Mining User Similarity from Semantic Trajectories

Josh Jia-Ching Ying
Institute of Computer Science and Information Engineering
National Cheng Kung University
No.1, University Road, Tainan City
701, Taiwan (R.O.C.)
jashying@gmail.com

Eric Hsueh-Chan Lu
Institute of Computer Science and Information Engineering
National Cheng Kung University
No.1, University Road, Tainan City
701, Taiwan (R.O.C.)
eric@idb.csie.ncku.edu.tw

Wang-Chien Lee
Dept. of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802, USA
wlee@cse.psu.edu

Tz-Chiao Weng
Institute of Computer Science and Information Engineering
National Cheng Kung University
No.1, University Road, Tainan City
701, Taiwan (R.O.C.)
airizumo@gmail.com

Vincent S. Tseng
Institute of Computer Science and Information Engineering
National Cheng Kung University
No.1, University Road, Tainan City
701, Taiwan (R.O.C.)
tsengsm@mail.ncku.edu.tw

ABSTRACT

In recent years, research on measuring trajectory similarity has attracted a lot of attentions. Most of similarities are defined based on the geographic features of mobile users' trajectories. However, trajectories geographically close may not necessarily be similar because the activities implied by nearby landmarks they pass through may be different. In this paper, we argue that a better similarity measurement should have taken into account the semantics of trajectories. In this paper, we propose a novel approach for recommending potential friends based on users' semantic trajectories for location-based social networks. The core of our proposal is a novel trajectory similarity measurement, namely, *Maximal Semantic Trajectory Pattern Similarity* (*MSTP-Similarity*), which measures the semantic similarity between trajectories. Accordingly, we propose a user similarity measurement based on *MSTP-Similarity* of user trajectories and use it as the basis for recommending potential friends to a user. Through experimental evaluation, the proposed friend recommendation approach is shown to deliver excellent performance.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications – *Data Mining, Spatial Databases and GIS*

General Terms

Performance, Design, Experimentation.

Keywords

Global Positioning System, Trajectory Database, Semantic Similarity; Data mining.

1. INTRODUCTION

With the rapid growth and fierce competition in the market of social networking services, many service providers have deployed various recommendation services, such as friend recommender, to introduce users to know each other in order to grow the underlying social networks. For example, several well known social networking systems, such as Facebook, Twitter, and FriendFeed, all have provided various friend search and recommendation services. These services are very useful for users to find people who have similar interests, learn and share information/experiences with others, and make friends. However, based on our observation, most of the friend recommendation engines (called *friend recommenders)* use profiles or on-line behavior of users to make recommendations (e.g., some systems often recommend friends' friends to their users) instead of capturing the ''real'' characteristics in user behavior.

In recent years, a new breed of social networking services, called *location-based social networks (LBSNs)*, have emerged. Thanks to the advances in mobile computing and wireless networking technologies, users of LBSNs can track and share location-related information with each other on the move. By adding this new dimension of location features, LBSNs bring its users from the virtual world back into their real lives and allow the real-life experiences be shared in the virtual world in a more convenient fashion. Among the LBSNs, many sites allows not only visited locations but also trajectories of users to be shared, e.g., Bikely [1]. Basically, a trajectory is geographic data which captures a user's physical moving behavior in real world. It typically consists of a sequence of spatio-temporal points (in form of latitude, longitude, and time) as shown in Figure 1. Thus, with the logs of user trajectories, the physical behaviors of users can be extracted from user trajectories.

With the development of Web 2.0 technology, many mobile users are willing to share their trajectories with others [12]. A number of forums have been established to facilitate sharing of trajectories among their users [1][2]. We envisage that such logs of user trajectories will also be available and sharable in many

LBSNs. When a user uploads his trajectory log, these LBSNs may invoke their friend recommenders to recommend her some users with similar moving behavior. Moreover, the system may recommend the newly uploaded trajectory not only to her friends but also other similar users.

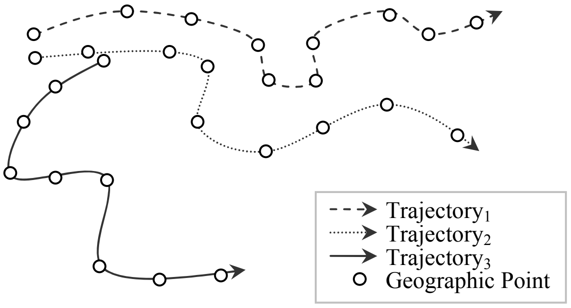


Figure 1. An example of geographic trajectory.

Obviously user similarity plays a crucial role in these recommendation services. Although the issue of measuring mobile users' similarity in terms of their trajectories has been discussed in the literature, existing studies mostly focus only on analyzing *geographic* features of user trajectories [7][8][12]. As mentioned earlier, a geographic trajectory typically consists of a sequence of geographic points (represented as <latitude, longitude>), tagged with timestamps. As a result, the measurement of user similarity based on geographic trajectory similarity is constrained by the geographic properties of the trajectory data. For example, two close trajectories are considered as more similar to each other than another trajectory that is far away. As Figure 1 shows, among the three trajectories of users, the geographic distance between *Trajectory$_1$* and *Trajectory$_2$* is closer than that between *Trajectory$_1$* and *Trajectory$_3$*. Thus, *Trajectory$_1$* is more similar to *Trajectory$_2$* than to *Trajectory$_3$*. We argue that merely using the geographic information to capture the trajectory similarity as well as user similarity is not sufficient.

The notion of *semantic trajectory* has been proposed by Alvares *et al.* [3][4]. Basically, a semantic trajectory consists of a sequence of locations with semantic tags to capture the landmarks passed by. Consider Figure 2 where trajectories are tagged with a number of semantic labels such as School, Park, etc. We observe that both *Trajectory$_1$* and *Trajectory$_3$* can be represented as the sequence <School, Park, Restaurant>. The semantic behaviors of *Trajectory$_1$* and *Trajectory$_3$* are quite the same and thus they are more similar to each other than to *Trajectory$_2$*. In this paper, we propose to consider semantic trajectories to measure the similarity.

To support friend recommendation based on the semantic trajectories of mobile users, we propose a novel similarity measurement, namely, *Semantic Trajectory Pattern Similarity* (*MSTP-Similarity*), to evaluate the similarity between two trajectories. Accordingly, we propose a novel similarity measurement to evaluate the user similarity based on the *MSTP-Similarity*. As such, potential friends with similar semantic trajectory patterns, even if they live in different cities, may be connected. To our best knowledge, this is the first work on mining mobile user similarity by considering both semantic tags and sequential relations from trajectories. Through an experimental evaluation by simulation, we show that the proposed friend recommendation approach delivers excellent performance.

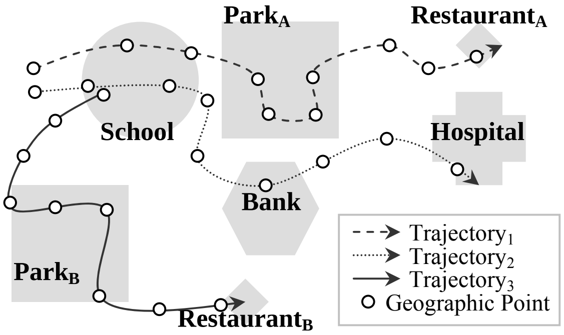


Figure 2. An example of semantic trajectory.

The remaining of this paper is organized as follows. We briefly review the related work in Section 2 and describe the proposed *MSTP-Similarity*, user similarity measurement based on *MSTP-Similarity,* and our friend recommendation approach. The empirical evaluation for performance study is made in Section 4. The conclusions and future work are given in Section 5.

2. RELATED WORK

Many studies had discussed the similarity measurement problems in data mining. Trajectory similarity measurement [6] and user similarity measurement [7][8][12] are two hot topics in this problem domain. In [6], Lee et al propose a Partition-and-Group method to calculate the similarity between two trajectories. They first find the characteristic points of each trajectory in a line segmentation process and then apply three kinds of distance measures, i.e., perpendicular distance, parallel distance, and angle distance, on these segments to group the trajectories. However, these distance measures are only applicable to geographic information and thus can not be used to measure user similarity based on semantic trajectory.

The main idea of trajectory based user similarity measurement is to derive the user similarity by analyzing the movement behavior of mobile users. In [12], Zheng *et al* propose a personalized friend and location recommendation system. To explore users' similarity, the system considers users' movement behaviors in various location granularities. Based on the notion of stay points which are the geographic regions mobile users stay for over a time threshold, the system discovers all of the stay points in trajectories and then employ a density-based clustering algorithm to organize these stay points as a hierarchical framework. Such cluster is named stay region (or stay location). As such, a personal hierarchical graph is formed for each user. For each level of the hierarchical graph, a user's trajectory can be transformed as a stay region sequence. To measure two users' similarity, some common sequences, named similar sequence, are discovered by matching their stay region sequences in each level of the hierarchical graph. Then, for each stay region, the TFIDF value for a similar sequence is calculated, where, TF value represents the minimum frequency of the two users accessed this stay region within the similar sequence, while the IDF value indicates the number of users who visited this stay region. Finally, the similarity between two users is derived by the summation of the TFIDF values of all stay regions within the similar sequences. However, this approach treats every stay region in the similar sequence independently, i.e., without considering the sequential property of stay regions in the similar sequence. In [8], an LBS-

Alignment method was proposed to calculate the similarity of two mobile users. The LBS-Alignment method calculates two users' similarity by using the longest common sequence within their Mobile Sequential Patterns to measure the similarity. By analyzing such longest common sequences, the ratio of common part in the Mobile Sequential Patterns are taken as the similarity. Although all these approaches have considered temporal information and location hierarchy, they do not take into account the semantic of locations.

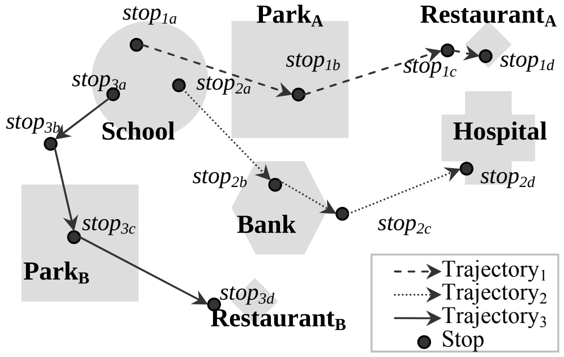


Figure 3. An example of semantic trajectory.

In recent years, a number of studies on Semantic Trajectory Data Mining have appeared in the literature [3][4]. In [3], Alvares et al propose to explore the geographic semantic information to mine Semantic Trajectory Pattern from mobile users' location histories. First, they discover the *stops* (similar to the stay points in [12]) of each trajectory and map these stops to semantic landmarks. Then, they apply a sequential pattern mining algorithm on this sequence dataset to obtain frequent pattern, namely, semantic trajectory pattern, to represent the frequent semantic behaviors of mobile user. In [4], Bogorny et al consider hierarchical geographic semantic information in order to discover more interesting patterns. Since the notion of stops in the above works only takes the viewpoint of 'stay' but not considering the positions of these stops in geographic space, many unknown stops are generated. For example, as shown in Figure 3, $stop_{1c}$, $stop_{2c}$, and $stop_{3b}$ are not associated with any semantic landmark. Hence, $Trajectory_1$ is transformed as the sequence <School, Park, Unknown, Restaurant>. From the figure, it is clear that $stop_{1c}$ is near the Restaurant_A. Thus, by taking into account the geometric distribution of these stops, $stop_{1c}$ and $stop_{1d}$ are grouped together such that the $Trajectory_1$ is transformed as the sequence <School, Park, Restaurant>. To the best of our knowledge, there is no existing work that incorporates semantics in the problems of trajectory similarity and user similarity measurement.

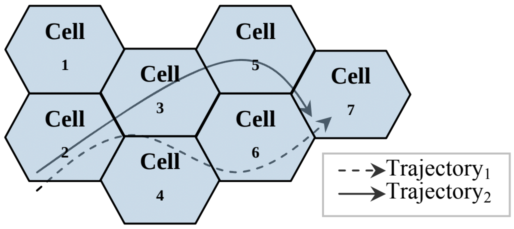


Figure 4. An example of semantic trajectories.

In addition to the GSP trajectory, Eagle *et al.* use another kind of mobile phone data to infer a social network. Such dataset contains users' movement behavior, we call it cell trajectory. The trajectory consists of a sequence of spatio-temporal points in form of cell station ID, arrive time, and leave time as shown in Figure 4. The difference between GPS trajectory and cell trajectory is that the geographic points in a cell trajectory are presented as a cell station ID. In other words, for the cell trajectory, the position of mobile user can not be obtained precisely since the signal coverage of a cell station may be very large. In previous studies [3][4], Alvares *et al* have introduced how to transform a GPS trajectory to a semantic trajectory. However, there is no research discuss about how to transform a cell trajectory to a semantic trajectory.

3. Semantic Trajectory Based Friend Recommendation

Based on the notion of semantic trajectory, in this section, we propose a novel framework, namely, *SemanTraj*, for friend recommendation. Different from conventional friend recommendations based on geographic features of trajectories, we stress on the semantic information in trajectories for recommending to users potential friends who may have completely different geographic behaviors, e.g., living in other cities, but they have similar semantic behaviors. The *SemanTraj* framework consists of four phases: 1) semantic trajectory transformation, 2) maximal semantic trajectory pattern mining, 3) semantic similarity measurement, and 4) potential friend recommendation. Figure 5 shows the framework and flow of data processing within *SemanTraj*.

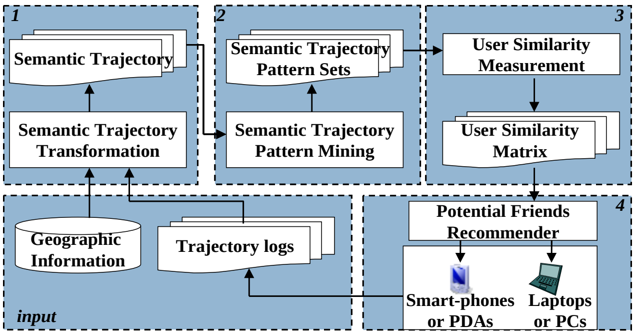

1
Semantic Trajectory
Semantic Trajectory Transformation
2
Semantic Trajectory Pattern Sets
Semantic Trajectory Pattern Mining
3
User Similarity Measurement
User Similarity Matrix
4
Potential Friends Recommender
Smart-phones or PDAs
Laptops or PCs
Geographic Information
Trajectory logs
input

Figure 5. A Framework for Potential Friends Recommendation

3.1 Semantic trajectory transformation

As mentioned earlier, geographic trajectories may appear in form of 1) GPS trajectory and 2) cell trajectory. In this phase, we basically follow Alvares *et al*'s approach [3][4] to transform GPS trajectories into semantic trajectories. To deal with cell trajectories, we treat a cell station as a geographic region. Then, the stay time can be derived by calculating the difference between the time a user arrives and leaves the cell. A user-specified time threshold is used to filter the cells with stay time shorter than the threshold. We call the remaining cells (i.e., their stay time is equal or greater than the threshold) *stay cell*. Therefore, we can transform each cell trajectory as a *stay cell sequence*. Then, we use a geographic information database to assign semantic terms to the discovered stay cells. The geographic information database is a customized spatial database which stores the semantic information of landmarks collected from Google Map. (Alternatively, a gazetteer can be used to as a general-purpose geographic information database for this operation.) In our geographic information database, we store landmarks, their geographic scopes, and the associated semantic term(s). In this paper, we use some general categories of the landmarks as their semantic terms. If a stay cell overlaps one or several landmarks stored in the geographic information database, the semantics of these landmarks would be assigned to this stay cell. Take Figure 6 as an example. The semantic term of the landmark **Park$_A$** is "Park". Since *Stay Cell$_2$* overlaps the landmark **Park$_A$**, the semantic term "Park" is assigned to *Stay Cell$_1$*. Similarly, we will assign the semantic term "School" to *Stay Cell$_1$*. It is possible that a Stay Cell overlaps none of landmark. For example, in Figure 6, there is no landmark overlapped with *Stay Cell$_0$*. If a Stay Cell overlaps no landmark, we assign the semantic term "Unknown" to the Stay Cell. After assigning semantic terms to the Stay Cell, a stay cell sequence can be transformed as a sequence of landmarks which is called the *semantic trajectory*. For example, the stay cell sequence <*Stay Cell$_0$, Stay Cell$_1$, Stay Cell$_2$, Stay Cell$_3$*> is transformed as <{Unknown}, {School, Park }, {Park}, {Hospital}>.

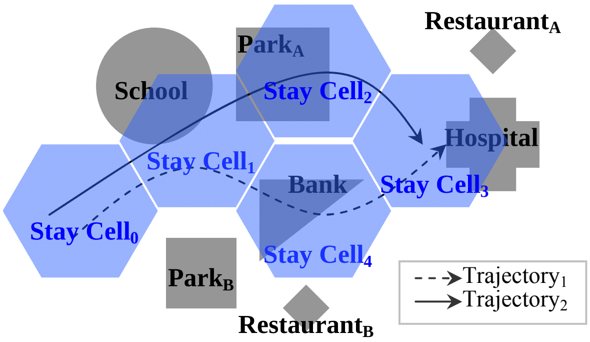


Figure 6. An example of semantic trajectories.

3.2 Maximal Semantic Trajectory Pattern Mining

After transforming each geographic trajectory to a semantic trajectory, each user's geographic trajectory set is transformed as a semantic trajectory dataset. The semantic trajectories of a user may be quite diverse since the user movements may change time to time. However, the main behaviors of a user may be fixed and thus can be discovered. For example, a user regularly goes to the school, but sometimes passes by a gas station. Hence, to identify the user frequent movement behaviors, we perform the sequential pattern mining algorithm *Prefix-Span* [10] on each user's semantic trajectory dataset to mine the frequent semantic trajectories. Take Figure 6 as an example. Given the *Trajectory$_1$* and *Trajectory$_2$* are from a mobile user, her trajectory log will be transformed as the semantic trajectory dataset as shown in Table 1. Suppose that we set the minimum support of *Prefix-Span* algorithm as 60%, the pattern <{Unknown}, {School, Park}, {Hospital}> and all of its subsequences will be mined.

However, it is clear to observe that the longer pattern we mine the more subsequences will be generated due to the downward closure property [9][10]. It leads to biased measure of users' similarity, because all the subsequences of a pattern will be involved in the user similarity calculation. For example, the subsequences of the pattern <{Unknown}, {School, Park}, {Hospital}> are <{Unknown}>, <{School, Park}>, <{Hospital}>, <{Unknown}, {School, Park}>, <{School, Park}, {Hospital}>, and <{Unknown}, {Hospital}>. If we use all of these patterns to represent a mobile user's behaviors, many behaviors will be duplicates, e.g., {School, Park}. Therefore, we only maintain the maximal patterns [9], named *maximal semantic trajectory pattern*, for representing user frequent movement behaviors.

Table 1. An example of Semantic trajectory dataset

Trajectory	**Semantic trajectory**
Trajectory$_1$	<{Unknown}, {School, Park}, {Park}, {Hospital}>
Trajectory$_2$	<{Unknown}, {School, Park}, {Bank}, {Hospital}>

3.3 Semantic Similarity Measurement

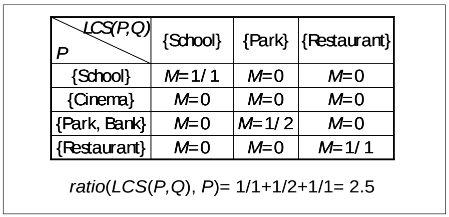
Next we explain how to measure the similarity between two users based on their maximal semantic trajectory pattern sets. We first propose Maximal Semantic Trajectory Pattern Similarity (*MSTP-Similarity*) to measure the similarity between two maximal semantic trajectory patterns. Next, we extend the *MSTP-Similarity* to measure the similarity between two users.

3.3.1 Similarity of two Patterns

Given two Maximal Semantic Trajectory Patterns, we argue that they are more similar when they have more common parts. Thus, we use the Longest Common Sequence (LCS) of these two patterns to represent their longest common part. For example, given a pattern *P* = <{School}, {Cinema}, {Park, Bank}, {Restaurant}> and a pattern *Q* = <{School, Market}, {Park}, {Restaurant}>, their longest common sequence is *LCS(P,Q)* = <{School}, {Park}, {Restaurant}>. Accordingly, we define the *participation ratio* of the common part to a pattern *P* as follows.

$$ratio(LCS(P,Q),P) = \frac{\sum_{i=1}^{|P|}\sum_{j=1}^{|LCS(P,Q)|} M(P_i, LCS_j)}{|P|},$$

$$\text{where}\, M(P_i, LCS_j) = \begin{cases} \dfrac{|P_i \cap LCS_j|}{|P_i|} & \text{, if } LCS_j \text{ is matching to } P_i \\ 0 & \text{, otherwise} \end{cases} \qquad (1)$$



P \ *LCS(P,Q)*	{School}	{Park}	{Restaurant}
{School}	*M*=1/ 1	*M*=0	*M*=0
{Cinema}	*M*=0	*M*=0	*M*=0
{Park, Bank}	*M*=0	*M*=1/ 2	*M*=0
{Restaurant}	*M*=0	*M*=0	*M*=1/ 1

ratio(*LCS*(*P*,*Q*), *P*)= 1/1+1/2+1/1= 2.5

Figure 7. An example of the participation ratio.

Consider the example in Figure 7, where the elements of *LCS(P,Q)*, i.e., {School}, {Park}, and {Restaurant}, are matched with the elements of pattern *P*, i.e., {School}, {Park, Bank}, and {Restaurant}, respectively. Since the element {Park} matches the element {Park, Bank} partially, *M(*{Park, Bank},{Park}*)* is 1/2. Similarly, *M(*{School},{School}*)* is 1, and *M(*{Restaurant },{ Restaurant}*)* is 1. Thus the participation ratio of *LCS(P,Q)* to P will be (1 + 1/2 + 1)/4 = 0.625.

Notice that, based on Equation (1), we need to first find the LCS of two patterns which is quite time-consuming. Therefore, we modify the longest common sequence algorithm to calculate the ratios at the same time (as shown in Figure 8). As we known, dynamic programming is the most popular solution for longest common sequence problem. Typically, a matrix is used to store the LCS at each step of the calculation. Each entry of the matrix is defined as (1), where *E[i, j]* indicates the entry of matrix in column *i* and row *j*.

$$E[i,j]=\begin{cases}0 & \text{, if } i=0 \text{ or } j=0\\ \max(E[i-1,j-1]+|P_i\cap Q_j|, E[i,j-1], E[i-1,j]), & \text{otherwise}\end{cases} \quad (2)$$

```
1   for $i \leftarrow 1$ to $|P|$ do
2       $c[i,0].count \leftarrow 0$, $c[i,0].P\_ratio \leftarrow 0$, $c[i,0].Q\_ratio \leftarrow 0$
3   for $j \leftarrow 1$ to $|Q|$ do
4       $c[0,j].count \leftarrow 0$, $c[0,j].P\_ratio \leftarrow 0$, $c[0,j].Q\_ratio \leftarrow 0$
5   for $i \leftarrow 1$ to $|P|$ do
6       for $j \leftarrow 1$ to $|Q|$ do
7           $temp \leftarrow c[i-1,j-1].count + |P_i \cap Q_j|$
8           if $temp > c[i-1,j].count$ and $temp > c[i,j-1].count$
9               $c[i,j].count \leftarrow temp$
10              $c[i,j].P\_ratio \leftarrow c[i-1,j-1] + \frac{|P_i \cap Q_j|}{|P_i|}$
11              $c[i,j].Q\_ratio \leftarrow c[i-1,j-1] + \frac{|P_i \cap Q_j|}{|Q_i|}$
12          else
13              if $c[i-1,j].count > c[i,j-1].count$
14                  $c[i,j] \leftarrow c[i-1,j]$
15              else
16                  $c[i,j] \leftarrow c[i,j-1]$
17  return $\frac{c[|P|,|Q|].P\_ratio}{|P|}$ and $\frac{c[|P|,|Q|].Q\_ratio}{|Q|}$
```

Figure 8. Modified longest common sequence algorithm.

Q

P

		{School, Market}	{Park}	{Restaurant}
	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
{School}	(0,0,0)	↖ (1,1,0.5)	←(1,1,0.5)	←(1,1,0.5)
{Cinema}	(0,0,0)	↑ (1,1,0.5)	←(1,1,0.5)	←(1,1,0.5)
{Park, Bank}	(0,0,0)	↑ (1,1,0.5)	↖ (2,1.5,1.5)	← (2,1.5,1.5)
{Restaurant}	(0,0,0)	↑ (1,1,0.5)	↑ (2,1.5,1.5)	↖ (3,2.5,2.5)

LCS(*P,Q*)=<{School}, {Park}, {Restaurant}>

ratio(LCS(*P,Q*), *P*)= 2.5/4 = 0.625

ratio(LCS(P,Q), *Q*)= 2.5/3 = 0.833

Figure 9. An example of the modified longest common sequence algorithm.

Such a matrix stores the maximum length of the common sequences at each step. The entries of the matrix are filled row by row when the dynamic programming algorithm is performed. Finally, the value of the last entry is the length of LCS. Instead of employing conventional dynamic programming algorithm to find the longest common sequence, we modify each entry of the matrix, which contains *count*, *P_ratio*, and *Q_ratio* as shown in Figure 9. The *count* is used to store the maximum length of the common sequences in each step, same as the conventional matrix. The *P_ratio*, and *Q_ratio* are used to store the ratio of LCS to the pattern *P* and pattern Q, respectively. Initially, we set each part of each entry in the first row and column as 0 (see Line 1 to 4 of Figure 8). Take As shown in Figure 9, all parts of each entry in the first row and column is set as 0. Then, we cumulate the *P_ratio* and *Q_ratio* when the *count* is increased (see Line 10 and 11 of Figure 8). Take Figure 9 as an example. First, the algorithm process the entry (1,1) , since |{School}∩{School, Market}| = 1, the *count*, first element, will be added by 1. Thus the *P_ratio*, is increased by 1/|{School}| and *Q_ratio* is increased by 1/|{School,Market}|. Then, the algorithm process the entry (1,2). Since |{School}∩{Park}| = 0, the *count*, first element, is not increased. Because the *count* of entry (1,1) is greater than the *count* of entry (0,2), we just copy entry (1,1) to entry (1,2). By this way, we can fill all entries of the matrix as shown in Figure 9. Finally, we only return *P_ratio*/|*P*| and *Q_ratio*/|*Q*| of the last entry (See 17 of Figure 8).

Finally, we calculate the similarity of two patterns, *MSTP-Similarity(P,Q)*, by averaging the participation ratios of their common part to them. Given P and Q, a simple approach is to directly compute the average of the two ratios to P and Q, as shown in Equation (3). Thus, we call this approach *Equal Aveage (EA)*. On the other hand, as shown in Equation (4), we can compute the *Weighted Average (WA),* in proportion to the lengths of the two patterns. The argument is that a longer pattern provides more information about user behaviors than a shorter pattern. Therefore, the longer pattern gives more weight than the shorter one in measuring the similarity between two patterns.

$$\textit{MSTP-Similarity}_{EA}(P,Q) = \frac{ratio(LCS(P,Q),P) + ratio(LCS(P,Q),Q)}{2} \quad (3)$$

$$\textit{MSTP-Similarity}_{WA}(P,Q) = \frac{|P|\times ratio(LCS(P,Q),P) + |Q|\times ratio(LCS(P,Q),Q)}{|P|+|Q|} \quad (4)$$

3.3.2 Similarity between two users

Since a maximal semantic trajectory pattern represents one of a user's real-world semantic behaviors, we consider the similarity between two users in terms of the similarity of their maximal semantic trajectory patterns. When two users have strong similarity between their maximal semantic trajectory patterns, the recommender recommends them as friends. Since a user may possibly possess several maximal semantic trajectory patterns, we extend *MSTP-Similarity* to measure two maximal semantic trajectory pattern sets. Let $S_U = \{M_1, M_2, ..., M_m\}$ and $S_V = \{M'_1, M'_2, ..., M'_n\}$ be the maximal semantic trajectory pattern sets corresponding to the users *U* and *V*, respectively. The user similarity between *U* and *V* is defined by Equation (5).

$$Similarity(U,V) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Weight(M_i, M'_j)\times \textit{MSTP-Similarity}(M_i, M'_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} Weight(M_i, M'_j)} \quad (5)$$

The idea is to obtain a weighted average by taking into account all possible *MSTS-Similarities* between patterns from the two pattern sets. To reflect the patterns' importance, we propose three ideas to form the weighting function: 1) equal weight, 2) weighting by support, and 3) weighting by TFIDF. The equal weight is to set all of weight as 1. In the following, we discuss the weighting schemes by support and TFIDF.

3.3.2.1 Weighting by support

When a sequential pattern is discovered from a sequence data set, its support can be calculated at the same time. Similarly, a maximal semantic trajectory pattern and its support will be mined at the same time. We argue that a pattern with a high support is more important than one with a low support. Thus, we use this information to represent the importance of patterns. To compute the average support between two patterns, we consider (i) the geometric mean as shown in Equation (6), and (ii) the arithmetic average as shown in Equation (7). The idea behind the geometric mean is that a *MSTP-Similarity* is important when both the two patterns are important. On the contrary, the arithmetic average is high as long as one of supports of patterns is high.

$$Weight_{\text{support\_geometric mean}}(M,M') = \sqrt{\text{support}(M)\times\text{support}(M')} \tag{6}$$

$$Weight_{\text{support\_arithmetic average}}(M,M') = \frac{\text{support}(M)+\text{support}(M')}{2} \tag{7}$$

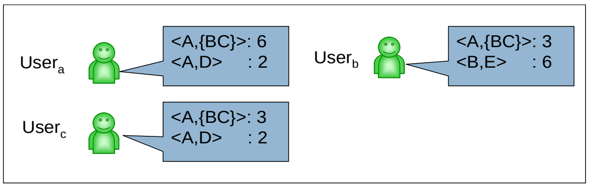


Figure 10. An example of users' pattern set.

Take Figure 10 as an example. There are 3 mobile users along with their mined pattern sets. The support of pattern <A,{BC}> in *User$_a$*'s pattern set is 6, and the support of pattern <A,D> in *User$_c$*'s pattern set is 2. When we calculate the similarity between User$_a$ and User$_c$, the weight of *MSTP-Similarity* can be calculated as follows:

$$Weight_{\text{support\_geometric mean}}(<A,\{BC\}>,<A,D>) = \sqrt{6\times 2}$$

$$Weight_{\text{support\_arithmetic average}}(<A,\{BC\}>,<A,D>) = \frac{6+2}{2}$$

3.3.2.2 Weighting by TFIDF

An alternative idea to support is to treat a pattern set of a user as a document and each pattern in a pattern set as a word, i.e., following the ideas of TFIDF weights in information retrieval to derive the weight for each pattern in a pattern set. Here the term frequency could be determined using the support of the pattern. Thus, the TFIDF value of a pattern can be defined as (10) Take Figure 10 as an example. There are 3 mobile users along with their mined patterns and supports. The TFIDF of pattern <A, {B, C}> in the pattern set of *User$_a$* is evaluated as 0. Similarly, there are two strategies, geometric mean and arithmetic average, to formulate the weighting function as shown in Equation (8) and Equation (9).

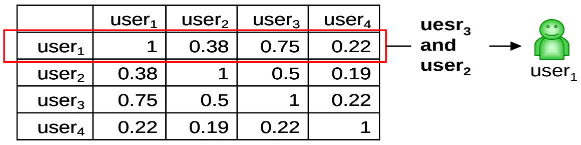
$$Weight_{\text{TFIDF\_geometric mean}}(M,M') = \sqrt{TFIDF(M)\times TFIDF(M')} \tag{8}$$

$$Weight_{\text{TFIDF\_arithmetic average}}(M,M') = \frac{TFIDF(M)+TFIDF(M')}{2} \tag{9}$$

$$TFIDF(M) = \text{support}(M)\times\log\frac{\text{number of total users}}{\text{number of users who have the pattern } M} \tag{10}$$

3.4 Potential Friends Recommendation

Based on the user similarity discussed above, we can build a user similarity matrix which can be used to provide the *k* most similar users to a targeted user. Take Figure 11 as an example. When the *user$_1$* logins to a LBSN site, by adopting our *SemanTraj* friend recommender, the system will recommend *user$_3$* and *user$_2$* to the *user$_1$*, if we set *k* as 2. As shown in the figure, the similarity matrix calculated by our approach has *user$_2$* and *user$_3$* as the two most-similar users to *user$_1$*.



	user$_1$	user$_2$	user$_3$	user$_4$
user$_1$	1	0.38	0.75	0.22
user$_2$	0.38	1	0.5	0.19
user$_3$	0.75	0.5	1	0.22
user$_4$	0.22	0.19	0.22	1

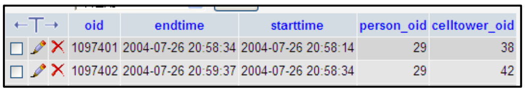
Figure 11. Recommendation based on user similarity

4. EXPERIMENTS

In this section, we conduct a series of experiments to evaluate the performance for the proposed friend recommendation system using MIT reality mining dataset [5]. All the experiments are implemented in Java JDK 1.6 on an Intel Core Quad CPU Q6600 2.40GHz machine with 1GB of memory running Microsoft Windows XP. We first present the data preparation on the MIT reality mining dataset and then introduce the evaluation methodology. Finally, we present our results followed by discussions.

4.1 MIT reality mining dataset

The MIT reality mining dataset is a mobile phone dataset collected by MIT Media Laboratory from 2004 to 2005. The dataset contains 106 mobile users over 500,000 hours (~60 years) of continuous data on daily human behavior. The dataset contains cell trajectory as shown in Figure 12. As we see, the stay time can be derived by calculating difference in timestamp between the user arrive and leave the cell. Thus, we can easily discover the stay cells of each cell trajectory.

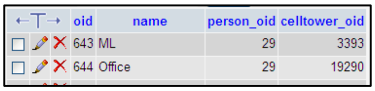


←T→	oid	endtime	starttime	person_oid	celltower_oid
□ ✎ ✕	1097401	2004-07-26 20:58:34	2004-07-26 20:58:14	29	38
□ ✎ ✕	1097402	2004-07-26 20:59:37	2004-07-26 20:58:34	29	42

Figure 12. An example of cell sequence of a mobile user.

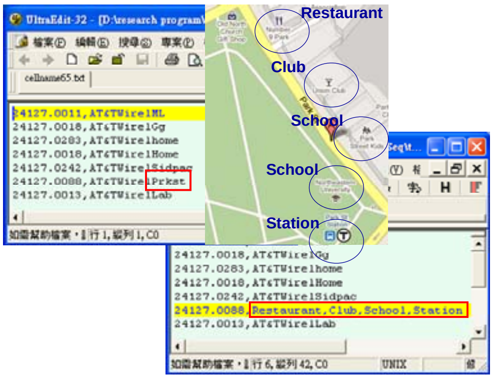
Since this dataset contains user annotated cell names, they inherently are semantic trajectories as shown in Figure 13. However, the annotation terms are very diverse. For example, one annotates a cell as "ML" and someone else annotates it as "Media Lab", even though it's obviously that this cell is MIT Media Laboratory. Besides, many terms are geographic terms such as "Park St.". To stem the annotation log, we use these terms as query term to find suitable semantic terms near these geographic terms. Although we make a lot of efforts to figure out the semantics of the annotation terms in the log, there are unfortunately still many terms which we can not be sure of their meanings. As a consequence, we stem such term as "Unknown". As shown in Figure 14, the term "Prkst" will first be stemmed as

"Park st.". Then, we search the term "Park st." in Google map. Finally, we use the terms "Restaurant", "Club", "School", and "Station" to replace the term "Prkst" as shown in Figure 14. Thus, we can understand the semantic mining of each cell for each user. Then, each stay cell sequence will be transformed to a semantic trajectory by the stemmed annotation log.



←T→			oid	name	person_oid	celltower_oid
□	✎	✕	643	ML	29	3393
□	✎	✕	644	Office	29	19290

Figure 13. An example of the annotation of cells by a user.



UltraEdit-32 - [D:\research program\
檔案(F) 編輯(E) 搜尋(S) 專案(P)
cellname65.txt
如需幫助檔案，| 行 1, 縱列 1, C0
Restaurant
Club
School
School
Station
Park St
SeqM...
如需幫助檔案，| 行 6, 縱列 42, C0
UNIX

```
24127.0011,AT&TWirelML
24127.0018,AT&TWirelGg
24127.0283,AT&TWirelhome
24127.0018,AT&TWirelHome
24127.0242,AT&TWirelSidpac
24127.0088,AT&TWirelPrkst
24127.0013,AT&TWirelLab
```

```
24127.0018,AT&TWirelGg
24127.0283,AT&TWirelhome
24127.0018,AT&TWirelHome
24127.0242,AT&TWirelSidpac
24127.0088,Restaurant,Club,School,Station
24127.0013,AT&TWirelLab
```

Figure 14. An example of the annotation term stemming.

MIT Media Laboratory has conducted an online survey, which was completed by 94 of the 106 Reality Mining mobile users[1]. The survey data present the summarized behavior of a mobile user. Therefore, we use the similarity of two users' survey data as the ground truth. To calculate the similarity of two users' survey data, we treat each mobile user's survey data as a vector, and calculate the Euclidean distance between the two vectors. Thus, each user has a list of Euclidean distances between him and other users. Then, for each list, we normalize the values into the range [0..4] and round them to integers. For example, if a list of Euclidean distances is <8.1, 0, 10.9, 4.1> and the range of Euclidean distance of two users is 0-15.7, the normalized list will be <2, 0, 3, 1>. Finally, for each normalized list, we subtract each value from 5. Following the above example, the normalized list <2, 0, 3, 1> will be transformed as <3, 5, 2, 4>.

Among the 94 mobile users, there are 7 users who do not have cell trajectory logs, and 10 users who do not have cell annotation logs. Thus, after omitting these users, data from the remaining 77 mobile users are used in our experiments. First, we can transform each cell trajectory to a semantic trajectory. Then, a sequential pattern algorithm is performed on each user's semantic trajectory dataset. Finally, we use the Equation (5) to evaluate the similarity of two users based on their semantic pattern sets.

4.2 Evaluation methodology

Our recommendation system is based on the ranking of user similarity and thus can be viewed as an information retrieval system if we consider a user as a query term. Therefore, we employ the Normalized Discounted Cumulative Gain (NDCG) [11] to measure the list of recommended potential friends. For each list of recommended potential friends, we can obtain a score list where the scores are provided by ground truth. Such a list is called relevance vector. For example, for the friends ordered by the recommender as <F1,F2,F3,F4>, the ground truth provides the following relevance vector of scores <2,3,0,1>. That is F1 has a relevance of 2, F2 has a relevance of 3, etc. The discounted cumulative gain of a relevance vector *G* is computed by Equation (11). The premise of DCG is that the highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. Here the parameter b is to control where we start to reduce the relevance value. For example, if the relevance vector is <2,3,0,1> and b is set as 3, the *DCG*[4] is 2+3+(0/log$_3$3)+(1/log$_3$4). (In our experiments, b = 2.)

$$DCG[i]=\begin{cases} G[i], & \text{if } i=1 \\ DCG[i-1]+G[i], & \text{if } i<b \\ DCG[i-1]+\dfrac{G[i]}{\log_b i}, & \text{if } i\geq b \end{cases} \tag{11}$$

NDCG is commonly used in information retrieval to measure the search engine's performance. A higher NDCG value to a list of search results indicates that the highly relevant items have appeared earlier (with higher ranks) in the result list. In particular, NDCG@p, measures the relevance of top *p* as shown in Equation (11).

$$NDCG@p=\frac{DCG[p]}{IDCG[p]} \tag{11}$$

where *IDCG*[*p*] is the *DCG*[*p*] value of ideal ranking list. For example, given a ranking list of 5 items with relevance as <4, 1, 3, 1, 1>, the ideal ranking list of this 5 items is <4, 3, 1, 1, 1>. NDCG ranges from 0 to 1. The higher NDCG is, the better a ranking result list is. In the above example, the NDCG @5 is

$$NDCG@5=\frac{4+1/\log_2 2+3/\log_2 3+1/\log_2 4+1/\log_2 5}{4+3/\log_2 2+1/\log_2 3+1/\log_2 4+1/\log_2 5}=0.913785$$

In our experiment, for the detection of stay cells, we set time threshold as 30 minutes. In the pattern mining step, we set minimum support threshold as 30%. In this study, we consider the semantic information as a critical factor in similarity measurement of two mobile users. Hence, we adapt our proposed system by using maximal sequential pattern mining instead of semantic trajectory pattern mining to generate a baseline for comparison. In other word, we directly perform a maximal sequential pattern mining algorithm on the stay cell sequence set for each mobile user, i.e., the patterns we mine is formed based on ONLY geographic information, i.e., stay cells.

4.3 Experimental results and discussions

This experiment evaluates our approach and geographic similarity under various similarity measurement strategies in terms of NDCG@5. In Figure 15, the NDCG@5 value of our approach, considering semantic information, outperforms geographic similarity in each strategy, respectively. We observed that the effect of weighting strategy of our approach is not significant. The reason is that the semantic trajectory pattern mining step already filters most of noisy semantic behaviors.

[1] In this paper, we do not present the questions of the online survey due to the space limitation.

Moreover, the average strategy of *MSTP-Similarity* is not significant, either. It is because that the length of patterns of the top five similar users may be very similar. If the difference of the length of two patterns is very large, the ratio of the longest common sequence to the longer pattern will be very low. As a result, the *MSTP-similarity* is low. Therefore, it is clear that the length of patterns for the top five similar users may be very similar.

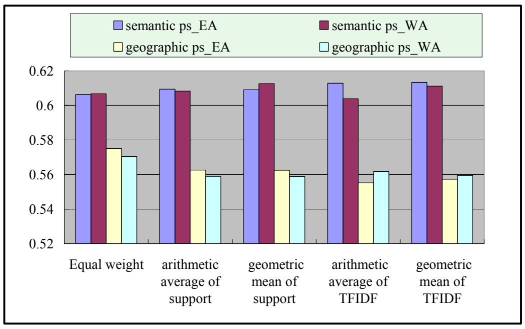


Figure 15. NDCG@5.

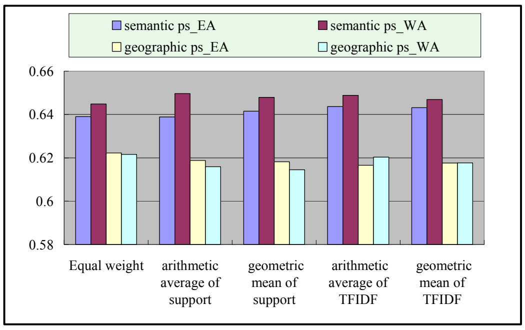


Figure 16. NDCG@10.

We also conduct experiments to analyze our approach and geographic similarity under various similarity measurement strategies in terms of NDCG@10. In Figure 16, the NDCG@10 value of our approach, considering semantic information, outperforms geographic similarity in each strategy, respectively. We also observe that the effect of weighting strategy in our approach is not significant. In the figures, we can observe the impact of the average strategies on *MSTP-Similarity*. As shown, the weighted average strategy outperforms equal average strategy, because there exist some patterns with very different lengths. Therefore, the performance will be better if we consider the information of pattern length.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel framework to support friend recommendation services for location-based social networking systems based on the semantic trajectories of mobile users. The core of our framework is a novel similarity measurement, namely, *Maximal Semantic Trajectory Pattern Similarity* (*MSTP-Similarity*), for measuring the similarity between two semantic trajectory patterns. Accordingly, we extend *MSTP-Similarity* to measure the semantic similarity between mobile users. To our best knowledge, this is the first work aiming at mining user similarity from GPS trajectory data by considering semantic meanings of GPS trajectories. Through a series of experiments, we validate our proposal and show that the proposed friend recommendation framework has excellent performance under various conditions. As for the future work, we plan to design more sophisticated similarity measurements to enhance the quality of friend recommendation systems for LBSNs.

6. ACKNOWLEDGMENTS

This research was supported by National Science Council, Taiwan, R.O.C. under grant no. NSC 99-2631-H-006 -002.

7. REFERENCES

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

[2] GPS route exchange forum: http://www.gpsxchange.com/.

[3] L. O. Alvares, V. Bogorny, A. Palma, B. Kuijpers, B. Moelans, and J. A. F. Macedo. Towards Semantic Trajectory Knowledge Discovery. Technical Report, Hasselt University, Belgium, Oct. 2007.

[4] V. Bogorny, B. Kuijpers, and L. O. Alvares. ST-DMQL: A Semantic Trajectory Data Mining Query Language. International Journal of Geographical Information Science, Vol. 23, No. 10, 1245-1276, Oct. 2009

[5] N. Eagle, A. Pentland, and D. Lazer. Inferring Social Network Structure using Mobile Phone Data. In proceedings of the National Academy of Sciences (PNAS),106(36), pp. 15274-15278, 2009.

[6] J.-G. Lee, J. Han and K.-Y. Whang. Trajectory Clustering: A Partition-and-Group Framework. In Proceedings of International Conference on Management of Data (ACM SIGMOD), pp. 593-604, Jun. 2007.

[7] Q. Li, Y. Zheng, X. Xie,Y. Chen, W. Liu, and W.-Y. Ma. Mining User Similarity Based on Location History. In Proceedings of 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Irvine, CA, USA, Nov. 2008.

[8] E. H.-C. Lu and V. S. Tseng. Mining Cluster-Based Mobile Sequential Patterns in Location-Based Service Environments. In Proceedings of IEEE International Conference on Mobile Data Management (MDM), May. 2009.

[9] C. Luo and S. Chung. Efficient mining of maximal sequential patterns using multiple samples. In proceeding of the 2005 SIAM international conference on data mining (SDM’05), Newport Beach, CA, pp 415–426, 2005

[10] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the 17th International Conference on Data Engineering (ICDE), 2001, 215-224.

[11] D. Manning, P. Raghavan and H. Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.

[12] Y. Zheng, L. Zhang, and X. Xie. Recommending friends and locations based on individual location history. ACM Transaction on the Web, 2010.