	0.20	0.11	0.14	0.18	0.11	0.17
017	0.20	0.12	1.00	0.17	0.13	0.25	0.24	0.19	0.17	0.10
030	0.52	0.27	0.17	1.00	0.18	0.15	0.16	0.16	0.13	0.13
068	0.23	0.20	0.13	0.18	1.00	0.26	0.24	0.16	0.08	0.17
113	0.18	0.11	0.25	0.15	0.26	1.00	0.48	0.37	0.19	0.14
151	0.23	0.14	0.24	0.16	0.24	0.48	1.00	0.62	0.59	0.38
160	0.25	0.18	0.19	0.16	0.16	0.37	0.62	1.00	0.51	0.29
163	0.19	0.11	0.17	0.13	0.08	0.19	0.59	0.51	1.00	0.48
164	0.23	0.17	0.10	0.13	0.17	0.14	0.38	0.29	0.48	1.00

Table 4.2: Result using the second way of removing infrequent behavior

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.75	0.27	0.84	0.00	0.00	0.00	0.00	0.00	0.00
004	0.75	1.00	0.32	0.71	0.11	0.00	0.00	0.00	0.00	0.33
017	0.27	0.32	1.00	0.27	0.46	0.25	0.11	0.00	0.45	0.33
030	0.84	0.71	0.27	1.00	0.00	0.27	0.09	0.00	0.00	0.00
068	0.00	0.11	0.46	0.00	1.00	0.00	0.00	0.00	0.00	0.28
113	0.00	0.00	0.25	0.27	0.00	1.00	0.53	0.41	0.45	0.50
151	0.00	0.00	0.11	0.10	0.00	0.53	1.00	0.82	0.56	0.54
160	0.00	0.00	0.00	0.00	0.00	0.41	0.82	1.00	0.86	0.61
163	0.00	0.00	0.45	0.00	0.00	0.45	0.56	0.86	1.00	0.92
164	0.00	0.33	0.33	0.00	0.28	0.50	0.54	0.61	0.92	1.00

Table 4.3: Result of the improved MTP similarity measure with semantics when the distance threshold is 0.5

Comparing Table 4.1 / 4.2 with Table 3.3, we see two trends between the CPS-based measure and its semantic version. First, some pairs of users whose similarity values are 0 when without considering semantics have some degree of similarity after considering semantics. Second, the similarity values of almost all pairs of users become greater after considering semantics than without considering semantics.

Comparing Table 4.1 with Table 4.2, for the majority of pairs of users the similarity values given by the two variations of the CPS-based measure with semantics are very close.

Comparing Table 4.1 / 4.2 with Table 4.3, we see that other than those pairs of users whose similarity values given by the improved MTP similarity measure with semantics are 0, for the majority of the remaining pairs of users the similarity values given by both variations of our measure are smaller than those given by the improved MTP similarity measure with semantics.

Our results are more reasonable than those of the improved MTP similarity measure with semantics. Next we give two corroborating examples.

We take the example of users 003 and 163. The two users' semantic pattern sets have some common patterns, so they should have a certain degree of similarity. From Figure 4.3, we see that the similarity value given by the improved MTP measure with semantics is 0, which is evidently wrong. The same circumstance occurs to all the pairs of users whose similarity values given by the improved MTP measure are zero.

We take another example of users 163 and 164. Since they have a large number of semantic patterns, we do not list all of their semantic patterns in a table. Instead we give some statistics of their semantic pattern sets. Using the first way of removing infrequent behavior, users 163 and 164 have 11 and 23 frequent semantic patterns, respectively. And they have 11 common patterns. Using the second way of removing infrequent behavior, users 163 and 164 have 33 and 38 frequent semantic patterns, respectively. And they have 21 common patterns. So we know that the average percentage of the common patterns accounting for all of their patterns cannot be as high as 0.92 given by the improved MTP measure with semantics. The similarity value between the two sets of non-repetitive support values of the common patterns will make the similarity value between the two users be smaller than the average percentage. So the similarity value between the two users must be much lower than 0.92 given by the improved MTP measure with semantics.

4.2.2 The experiment on the Yonsei dataset

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.29	0.30	0.28	0.13	0.34	0.13	0.28	0.33	0.29
002	0.29	1.00	0.20	0.24	0.16	0.33	0.12	0.31	0.28	0.24
003	0.30	0.20	1.00	0.40	0.16	0.16	0.02	0.17	0.36	0.26
004	0.28	0.24	0.40	1.00	0.25	0.18	0.04	0.23	0.34	0.29
007	0.13	0.16	0.16	0.25	1.00	0.21	0.06	0.18	0.24	0.31
009	0.34	0.33	0.16	0.18	0.21	1.00	0.19	0.40	0.23	0.17
081	0.13	0.12	0.02	0.04	0.06	0.19	1.00	0.28	0.09	0.08
082	0.28	0.31	0.17	0.23	0.18	0.40	0.28	1.00	0.26	0.21
121	0.33	0.28	0.36	0.34	0.24	0.23	0.09	0.26	1.00	0.76
122	0.29	0.24	0.26	0.29	0.31	0.17	0.08	0.21	0.76	1.00

Table 4.4: Result using the first way of removing infrequent behavior and $\epsilon = \frac{1}{9}$

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.35	0.29	0.29	0.17	0.36	0.07	0.18	0.28	0.25
002	0.35	1.00	0.28	0.27	0.16	0.31	0.05	0.19	0.29	0.22
003	0.29	0.28	1.00	0.36	0.20	0.17	0.01	0.12	0.35	0.25
004	0.29	0.27	0.36	1.00	0.27	0.17	0.02	0.15	0.28	0.25
007	0.17	0.16	0.20	0.27	1.00	0.20	0.04	0.14	0.27	0.35
009	0.36	0.31	0.17	0.17	0.20	1.00	0.14	0.33	0.20	0.16
081	0.07	0.05	0.01	0.02	0.04	0.14	1.00	0.23	0.05	0.06
082	0.18	0.19	0.12	0.15	0.14	0.33	0.23	1.00	0.20	0.16
121	0.28	0.29	0.35	0.28	0.27	0.20	0.05	0.20	1.00	0.76
122	0.25	0.22	0.25	0.25	0.35	0.16	0.06	0.16	0.76	1.00

Table 4.5: Result using the second way of removing infrequent behavior

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.56	0.36	0.31	0.35	0.44	0.31	0.34	0.33	0.34
002	0.56	1.00	0.45	0.38	0.38	0.39	0.33	0.41	0.38	0.32
003	0.36	0.45	1.00	0.72	0.38	0.48	0.31	0.37	0.36	0.37
004	0.31	0.38	0.72	1.00	0.33	0.33	0.29	0.35	0.33	0.28
007	0.35	0.38	0.38	0.33	1.00	0.64	0.32	0.52	0.54	0.59
009	0.44	0.39	0.48	0.33	0.64	1.00	0.39	0.42	0.45	0.50
081	0.31	0.33	0.31	0.29	0.32	0.39	1.00	0.69	0.29	0.31
082	0.34	0.41	0.37	0.35	0.52	0.42	0.69	1.00	0.38	0.41
121	0.33	0.38	0.36	0.33	0.54	0.45	0.29	0.38	1.00	0.94
122	0.34	0.32	0.37	0.28	0.59	0.50	0.31	0.41	0.94	1.00

Table 4.6: Result of the improved MTP similarity measure with semantics when the distance threshold is 0.5

Comparing Table 4.4 / 4.5 with Table 4.6, we see that the results of both variations of our measure are smaller than those of the improved MTP similarity measure with semantics.

Likewise, our results are more reasonable than the improved MTP similarity measure with semantics. Next we give two corroborating examples.

We take the example of the users 003 and 081. We will not list their semantic patterns in a table, since the numbers of those patterns are too large. Using the first way of removing infrequent behavior, users 003 and 081 have 51 and 252 frequent semantic patterns, respectively. And they have 10 common patterns. Using the second way of removing infrequent behavior, users 003 and 081 have 158 and 769 frequent semantic patterns, respectively. And they have 7 common patterns. So we know that the average percentage of the common semantic patterns accounting for all of their patterns is rather small and must be below 0.31 given by the improved MTP similarity measure. After considering the similarity between the two sets of non-repetitive support values of the common patterns, the similarity value between the two users will become smaller than the average percentage. So the similarity value between the two users cannot be as high as 0.31 given by the improved MTP similarity measure.

Take another example of users 007 and 009. Using the first way of removing infrequent behavior, users 007 and 009 have 150 and 36 frequent semantic patterns, respectively. And they have 26 common patterns. Using the second way of removing infrequent behavior, users 007 and 009 have 228 and 41 frequent semantic patterns, respectively.

And they have 22 common patterns. So we know that the average percentage of the common semantic patterns accounting for all of their patterns must be below 0.5. Likewise, after considering the similarity between the two sets of non-repetitive support values of the common patterns, the similarity value between the two users will become smaller than the average percentage. So the similarity value between the two users cannot be as high as 0.64 given by the improved MTP similarity measure.

Chapter 5

The Hausdorff distance-based similarity measure with semantics

The CPS-based similarity measure with semantics removes some infrequent or noisy behavior and only uses users' frequent behavior. So if we want to compare users using their whole behavior, we need to devise another similarity measure. The solution we come up with is the novel Hausdorff distance-based similarity measure.

In this chapter we introduce the Hausdorff distance-based similarity measure that takes semantics into account, and present its experimental results to show the effectiveness of the measure.

5.1 Fundamentals

The basic idea is to measure the distance of two users' LS-vector sequence pattern sets based on a variation of Hausdorff distance.

The Hausdorff distance measures how far two subsets of a metric space are from each other. Informally, two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set. It is the greatest of all the distances from a point in one set to the closest point in the other set. [8]

Definition 17 (Hausdorff distance)**.** *Let X and Y be two non-empty subsets of a metric space. Their Hausdorff distance $d_H(X, Y)$ is defined by:*

$$d_H(X, Y) = \max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\}$$

We modify the original definition of Hausdorff distance by replacing the max operation with the averaging operation. Either of the two components in the max operation denotes the distance from one set to the other one, so using the max operation to obtain the overall distance between the two sets will go to extremes and at most times the overall distance obtained will be unreasonably greater than our expectation.

Definition 18 (modified Hausdorff distance)**.** *Let X and Y be two non-empty subsets of a metric space. Their modified Hausdorff distance $d_{mH}(X, Y)$ is defined by:*

$$d_{mH}(X, Y) = avg\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\}$$

The calculation of the modified Hausdorff distance consists of three phases.

- The first phase is to calculate the distance between every element x in the set X and the set Y by taking the minimum value of all the distances between x and every element in Y, and similarly to calculate the distance between every element y in Y and X.
- The second phase is to calculate the distance from X to Y by taking the maximum value of all the distances between every element x in X and Y, and similarly to calculate the distance from Y to X.
- The third phase is to average the distances from X to Y and from Y to X as the distance between the two sets.

Thus the modified Hausdorff distance between two sets is essentially the average of the two distances of two pairs of elements.

To calculate the similarity value between two users, first we convert user u's frequent T-patterns to LS-vector sequence patterns by performing Step 1 of the CPS-based similarity measure with semantics in Chapter 4. Then we merge equal LS-vector patterns by adding up their support values. In this way we get u's LS-vector pattern set $\overline{PS}_v^u$.

To apply this notion of the modified Hausdorff distance, we need to figure out how to calculate the distance between two LS-vector patterns, which depends on the way of calculating the distance between two LS-vectors. There are a couple of methods that can be used to measure the distance between two probability distributions in the literature from which we need to find out the appropriate ones.

Definition 19 (Bhattacharyya distance [2]). *For discrete probability distributions* $P = \langle p_1, p_2, \ldots, p_n \rangle$ *and* $Q = \langle q_1, q_2, \ldots, q_n \rangle$*, the Bhattacharyya distance between them is defined as:*

$$d_B(P, Q) = -\ln(\sum_{i=1}^{n} \sqrt{p_i \cdot q_i})$$

Definition 20 (Euclidean distance [5]). *For discrete probability distributions* $P = \langle p_1, p_2, \ldots, p_n \rangle$ *and* $Q = \langle q_1, q_2, \ldots, q_n \rangle$*, the Euclidean distance between them is defined as:*

$$d_E(P, Q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Definition 21 (Hellinger distance [9]). *For discrete probability distributions* $P = \langle p_1, p_2, \ldots, p_n \rangle$ *and* $Q = \langle q_1, q_2, \ldots, q_n \rangle$*, the Hellinger distance between them is defined as:*

$$d_{He}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2} = \sqrt{1 - \sum_{i=1}^{n} \sqrt{p_i \cdot q_i}}$$

Definition 22 (Total variation distance [11]). *For discrete probability distributions* $P = \langle p_1, p_2, \ldots, p_n \rangle$ *and* $Q = \langle q_1, q_2, \ldots, q_n \rangle$*, the total variation distance between them is defined as:*

$$d_T(P, Q) = \frac{1}{2} \sum_{i=1}^{n} \mid p_i - q_i \mid$$

Definition 23 (Relative entropy distance [10]). *For discrete probability distributions* $P = \langle p_1, p_2, \ldots, p_n \rangle$ *and* $Q = \langle q_1, q_2, \ldots, q_n \rangle$*, the relative entropy distance between them is defined as:*

$$d_R(P, Q) = \sum_{i=1}^{n} p_i \cdot \ln(\frac{p_i}{q_i})$$

The similarity value between two LS-vectors is inversely proportional to their distance. The two extreme cases are that when their distance is 0 their similarity value should be 1, and when their distance reach the maximum value their similarity value should is 0.

There are two properties that we expect the method used for calculating the distance between two probability distributions to possess. We use them as the criteria for picking out the appropriate methods.

- The method can produce values within finite bounds, since we need to convert the distance we obtain finally to a similarity value. If the maximum distance a method can generate is infinity, we cannot convert a certain distance to an appropriate similarity value.

- The second property consists of the following two observations.

    1. The similarity value $sim(v_1, v_2)$ between two LS-vectors $v_1 = \langle p_1, \ldots, p_n \rangle$ and $v_2 = \langle q_1, \ldots, q_n \rangle$ should be one if and only if their probability distributions are identical; that is, two places are considered the same if and only if they provide the same functionalities, and for each functionality the likelihoods of a user utilizing it at the two places are the same.

    $$sim(v_1, v_2) = 1 \Leftrightarrow \forall_{i=1}^{n} p_i = q_i$$

    2. The similarity value $sim(v_1, v_2)$ between two LS-vectors $v_1 = \langle p_1, \ldots, p_n \rangle$ and $v_2 = \langle q_1, \ldots, q_n \rangle$ should be zero if and only if v_1 assigns zero to every element to which v_2 assigns a positive probability, and vice versa; that is, two places are considered completely different if and only if they provide no common functionalities.

    $$sim(v_1, v_2) = 0 \Leftrightarrow \forall_{i=1}^{n} (p_i \neq 0 \Rightarrow q_i = 0 \wedge q_i \neq 0 \Rightarrow p_i = 0)$$

In the following table the five methods of calculating the distance between two probability distributions are compared in terms of the above two properties.

Distance notion	Range	Observation 1	Observation 2
Bhattacharyya	$[0, +\infty)$	Yes	Yes
Euclidean	$[0, \sqrt{2}]$	Yes	No
Hellinger	$[0, 1]$	Yes	Yes
Total variation	$[0, 1]$	Yes	Yes
Relative entropy	$[0, +\infty)$	Yes	No

Table 5.1: Comparison between methods of measuring the distance between two probability distributions

We see that Hellinger distance and Total variation distance satisfy the two properties. So we will use these two notions as the methods of calculating the distance between two LS-vectors.

We analyze these two notions in more detail. In the Hellinger distance, $\sum_{i=1}^{n} \sqrt{p_i \cdot q_i}$, called the Bhattacharyya coefficient [1], is an approximate measurement of the amount of overlap between two distributions. It can be used to determine the relative closeness of two distributions. In the Total variation distance, the distance between two distributions

is derived from the difference between each pair of corresponding probabilities of the same index.

There are two ways of calculating the similarity value between two users' LS-vector pattern sets which is the similarity value between two users, depending on whether we are able to calculate the distance between two LS-vector patterns of different lengths.

- The first way is that we divide each user's LS-vector pattern set into subsets each of which consists of all the user's patterns of some specific length, and then apply the Definition 18 to each pair of subsets whose patterns have the same length.

  To calculate the distance between each pair of subsets using the Definition 18, we need to know how to calculate the distance between two LS-vector patterns of the same length. It is not only affected by the distance between each pair of corresponding LS-vectors of the same index, but also by the support values of the two LS-vector patterns. The greater the support values are, the more frequent the behavior represented by the two patterns is and thus the more contribution the distance between the two patterns should have to the similarity between the two users. According to the previous analysis, this means that the two patterns are more likely to be one of the two pairs of patterns which are selected in the second phase of the modified Hausdorff distance calculation and the two distances of which average out at the distance between the two subsets. Thus the greater the support values are, the more likely the two patterns are to be the pair of patterns chosen by the first phase of the modified Hausdorff distance. Thus the greater the support values are, the smaller the distance between the two patterns should be. The distance between the LS-vector pattern $P_v = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_n$ and $P'_v = v'_1 \rightarrow v'_2 \rightarrow \ldots \rightarrow v'_n$ is defined as:

  $$d(P_v, P'_v) = \frac{\sum_{i=1}^{n} d_{He/T}(v_i, v'_i)}{n} \cdot (1 - support(P_v) \cdot support(P'_v)) \tag{5.1}$$

  From Equation (5.1) we see that the distance between two LS-vector patterns of the same length is between 0 and 1. Thus the range of the distance between a pair of subsets mentioned above is also [0,1].

  An problem occurs when the longest LS-vector patterns of the two users are not of the same length, which is that the user whose longest patterns are longer than those of the other user has subsets that do not have corresponding subsets in the other user's pattern set. So in order to solve this problem we introduce the notion of "virtual subset", which has the property that the distance between any LS-vector pattern set and it is the maximum possible value, which is 1 in the cases of Hellinger distance and Total variation distance. Then we have all the subsets that do not have corresponding subsets pair the "virtual subset".

  What follows is to convert the distance d between each pair of subsets to the similarity value s using the equation below.

  $$s = 1 - d \tag{5.2}$$

  Then all the similarity values between all pairs of subsets are integrated into an overall one between the two users. The longer a frequent pattern is, the more difficult it is for a user to have the pattern. Thus to compensate this difficulty, the longer the patterns in a pair of subsets are, the more weight the similarity value between the pair of subsets should carry. So we follow the practice of Equation 3.1

with respect to the weight of patterns' lengths. Assume that the longest patterns of the users u and u' are of length n, then the similarity value $sim(u, u')$ between the two users is calculated as:

$$sim(u, u') = \frac{\sum_{i=1}^{n} i^2 \cdot s_i}{\sum_{i=1}^{n} i^2} \tag{5.3}$$

where s_i is the similarity value between the pair of subsets whose patterns are of length i.

- The second way is to directly calculate the distance between two users' LS-vector pattern sets. We extend the above-mentioned method of calculating the distance between two LS-vector patterns of the same length to enable it to calculate the distance between two patterns of different lengths.

  Similarly, we introduce the notion of "virtual LS-vector", which has the property that the distance between any LS-vector and it is the maximum possible value, which is 1 in the cases of Hellinger distance and Total variation distance. When calculating the distance between two LS-vector patterns of different lengths, the LS-vectors that come from the longer pattern and do not have corresponding LS-vectors in the shorter pattern are paired with the "virtual LS-vector". In this way the two LS-vector patterns have the same length.

  Consequently we can obtain the distance between LS-vector pattern sets of users u and u' by directly applying Definition 18 to the two sets, which is in the range [0,1].

$$d_{mH}(PS_v^u, PS_v^{u'}) = \text{avg}\{\sup_{P_v \in PS_v^u} \inf_{P'_v \in PS_v^{u'}} d(P_v, P'_v), \sup_{P'_v \in PS_v^{u'}} \inf_{P_v \in PS_v^u} d(P_v, P'_v)\}$$

  Then $d_{mH}(PS_v^u, PS_v^{u'})$ is converted to the similarity value s between the two users using Equation 5.2.

Example 6. *Assume that the frequent patterns of the users u and u' and their support values are as follows, respectively.*

A	0.50
B	0.50
$A \to B$	0.20

A	0.50
C	0.50

The LS-vectors of all the RoIs are given below.

$$A \mapsto (0.0, 0.1, 0.9) \quad B \mapsto (0.8, 0.2, 0.0) \quad C \mapsto (0.1, 0.8, 0.1)$$

We use the notion of the total variation distance to calculate the distance between two LS-vectors. If we use the first way of calculating the similarity between two users mentioned previously, the distance between the two subsets whose patterns are of length one of u and u' is 0.525, and there is a distance 1 between u's subset which has a pattern of length 2 and the "virtual subset". Then the similarity values of the two pairs of pattern subsets are 0.475 and 0, respectively. According to Equation 5.3, the similarity value between u and u' is about 0.10.

If we use the second way of calculating the similarity between two users, the distance between the two users' LS-vector pattern sets is 0.525, and then the similarity value between u and u' is about 0.48.

Intuitively, the similarity value 0.48 should be more appropriate than 0.10, so for this example we can see that the second way mentioned above, which is to directly calculate the distance of the two users' LS-vector pattern sets, has a better performance. The reason is that in the first way when the longest patterns of two users have different lengths, the similarity value 0 between a user's actual subset whose patterns are of some length and the "virtual subset"will be involved in the calculation of the similarity value between the two users, and the problem that the similarity value between the two users obtained finally seems smaller than the expectation is exacerbated by considering the weight of the length factor is the square of it.

5.2 Experiments

We carried out experiments on the same two datasets using the same LS-vectors as those used in the previous chapter. In the following we give the results.

5.2.1 The experiment on the Geolife dataset

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.31	0.40	0.34	0.37	0.42	0.15	0.39	0.16	0.16
004	0.31	1.00	0.17	0.39	0.18	0.20	0.06	0.18	0.07	0.08
017	0.40	0.17	1.00	0.18	0.47	0.49	0.15	0.39	0.15	0.15
030	0.34	0.39	0.18	1.00	0.18	0.20	0.07	0.18	0.07	0.07
068	0.37	0.18	0.47	0.18	1.00	0.45	0.13	0.41	0.12	0.11
113	0.42	0.20	0.49	0.20	0.45	1.00	0.22	0.63	0.19	0.16
151	0.15	0.06	0.15	0.07	0.13	0.22	1.00	0.24	0.69	0.41
160	0.39	0.18	0.39	0.18	0.41	0.63	0.24	1.00	0.24	0.15
163	0.16	0.07	0.15	0.07	0.12	0.19	0.69	0.24	1.00	0.65
164	0.16	0.08	0.15	0.07	0.11	0.16	0.41	0.15	0.65	1.00

Table 5.2: Result using Hellinger distance and the first way

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.32	0.37	0.34	0.39	0.39	0.15	0.43	0.17	0.17
004	0.32	1.00	0.16	0.39	0.20	0.18	0.06	0.20	0.07	0.08
017	0.37	0.16	1.00	0.17	0.44	0.48	0.17	0.40	0.16	0.14
030	0.34	0.39	0.17	1.00	0.18	0.20	0.07	0.19	0.07	0.07
068	0.39	0.20	0.44	0.18	1.00	0.43	0.10	0.43	0.12	0.11
113	0.39	0.18	0.48	0.20	0.43	1.00	0.20	0.58	0.17	0.15
151	0.15	0.06	0.17	0.07	0.13	0.20	1.00	0.23	0.64	0.37
160	0.43	0.20	0.40	0.19	0.43	0.58	0.23	1.00	0.25	0.17
163	0.17	0.07	0.16	0.07	0.12	0.17	0.64	0.25	1.00	0.67
164	0.17	0.08	0.14	0.06	0.11	0.15	0.37	0.17	0.67	1.00

Table 5.3: Result using total variation distance and the first way

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.63	0.40	0.57	0.35	0.38	0.34	0.36	0.31	0.35
004	0.63	1.00	0.33	0.51	0.35	0.38	0.33	0.37	0.30	0.36
017	0.40	0.33	1.00	0.36	0.40	0.47	0.40	0.36	0.37	0.36
030	0.57	0.51	0.36	1.00	0.32	0.38	0.32	0.32	0.29	0.36
068	0.35	0.35	0.40	0.32	1.00	0.39	0.34	0.38	0.30	0.27
113	0.38	0.38	0.47	0.38	0.39	1.00	0.52	0.51	0.45	0.37
151	0.34	0.33	0.40	0.32	0.37	0.52	1.00	0.50	0.49	0.39
160	0.36	0.37	0.36	0.32	0.38	0.51	0.50	1.00	0.45	0.42
163	0.31	0.30	0.37	0.29	0.30	0.45	0.49	0.45	1.00	0.73
164	0.35	0.36	0.36	0.36	0.27	0.37	0.39	0.42	0.73	1.00

Table 5.4: Result using Hellinger distance and the second way

	003	004	017	030	068	113	151	160	163	164
003	1.00	0.66	0.37	0.55	0.35	0.35	0.36	0.39	0.35	0.37
004	0.66	1.00	0.32	0.51	0.35	0.35	0.33	0.39	0.31	0.35
017	0.37	0.32	1.00	0.34	0.37	0.45	0.41	0.37	0.39	0.36
030	0.55	0.51	0.34	1.00	0.33	0.38	0.34	0.36	0.32	0.36
068	0.35	0.35	0.37	0.33	1.00	0.39	0.32	0.38	0.30	0.26
113	0.35	0.35	0.45	0.38	0.39	1.00	0.48	0.48	0.41	0.34
151	0.36	0.33	0.41	0.34	0.32	0.48	1.00	0.46	0.43	0.35
160	0.39	0.39	0.37	0.36	0.38	0.48	0.46	1.00	0.50	0.47
163	0.35	0.31	0.39	0.32	0.30	0.41	0.43	0.50	1.00	0.76
164	0.37	0.35	0.36	0.36	0.26	0.34	0.35	0.47	0.76	1.00

Table 5.5: Result using total variation distance and the second way

Comparing Table 5.2 with Table 5.3, or Table 5.4 with Table 5.5, we see that when using the same way of calculating the similarity value between two users' LS-vector pattern sets, the similarity values obtained using Hellinger distance are quite close to those obtained using total variation distance.

Comparing Table 5.2 with Table 5.4, or Table 5.3 with Table 5.5, we see that the phenomenon revealed in Example 6 exists, which is that for each pair of users whose longest frequent patterns are of different lengths the similarity value obtained using the first way of calculating the similarity value between two users' pattern sets is much smaller than that obtained using second way. For other pairs of users the similarity values obtained using the two ways are close.

Comparing the above four tables with Table 4.3, we see that for the pairs of users whose similarity values given by the improved MTP similarity measure with semantics are 0, they have some degree of similarity using the Hausdorff distance-based measure.

Our results are more reasonable than those of the improved MTP similarity measure with semantics. We take the example of users 003 and 163 to elucidate it.

This pair of users also appears as an example in the previous chapter, and according to the analysis in the example of the previous chapter the two users have some degree of similarity. From Table 4.3, we see that the similarity value between them given by the improved MTP similarity measure is 0, which is obviously inappropriate.

5.2.2 The experiment on the Yonsei dataset

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.56	0.48	0.45	0.55	0.16	0.23	0.19	0.40	0.46
002	0.56	1.00	0.50	0.47	0.37	0.14	0.21	0.20	0.46	0.45
003	0.48	0.50	1.00	0.66	0.53	0.15	0.18	0.19	0.48	0.49
004	0.45	0.47	0.66	1.00	0.50	0.15	0.18	0.22	0.51	0.51
007	0.55	0.37	0.53	0.50	1.00	0.26	0.22	0.21	0.56	0.62
009	0.16	0.14	0.15	0.15	0.26	1.00	0.09	0.09	0.18	0.18
081	0.23	0.21	0.18	0.19	0.22	0.09	1.00	0.50	0.19	0.18
082	0.19	0.20	0.19	0.22	0.21	0.09	0.50	1.00	0.21	0.20
121	0.40	0.46	0.48	0.51	0.56	0.18	0.19	0.21	1.00	0.82
122	0.46	0.45	0.49	0.51	0.62	0.18	0.18	0.20	0.82	1.00

Table 5.6: Result using Hellinger distance and the first way

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.54	0.46	0.46	0.54	0.16	0.22	0.18	0.42	0.45
002	0.54	1.00	0.49	0.46	0.36	0.14	0.21	0.20	0.48	0.46
003	0.46	0.49	1.00	0.65	0.52	0.15	0.17	0.18	0.45	0.45
004	0.46	0.46	0.65	1.00	0.53	0.15	0.18	0.22	0.52	0.52
007	0.54	0.36	0.52	0.53	1.00	0.26	0.22	0.23	0.56	0.62
009	0.16	0.14	0.15	0.15	0.26	1.00	0.09	0.09	0.17	0.18
081	0.22	0.21	0.17	0.18	0.22	0.09	1.00	0.50	0.20	0.18
082	0.18	0.20	0.18	0.22	0.23	0.09	0.50	1.00	0.23	0.23
121	0.42	0.48	0.45	0.52	0.56	0.17	0.20	0.23	1.00	0.83
122	0.45	0.46	0.45	0.52	0.62	0.18	0.18	0.23	0.83	1.00

Table 5.7: Result using total variation distance and the first way

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.56	0.42	0.41	0.42	0.41	0.40	0.39	0.38	0.39
002	0.56	1.00	0.35	0.37	0.30	0.36	0.37	0.38	0.38	0.38
003	0.42	0.35	1.00	0.46	0.51	0.39	0.31	0.34	0.42	0.42
004	0.41	0.37	0.46	1.00	0.47	0.39	0.35	0.40	0.46	0.46
007	0.42	0.30	0.51	0.47	1.00	0.57	0.41	0.41	0.53	0.54
009	0.41	0.36	0.39	0.39	0.57	1.00	0.47	0.48	0.46	0.47
081	0.40	0.37	0.31	0.35	0.41	0.47	1.00	0.50	0.34	0.34
082	0.39	0.38	0.34	0.40	0.41	0.48	0.50	1.00	0.38	0.39
121	0.38	0.38	0.42	0.46	0.53	0.46	0.34	0.38	1.00	0.63
122	0.39	0.38	0.42	0.46	0.54	0.47	0.34	0.39	0.63	1.00

Table 5.8: Result using Hellinger distance and the second way

	001	002	003	004	007	009	081	082	121	122
001	1.00	0.54	0.40	0.40	0.42	0.42	0.40	0.36	0.43	0.44
002	0.54	1.00	0.34	0.35	0.29	0.36	0.38	0.38	0.41	0.41
003	0.40	0.34	1.00	0.48	0.52	0.39	0.29	0.31	0.42	0.42
004	0.40	0.35	0.48	1.00	0.47	0.39	0.33	0.40	0.46	0.46
007	0.42	0.29	0.52	0.47	1.00	0.56	0.40	0.45	0.55	0.55
009	0.42	0.36	0.39	0.39	0.56	1.00	0.49	0.47	0.46	0.47
081	0.40	0.38	0.29	0.33	0.40	0.49	1.00	0.51	0.38	0.37
082	0.36	0.38	0.31	0.40	0.45	0.48	0.51	1.00	0.40	0.41
121	0.43	0.41	0.42	0.46	0.55	0.46	0.38	0.40	1.00	0.65
122	0.44	0.41	0.42	0.46	0.55	0.47	0.37	0.41	0.65	1.00

Table 5.9: Result using total variation distance and the second way

Comparing Table 5.6 with Table 5.7, or Table 5.8 with Table 5.9, we see that when using the same way of calculating the similarity value between two users' LS-vector pattern sets, the similarity values obtained using Hellinger distance are quite close to those obtained using total variation distance.

Comparing Table 5.6 with Table 5.8, or Table 5.7 with Table 5.9, we see that the phenomenon revealed in Example 6 exists, which is that for each pair of users whose longest frequent patterns are of different lengths the similarity value obtained using the first way of calculating the similarity value between two users' pattern sets is much smaller than that obtained using second way. For other pairs of users the similarity values obtained using the two ways are close.

Comparing the above four tables with Table 4.4 / 4.5, we see that for the great majority of pairs of users, the similarity values given by the Hausdorff distance-based measure are greater than those given by the CPS-based measure with semantics. Comparing the four tables with Table 4.6, we see that there are some pairs of users whose similarity values given by the Hausdorff distance-based measure are greater and smaller than those given by the improved MTP measure with semantics, respectively.

We take the example of users 121 and 122 to show that the results of the Hausdorff distance-based measure are more reasonable than those of the improved MTP measure with semantics. Using the first way of removing infrequent behavior in the CPS-based measure with semantics, user 121 has 182 frequent semantic patterns and user 122 has 234 patterns. They have 182 common patterns. Using the second way of removing infrequent behavior in the CPS-based measure with semantics, user 121 has 192 frequent semantic patterns and user 122 has 207 patterns. They have 172 common patterns. So we see that the average percentage of common patterns accounting for all of their patterns cannot be as high as 0.94 given by the improved MTP measure with semantics. After considering the similarity between the frequencies of the common patterns, the similarity value between the two users will become smaller than the average percentage. So their similarity value cannot be as high as 0.94 given by the improved MTP measure with semantics.

Chapter 6

The MinUS Tool

We developed a tool, named MinUS (Mine User Similarity), that can abstract users' profiles from the source data of their geographical trajectories and then use them to compare user similarity by virtue of different similarity measures. It provides four major functions, which are managing datasets, constructing user mobility profiles, viewing files and visualization on maps, and measuring user similarity. Below is the general workflow of the tool.

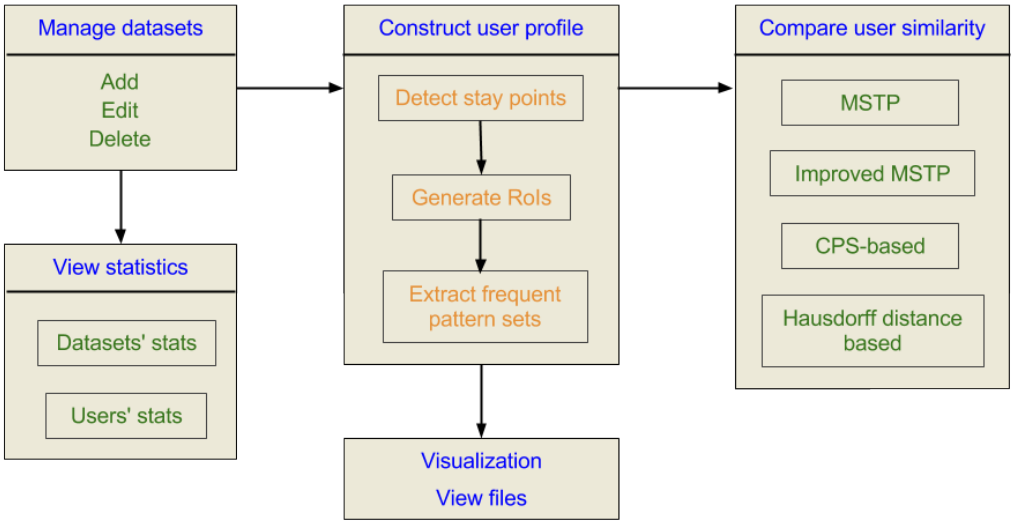


Figure 6.1: The general workflow

6.1 Managing datasets

This module consists of two parts, which are basic operations of datasets and viewing statistic information.

6.1.1 Basic operations of datasets

We allow users to add a new dataset, edit and delete an existing dataset.

The tool is able to handle two types of datasets according to the different types of source data. Source data of the new dataset a user provides should be of one of the two types, GPS points and stay points. When adding a new dataset of some type, the tool

will check if the source data of the dataset is in accord with the format requirements of that type. If the dataset is added successfully, the tool will continue gathering users' statistics of GPS trajectories or stay point trajectories for the two types of datasets, respectively.

The name and output directory of an existing dataset can be changed. Users can choose whether to delete contents on disk as well while deleting a dataset.

6.1.2 Viewing statistics

Users can view a dataset's general information, which includes its name, type, input directory, output directory, and the number of users it contains.

The tool also permits viewing statistics of users included in a dataset, which includes statistics of GPS trajectories and possible statistics of stay point trajectories if already detected for a dataset of GPS point type, or only statistics of stay point trajectories for a dataset of stay point type. GPS points (stay points) of a user in a day form a GPS (stay point) trajectory. Statistics of GPS (stay point) trajectories of a user contain the number of days in which the user has GPS (stay point) trajectories, the total number of GPS points (stay points), the minimum number of GPS points (stay points) in a GPS (stay point) trajectory, the maximum number of GPS points (stay points) in a GPS (stay point) trajectory, and the average number of GPS points (stay points) in a GPS (stay point) trajectory. The user list in the user interface allows multiple selection, so statistics of any number of users can be viewed at a time.

Taking advantage of these statistics, the user of the MinUS tool can choose appropriate users in a dataset to compare their similarity.

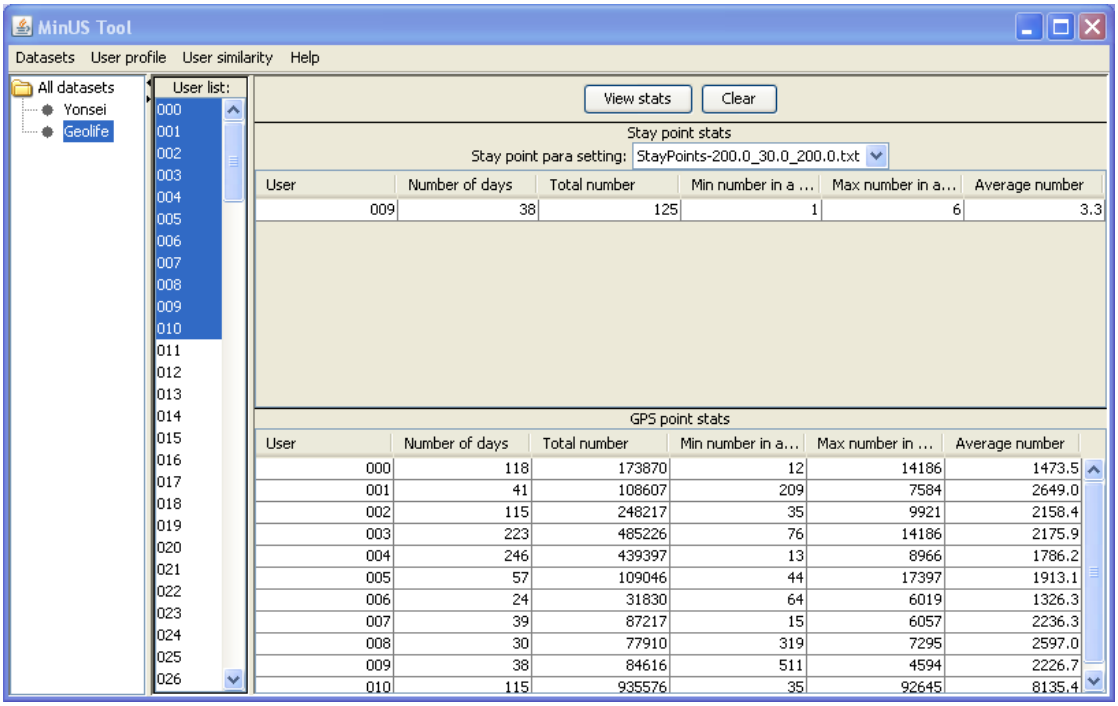


Figure 6.2: User interface of viewing users' statistics

6.2 Constructing user mobility profiles

For datasets of GPS point type the mobility profile construction process of Chen et al. is made up of three sequential steps, which are detecting stay points, generating RoIs and extracting frequent pattern sets, while for datasets of stay point type it only consists of the latter two steps. The intermediate results can be visualized on maps and viewed by opening files.

6.2.1 The construction process

Three parameters need setting before detecting stay points, which are the time interval threshold θ_t, the distance threshold θ_d and the threshold θ_m that are mentioned in Chapter 2. A user will generate different files of stay point trajectories using different parameter settings.

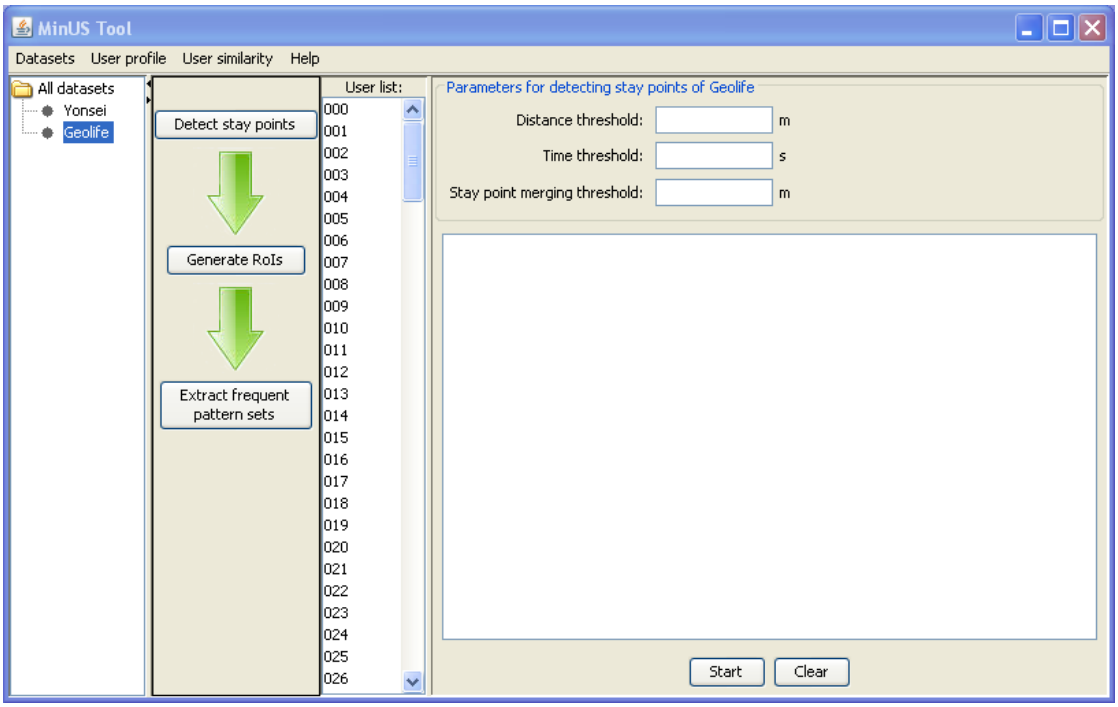


Figure 6.3: User interface of detecting stay points

After obtaining users' stay points, RoIs can be generated from multiple users' stay points. Three parameters need setting, which are the deletion percentage that refers to how many stay points at most will constitute outliers and then be deleted before proceeding to the clustering process, and the upper and lower bounds of a parameter K that will be used in the LOF algorithm of removing the outliers. It is meaningful to generate RoIs from multiple users' stay points only when all the stay points are detected using the same parameter setting.

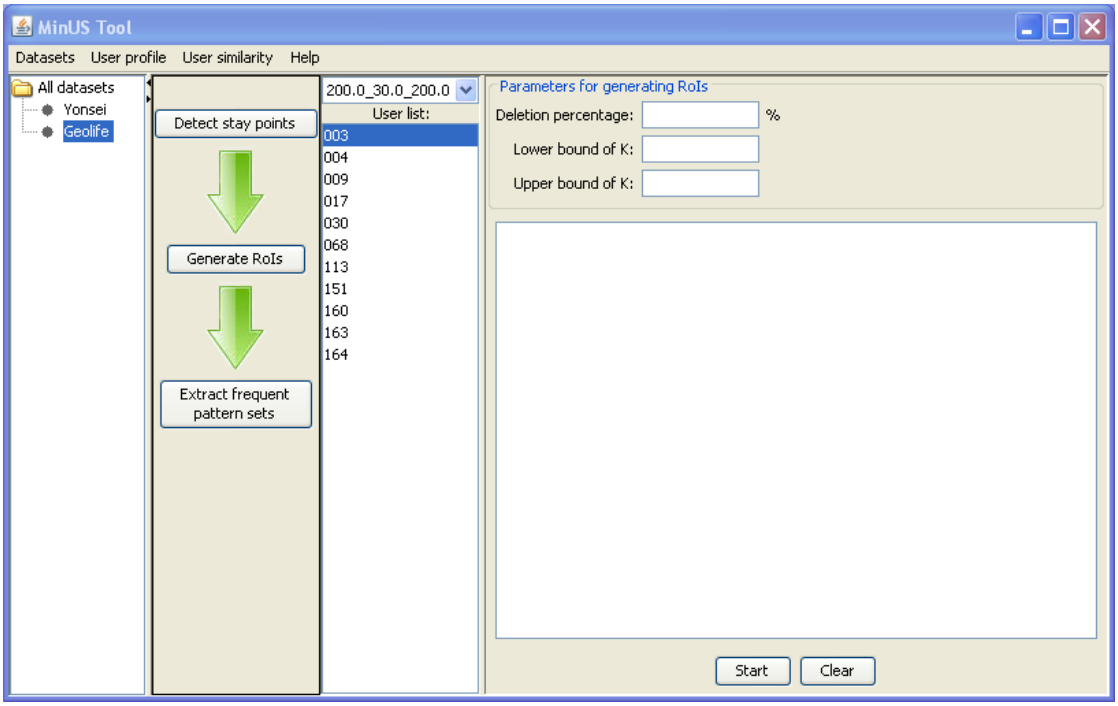


Figure 6.4: User interface of generating RoIs

After obtaining the RoI file, frequent pattern sets will be able to be extracted from stay point trajectories. Only the frequent patterns of those users who are involved in generating a RoI file can be extracted based on the RoI file from the users' files of stay point trajectories used to generate the RoI file. Parameters which need setting include the support value threshold and the time tolerance introduced in Chapter 2 for a T-pattern to become frequent, the side length of a cell which will be used in the T-pattern mining algorithm, and a usable RoI file. The tool uses the trajectory pattern mining tool initially created in [18] to carry out this task, and we improved one defect, which is that it cannot produce results when the time tolerance threshold τ is set below 18000 seconds. We also enhance its functionality by not only permitting extracting frequent patterns from all the stay point trajectories of a user, but also permitting extracting frequent patterns from his stay point trajectories in some particular type of days, like on weekdays or weekends.

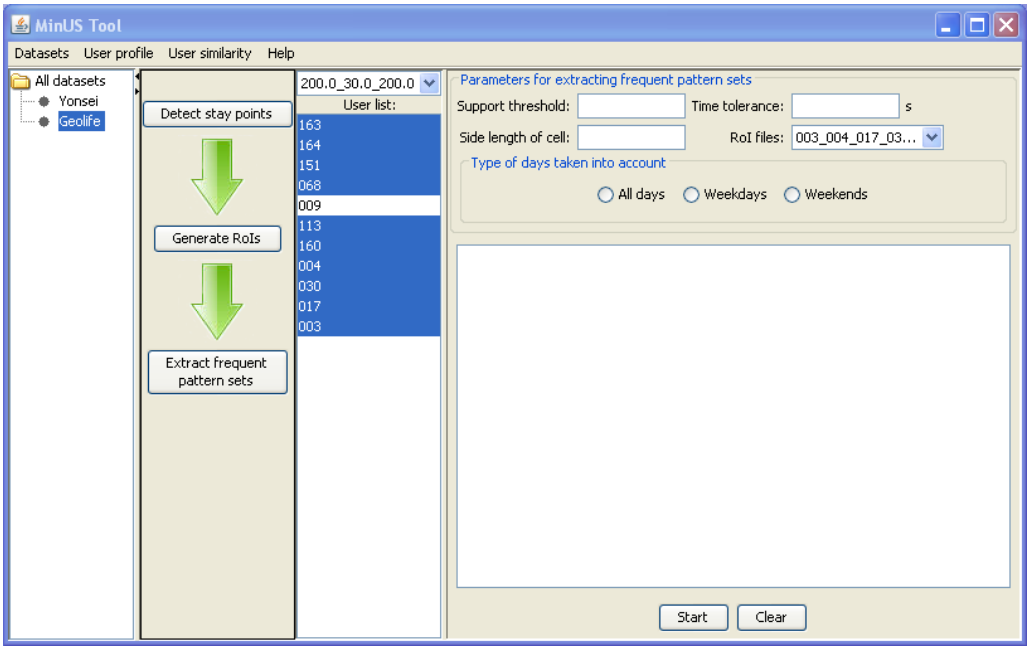


Figure 6.5: User interface of extracting frequent pattern sets

6.2.2 Visualization and viewing files

In the midst of constructing a user's profile, we can visualize the intermediate results, like his GPS and stay point trajectories, and RoI files which he is involved in forming, by displaying them on maps. Any combination of the three types of intermediate results can be chose at a time in the visualization panel. By clicking context menus an intermediate result can be visualized individually as well. A user's files that store his intermediate results can also be opened and viewed via right-click menus of the list components in the visualization panel and those panels of constructing user profiles.

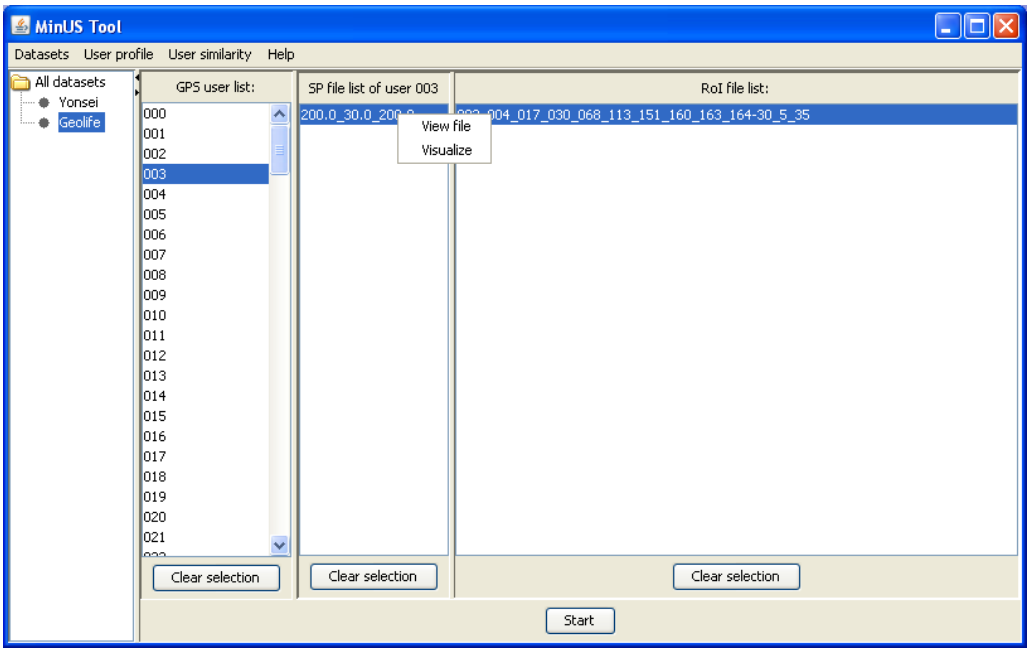


Figure 6.6: User interface of viewing files and visualization

6.3 Measuring user similarity

6.3.1 Managing semantic files

Since semantics is taken into account in the form of LS-vectors, we should know the LS-vector associated with each RoI in the file if we want to compare user similarity semantically based on an RoI file. A file which stores all RoI IDs in an RoI file and their associated LS-vectors is called a semantic file.

The tool provides two ways of adding a semantic file, which are to have the tool automatically generate one whose LS-vectors are normally distributed based on an RoI file, or to add an existing one manually. Each semantic file is only attached to one RoI file, so a semantic file can only be used when comparing users' pattern sets that are produced based on the RoI file to which the semantic file is attached.

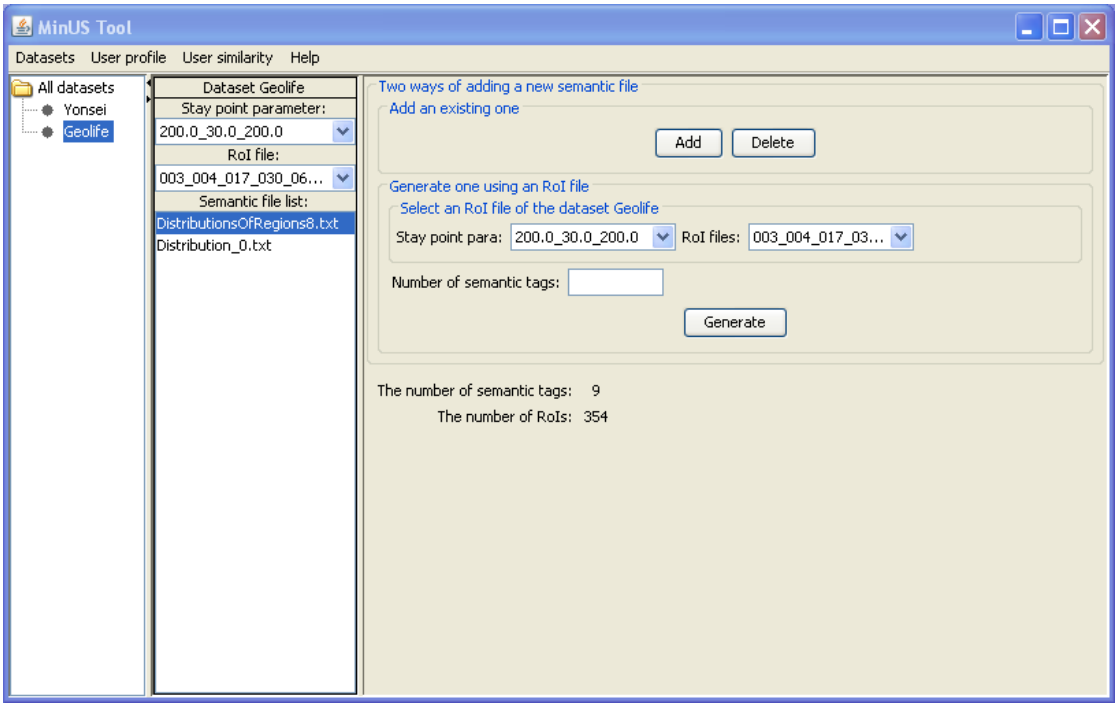


Figure 6.7: User interface of managing semantic files

6.3.2 Comparing users

The tool provides four user similarity measures, which are the MTP similarity measure (with time), the improved MTP similarity measure (with sematics) (with time), the CPS-based similarity measure (with sematics) and the Hausdorff distance-based similarity measure (with sematics). To list the users' pattern sets which can be compared with each other, we need to specify three parameters sequentially for datasets of GPS point type, which are the parameter setting used when detecting stay points, the RoI file based on and the parameter setting when extracting these pattern sets, by selecting items from the three combo boxes from top to bottom. For datasets of stay point type, only the latter two parameters need setting. An arbitrary number of users can be chosen to be compared at a time. The result will be displayed in a table in which the gray value of each cell' background color is linearly proportional to the similarity value contained in the cell.

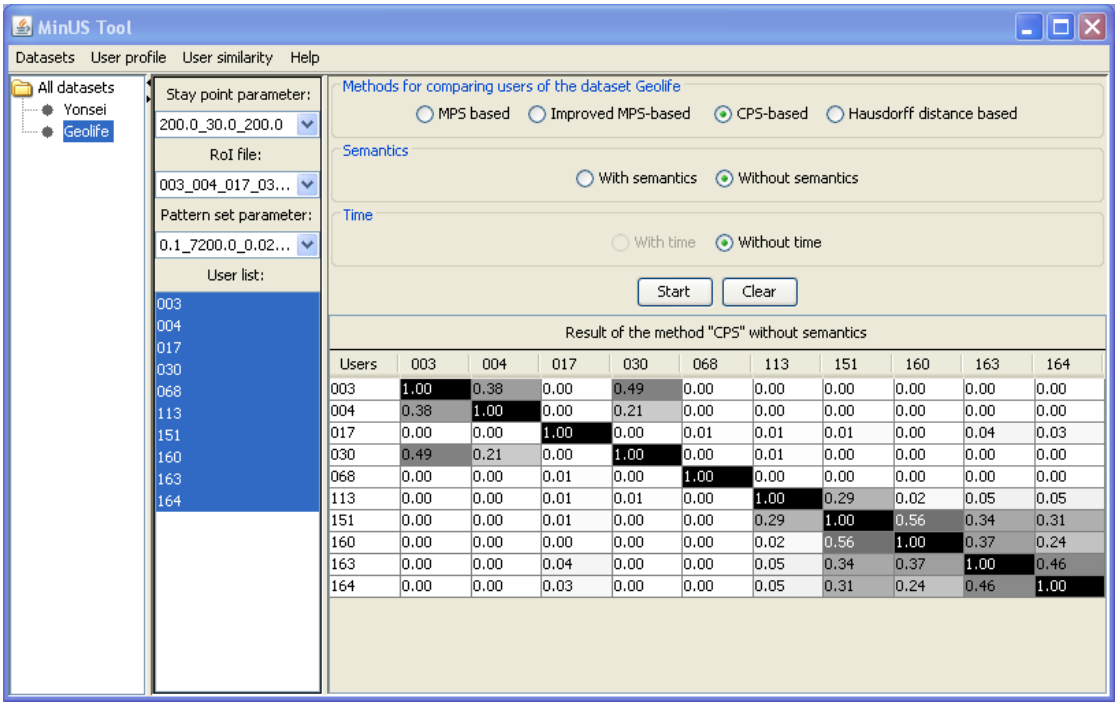


Figure 6.8: User interface of comparing users

Chapter 7

Conclusion

In this thesis, we focused on comparing user similarity based on mobility profiles in the form of sequence pattern sets. First, we found a couple of defects of the (improved) MTP similarity measure in the literature. These defects are all related to the measure's main idea which is to use the comparison between frequent patterns as a building block for the comparison between users. Then we gave four rudimentary principles that similarity measures should follow, and affirmed that the (improved) MTP similarity measure does not completely satisfy the principles. Next we proposed the CPS-based similarity measure that follows all of those principles. And we extended it further to enable it to take semantics into consideration. In addition, we proposed another Hausdorff distance-based similarity measure with semantics which directly calculate the similarity value between users' pattern sets. The experiments conducted on two real datasets demonstrate that our newly proposed similarity measures are able to efficiently produce more reasonable results than the improved MTP similarity measure (with semantics). Finally, we developed the MinUS software tool which implements the mobility profile construction process of Chen et al. and all of the involved similarity measures in this thesis.

Next we conclude that which similarity measures apply in some specific situation.

- If we would like to search for similar users who live in the same geographical region, like a city, we should use the CPS-based similarity measure.
- If we would like to search for similar users no matter whether their living places are close, e.g., they can live in different cities, we should use the CPS-based similarity measure with semantics or the Hausdorff distance-based similarity measure.

With our new similarity measures and the MinUS tool, location-based social networks are able to measure the similarity among users more accurately and thus provide a more effective user recommendation service.

There are a few limitations on comparing user similarity using frequent pattern sets. First, the user mobility profile construction process might be a bit time-consuming. Second, in this thesis we used simulated LS-vectors which are randomly generated. To obtain precise similarity values after considering semantics in reality, we need to obtain real and precise LS-vectors associated with the involved RoIs; in other words, we need to precisely gather real information about the probabilities of users utilizing the functionalities of the involved places, which might be a laborious work.

For future work, we can apply our similarity measures to location privacy analysis. A high similarity value between a given set of anonymous trajectories and a user's mobility profile indicates a high likelihood for the user to be the owner of the trajectories.

Bibliography

[1] Bhattacharyya coefficient. http://en.wikipedia.org/wiki/Bhattacharyya_distance. Wikipedia.

[2] Bhattacharyya distance. http://en.wikipedia.org/wiki/Bhattacharyya_distance. Wikipedia.

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

[4] Bray-curtis similarity. http://www.code10.info/index.php?option=com_content&view=article&id=46:articlebray-curtis-dissim&catid=38:cat_coding_algorithms_data-similarity&Itemid=57.

[5] Euclidean distance. http://en.wikipedia.org/wiki/Euclidean_distance. Wikipedia.

[6] Foursquare. https://foursquare.com.

[7] Geolife GPS trajectories. http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/. Microsoft Research.

[8] Hausdorff distance. http://en.wikipedia.org/wiki/Hausdorff_distance. Wikipedia.

[9] Hellinger distance. http://en.wikipedia.org/wiki/Hellinger_distance. Wikipedia.

[10] Relative entropy distance. http://en.wikipedia.org/wiki/KullbackLeibler_divergence. Wikipedia.

[11] Total variation distance of probability measures. http://en.wikipedia.org/wiki/Total_variation_distance_of_probability_measures. Wikipedia.

[12] Yonsei dataset. http://crawdad.cs.dartmouth.edu/meta.php?name=yonsei/lifemap.

[13] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proc. 11th International Conference on Data Engineering (ICDE)*, pages 3–14. IEEE CS, 1995.

[14] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. In *Proc. 28th Annual ACM Symposium on Applied Computing (SAC)*, pages 261–266. ACM, 2013.

[15] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. *ACM Transactions on the Web (TWEB)*, 2013. Under review.

[16] Fosca Giannotti, Mirco Nanni, and Dino Pedreschi. Efficient mining of temporally annotated sequences. In *Proc. 6th International Conference on Data Mining (SDM)*, pages 346–357. SIAM, 2006.

[17] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, and Fabio Pinelli. Mining sequences with temporal annotations. In *Proc. ACM Symposium on Applied Computing (SAC)*, pages 593–597. ACM Press, 2006.

[18] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proc. 13th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 330–339. ACM Press, 2007.

[19] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In *Proc. 17th International Conference on Data Engineering (ICDE)*, pages 215–224. IEEE CS, 2001.

[20] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proc. 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, pages 442–445. ACM, 2010.

[21] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. On the semantic annotation of places in location-based social networks. In *Proc. 17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 520–528. ACM, 2011.

[22] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Mining user similarity from semantic trajectories. In *Proc. International Workshop on Location Based Social Networks (GIS-LBSN)*, pages 19–26. ACM, 2010.

[23] Mohammed Javeed Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, pages 31–60, 2001.

[24] Yu Zheng, Longhao Wang, Ruochi Zhang, Xing Xie, and Wei-Ying Ma. Geolife: Managing and understanding your past life over maps. In *Proc. 9th International Conference on Mobile Data Management (MDM)*, pages 211–212. IEEE CS, 2008.

[25] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, pages 1–44, 2011.