# Semantic Trajectory Mining for Location Prediction

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

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
Research on predicting movements of mobile users has attracted a lot of attentions in recent years. Many of those prediction techniques are developed based only on geographic features of mobile users' trajectories. In this paper, we propose a novel approach for predicting the next location of a user's movement based on both the geographic and *semantic* features of users' trajectories. The core idea of our prediction model is based on a novel cluster-based prediction strategy which evaluates the next location of a mobile user based on the frequent behaviors of similar users in the same cluster determined by analyzing users' common behavior in semantic trajectories. Through a comprehensive evaluation by experiments, our proposal 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
Measurement, Experimentation.

## Keywords
Trajectory Database, Trajectory Pattern, Semantic Prediction, Data mining.

## 1. INTRODUCTION
The market of location based services, including navigational services, traffic management and location-based advertisement, have grown rapidly in recent years. Due to the needs of effective marketing and efficient system operations, it is beneficial for these LBSs to be able to forecast the activities a user may perform at the *next location* to visit. Thus, effective and effective location prediction techniques for LBSs targeting on mobile users are desirable.

In recent years, a new breed of location prediction methods, called *general-pattern-based prediction*, have emerged. Such prediction methods usually use the *frequent common behaviors* of users mined from collections of mobile users' GPS trajectories, to predict the next move of a user. Figure 1 shows some examples of the GPS trajectory, which typically consists of a sequence of spatio-temporal points (in form of latitude, longitude, and time). Among the general-pattern-based prediction methods, mobile sequential pattern mining techniques [6] [9] have been widely used for analyzing patterns in mobile user movement data sets. However, they tend to predict popular locations where most people visited, leading to the imbalanced data problem [13]. Additionally, these pattern-based prediction methods usually make a prediction only if an anticipated movement has a full match with the prefix of a pattern, leading to loss of recall in predictions.

Although the issues of discovering mobile users' frequent patterns in their trajectories have been discussed in the literature, existing studies mostly consider only on the *geographic* features of user trajectories [6] [9]. Notice that a *geographic trajectory* typically consists of a sequence of geographic points (represented as <latitude, longitude>) tagged with timestamps. As a result, the frequent pattern of user movement behavior based on geographic trajectory is constrained by the geographic properties of the

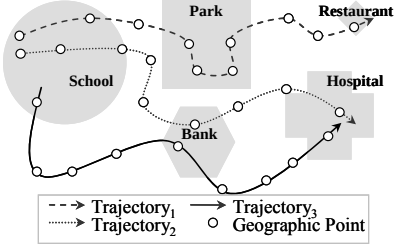



**Figure 1. An example of semantic trajectory.**

trajectory data. For example, as Figure 1 shows, the geographic distance and shape between $Trajectory_1$ and $Trajectory_2$ is closer and more similar than that between $Trajectory_1$ and $Trajectory_3$. Thus, some location prediction techniques would predict the destination of $Trajectory_1$ based on its geographical similarity to $Trajectory_2$. Additionally, such prediction strategies only consider the previously visited locations and thus do not work well when previously unvisited locations are considered. We argue that merely using geographic information to predict the destination of a trajectory or a user's next location is not sufficient.

The notion of *semantic trajectory* has been proposed by Alvares *et al.* [1] [2]. Basically, a semantic trajectory consists of a sequence of locations labeled with semantic tags (called *semantic locations*) to capture the landmarks passed by [5]. These semantic tags of locations imply the activities being carried out in the trajectory. Consider Figure 1 where trajectories are tagged with a number of semantic tags such as School, Park, etc. We observe that both $Trajectory_2$ and $Trajectory_3$ can be denoted by the sequence <School, Bank, Hospital>, implying that the semantic *behaviors* of users in $Trajectory_2$ and $Trajectory_3$ are quite the same. Thus, we exploit their similarity in visited semantic locations to predict the next locations of mobile users.

To support location prediction based on the semantic trajectories of mobile users, we propose a novel location prediction framework, called *SemanPredict,* to evaluate the next location of a user's movement. The framework consists of two major modules: i) offline mining module, and ii) on-line prediction module. In the offline mining module, we adopt the notion of *stay locations* to represent the users' movement behavior. To extract the semantic feature from individual user's movement behavior, we mine the semantic trajectory patterns for each individual user. Moreover, we form user clusters based on the notion of *semantic trajectory similarity* we proposed. Furthermore, we mine the frequent trajectory patterns of users in the same cluster based on their geographic features. In the on-line prediction module, based on these semantic and geographic patterns, we develop a novel cluster-based prediction technique to predict a mobile user's next location. To our best knowledge, this is the first work on predicting a mobile user's next location by exploiting both geographic and semantic features of trajectories. Through an experimental evaluation, we show that the proposed location prediction approach delivers excellent performance.

The contributions of our research are six-fold.

- We propose the *SemanPredict* framework, a new approach for mobile users' movement behavior mining and prediction. The problems and ideas in *SemanPredict* have not been explored previously in the research community.
- We develop data mining algorithms to discover semantic trajectory patterns for individual users and geographic trajectory patterns for clusters of similar users.
- We employ the notion of *semantic trajectory similarity* we proposed to cluster similar users together.
- We develop index structures based on prefix tree to represent semantic and geographic trajectory patterns in a compact form in order to facilitate efficient prediction computation.
- Based on the semantic and geographic trajectory patterns, we propose a novel location prediction strategy to predict a user's next location.
- We use a real dataset, namely, MIT reality dataset [3], in a series of experiments to evaluate the performance of our proposal. The results show superior performance over other location prediction techniques in terms of precision and recall.

The rest of this paper is organized as follows. We briefly review the related work in Section 2 and provide an overview of our prediction framework in Section 3. We detail the proposed Semantic Mining and Geographic Mining in Section 4 and describe our location prediction technique in Section 5. Finally, we present the evaluation result of our empirical performance study in Section 6 and discuss our conclusions and future work in Section 7.

## 2. RELATED WORK

Many data mining studies have discussed the problems of predicting the next location where a mobile user moves to. Personal-based prediction [4] [11] [12] and general-based prediction [7] [8] [9] [16] [17] are two approaches often adopted in this problem domain. The personal-based prediction approach considers movement behavior of each individual as independent and thus uses only the movements of an individual user to predict his/her next location. On the contrary, the general-based prediction makes a prediction based on the common movement behavior of general mobile users. In [4], Jeung *et al.* propose an innovative approach which forecasts future locations of a user by combining predefined motion functions, i.e., linear or non-linear models that capture object movements as sophisticated mathematical formulas, with the movement patterns of the user, extracted by a modified version of the *Apriori* algorithm. In [11], Yavas *et al.* mine the movement patterns of an individual user to form association rules and use these rules to make location prediction. Additionally, they consider the support and confidence in selecting the association rules for making predictions. In [12], Ye *et al.* propose a novel pattern, called Individual Life Pattern, which is mined form individual trajectory data, and they uses such pattern to describe and model the mobile users' periodic behaviors. In [7], Morzy uses a modified version of *Apriori* algorithm to generate association rules, and in [8], he uses a modified version of *PrefixSpan* algorithm to discover frequent patterns of users' movements for generating the prediction rules. The matching functions employed in these previous works are based on the notions of support and confidence. Although all of Morzy's approaches have considered temporal information and location hierarchy, they do not take into account the semantic tags of locations. In [9], Monreale *et al.* proposes a method aiming to predict with a certain level of accuracy the next location of a moving object. The movement patterns extracted for prediction covers three different movement behaviors, including order of locations, travel time, and frequency of user visits. In [16], Zheng *et al.* uses a HITS-based model to mine users' interesting location and detect users' travel sequence to make locations prediction, and in [17], they consider the location correlation for generating the users' interesting locations and travel sequence. Note that the above-mentioned prediction methods are based on geographic information only. On the contrary, our proposal predicts the next location of a user based on both geographic and semantic information in trajectories.

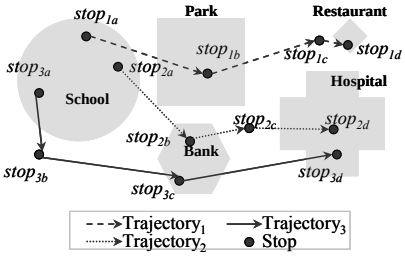



**Figure 2. An example of semantic trajectory.**

In recent years, a number of studies on *semantic trajectory data mining* have appeared in the literature [1] [2]. In [1], Alvares *et al.* propose to explore the geographic semantic information to mine semantic trajectory patterns from mobile users' movement histories. First, they discover the *stops* of each trajectory and map these stops to semantic landmarks to transform geographic trajectories into semantic trajectories. By applying a sequential pattern mining algorithm on semantic trajectories, they obtain frequent patterns, namely, *semantic trajectory patterns*, to represent the frequent semantic behaviors of mobile users. In [2], Bogorny *et al.* use a hierarchy of geographic semantic information to discover more interesting patterns. Notice that the notion of stops in the above-mentioned works only considers the aspect of 'stay' in stops but not the 'positions' of these stops in geographic space. As a result, many unknown stops are generated. For example, as shown in Figure 2, $stop_{1c}$, $stop_{2c}$, and $stop_{3b}$ are not associated with any semantic landmark and thus marked as *Unknown*. 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. Thus, in our work, 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> instead.

Besides, a feature vector is proposed by Zheng to describe the semantics of each location. Based on the feature vector, the semantic similarity between two mobile users could be calculated. In addition to the GPS trajectory, Ying *et al.* [14] also exploit the *cell trajectory* to derive the semantic similarity between two mobile users. The cell trajectory consists of a sequence of spatio-temporal points in form of cell station ID, arrive time, and leave time as shown in Figure 3. They propose a novel similarity measurement, namely, *Maximal Semantic Trajectory Pattern Similarity* (*MSTP-Similarity*) to evaluate the user similarity. As such, the similarity of two mobile users, even if they live in different cities, may be evaluated based on their similar semantic trajectory patterns.

# 3. OVERVIEW OF *SemanPredict*

With the notion of semantic trajectory, we propose a novel location prediction framework, namely, *SemanPredict*, based on both the geographic and semantic features in trajectories. The proposed approach works for locations where the users may have never visited, e.g., a location in other cities. The *SemanPredict* framework consists of 1) an offline training module, and 2) an online prediction module.

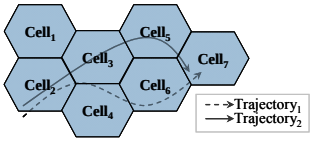



**Figure 3. An example of cell trajectories.**

Figure 4 shows the framework and its flow of data processing. The idea is to explore the activities of mobile users, captured in semantic trajectories, to improve accuracy of location prediction. As shown, the training module includes three steps. The first step, called *data preprocessing*, transforms each user's trajectories as *stay location sequences*. The second step, called *semantic mining*, extracts users' semantic behaviors (as 'semantic trajectory patterns' which will be detailed later). It also obtains user clusters based on the semantic behavior similarity of users. The third step, called *geographic mining*, extracts the geographic behaviors of users in each cluster (as 'stay location patterns' which will be detailed later). In the online module, we propose a scoring function to evaluate the probability for a location to be the next location. Here, we consider not only geographic information but also semantic information. First, we calculate the geographic score and derive several candidate paths. Then, the semantic score of each candidate path is evaluated. Finally, we compute a weighted average of geographic score and semantic score for each candidate path to select the most probable path for predicting the next location in a user's move.

# 4. OFFLINE TRANING MODULE

In this section, we propose an approach to extract the users' frequent movement behaviors which includes the semantic behavior information for individual users and the geographic behavior information for clusters of similar users. We mine a kind of frequent patterns, called semantic *trajectory patterns* [1] [14], from trajectories of individual users and adopt a prefix tree, called *semantic trajectory pattern tree*, to compactly represent a collection of semantic trajectory patterns. Based on individual semantic information (i.e., the semantic trajectory patterns and their support values), we cluster mobile users. For each cluster, the sequential pattern mining is used to extract cluster geographic information, called stay location patterns. Similarly, we also adopt a prefix tree to compactly represent a collection of stay location patterns. As mentioned earlier, this mining module consists of 1) Data Preprocessing step, 2) Semantic Mining step, and 3) Geographic Mining step.

## 4.1 Data Preprocessing

To the data preprocessing step transforms each user's GPS trajectories into *stay location sequences*. We argue that most activities of a mobile user are usually performed at where the user stays. For example, a user may stay with a café to have a drink.

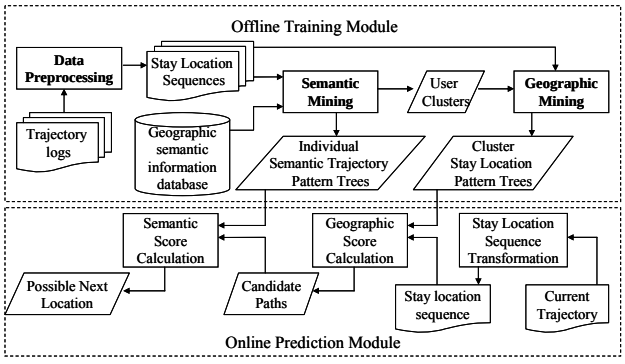



**Figure 4. The *SemanPredict* framework for location prediction**

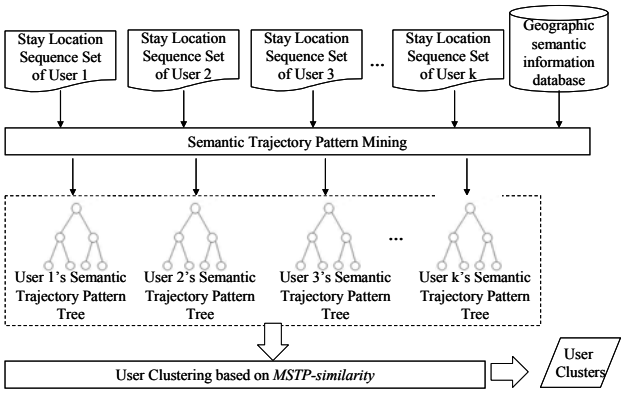



**Figure 5. A work flow of semantic mining**

Thus, we have to first capture the stay locations where a user stops for a while.

Our framework is able to deal with both the GPS trajectories and cell trajectories [12]. For GPS trajectory, we follow Zheng *et al.*'s work [15] to discover *stay points* from users' GPS trajectories. Then, a density-based clustering algorithm is performed on these stay points to obtain stay locations. For cell trajectories, we follow Ying *et al.*'s approach [14] which treats a cell as a geographic location. The stay time in a cell is derived by calculating the difference between the time a user arrives in and leaves the cell. A user-specified time threshold is used to filter the cells with stay time shorter than the threshold. The remaining cells are further filtered by the number of users passed through (i.e., a crowd threshold). Finally, the stay locations (i.e., the cells with stay time equal or greater than the time threshold and the number of visitors equal or greater than the crowd threshold) are obtained and each trajectory is transformed into a *stay locations* sequence. Take Figure 6 as an example. $Trajectory_1$, $Trajectory_2$ and $Trajectory_3$ are transformed into the sequences <*Stay Location*$_1$, *Stay Location*$_5$, *Stay Location*$_6$>, <*Stay Location*$_0$, *Stay Location*$_1$, *Stay Location*$_4$, *Stay Location*$_3$>, and <*Stay Location*$_0$, *Stay Location*$_1$, *Stay Location*$_2$, *Stay Location*$_3$>, respectively.

## 4.2 Semantic Mining

In this section we describe how to extract semantic trajectory patterns from a user's stay location sequences and build semantic trajectory pattern tree based on the discovered patterns. Figure 5 shows the flow of semantic information extraction. We can observe that there are two main steps in the flow. First, we mine semantic trajectory pattern form each user's stay location sequence set. Then, we perform a hierarchical clustering method to cluster users, where the user's similarity is based on *MSTP-Similarity* [14].

### *4.2.1 Semantic Trajectory Pattern Mining*

We follow Ying *et al.*'s approach [14] to mine semantic trajectory pattern from each user's stay location sequences. A geographic semantic information database (GSID) is used to assign semantic labels to the discovered stay locations. The GSID is a customized spatial database which stores the semantic information of landmarks that we collect via Google Map (alternatively, a gazetteer can be used as a general-purpose GSID for this operation.) In our GSID, we store landmarks, their geographic scopes, and the associated semantic labels. In this paper, we use some general categories of the landmarks as their semantic labels.

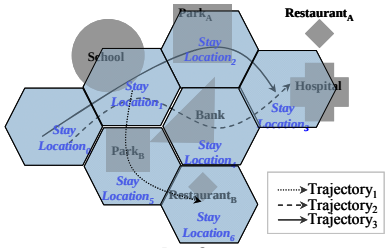



**Figure 6. An example of semantic trajectories.**

If a stay location overlaps one or several landmarks stored in the GSID, the semantic labels of these landmarks are assigned to this stay location. Take Figure 6 as an example, the semantic label of the landmark $\mathbf{Park_B}$ is "Park". Since *Stay Location*$_5$ overlaps the landmark $\mathbf{Park_B}$ and **Bank**, the semantic labels "Park" and "Bank" are assigned to *Stay Location*$_5$. Similarly, we will assign the semantic label "School" to *Stay Location*$_1$. It is possible that a stay location overlaps none of landmark. For example, in Figure 6, there is no landmark overlapped with *Stay Location*$_0$. In this case, we assign the semantic label "Unknown" to the stay location. After assigning semantic labels to the stay location, a stay location sequence can be transformed into a *semantic trajectory*. For example, the stay location sequence <*Stay Location*$_0$, *Stay Location*$_1$, *Stay Location*$_4$, *Stay Location*$_3$> is transformed as <Unknown, School, Park, Hospital>.

After transforming each stay location sequence into a semantic trajectory, each user's stay location sequences are transformed into 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 exhibit some patterns and thus can be discovered. For example, a user goes to her school regularly and sometimes passes by a gas station. Hence, to identify the user frequent movement behaviors, we apply 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 $Trajectory_1$ and $Trajectory_2$ of a mobile user, her trajectory log is transformed into the semantic trajectory dataset as shown in Table 1. Suppose that we set the minimum support of *Prefix-Span* algorithm as 50%, the patterns <Unknown, School, Park, Hospital>, <School, {Bank, Park}> and all of its subsequences are discovered as frequent patterns.

**Table 1. An example of semantic trajectory dataset**

| Trajectory | Semantic trajectory |
|---|---|
| $Trajectory_1$ | < School, {Bank, Park}, Restaurant > |
| $Trajectory_2$ | < Unknown, School, {Bank, Park}, Hospital> |
| $Trajectory_3$ | <Unknown, School, Park, Hospital> |

Such patterns, called *semantic trajectory patterns*, could provide several decision rules for location prediction. For example, if a pattern <Unknown, School, Park, Hospital> is discovered from a mobile user's semantic trajectory set, we can predict that he/she may go to a hospital after going to a school and then to a park. Therefore, by matching a mobile user's recent moves to his/her semantic trajectory patterns, we can predict the semantic label of her next location. However, it is clear to observe that the longer pattern we mine the more subsequences will be generated due to the downward closure property [10]. It leads to a loss of efficiency because all the subsequences of a long pattern need to be considered in the next location prediction. For example, the

```
Input: A semantic trajectory pattern set STP-Set
Output: A semantic trajectory pattern tree STP-Tree
 1  root ← CreateNode(∅,∅,∅)
 2  foreach semantic trajectory pattern STP in STP-Set do
 3      node ← root
 4      foreach semantic S in STP do
 5          if ∃ a child nc of node s.t. S ⊆ nc.semantic then
 6              node ← nc
 7              if S is the last element in STP then
 8                  node.support = STP.support
 9              end
10          else
11              child ← CreateNode(S, STP.support ,∅)
12              node.appendChild(child)
13              node ← child
14          end
15      end
16  end
17  return root
```

**Figure 7. *STP-Tree*Building algorithm.**

subsequences of the pattern <School, Park, Hospital> are <School>, <Park>, <Hospital>, <School, Park>, <School, Hospital>, and <Park, Hospital>. It is very time-consuming to match the current move of a mobile user to all his/her semantic trajectory patterns one by one. To make the prediction phase efficient, we adopted a prefix tree, named semantic trajectory pattern tree (*STP-Tree*), to compactly represent a collection of semantic trajectory patterns. Note that the path of an *STP-Tree* indicates a decision rule. The *STP-Tree* is a kind of decision tree, where each node *v* consists of tree element, semantic set, support, and children.

The *STP-Tree*Building algorithm, shown in Figure 7, describes how to build the STP-Tree from a semantic trajectory patterns set (*STP-Set*). In the following, we introduce the notion of prefix of a semantic trajectory pattern. For simplicity, we consider a semantic trajectory pattern as a sequence of semantic labels. Each semantic trajectory pattern belonging to the *STP-Set* is inserted into the *STP-Tree*. Intuitively, given a semantic trajectory pattern *STP*, we search the tree for the path corresponding to the longest prefix of *STP*. Next, we append a branch to cover the remaining elements of *STP* in this path. A semantic trajectory pattern is appended to a path in the tree if this path is a prefix of semantic trajectory pattern. When the pattern is appended to a path, the support value will be updated if the support value of pattern is greater than the support value of the node (see Line 5 to 9 of Figure 7). The *CreateNode(semantic, support, children)* function returns the node which stores the semantic label, support value, and children list. The *appendChild(child)* procedure appends another node to the children list of a node (see Line 10 to 13 of Figure 7).

Take Table 1 as an example. Given that the $Trajectory_1$ and $Trajectory_2$ are from a mobile user, his/her semantic trajectory pattern will be mined from the semantic trajectory dataset. Suppose that we set the minimum support of *Prefix-Span* algorithm as 50%, all the patterns will be mined along with their support values as shown in Table 2. Figure 8 shows the corresponding semantic trajectory pattern tree. Notice that a path may group together several semantic trajectory patterns. For instance, the path ({Park, Bank}, 1.0) → (Hospital, 0.667) in the semantic trajectory pattern tree represents both the semantic trajectory patterns <Park, Hospital>, <Park>, <Bank>, <{Park, Bank}>, and <Hospital>. Since the prefix tree is a compact representation of a semantic trajectory patterns set. The prefix tree may group the patterns with different support values in one node, such as <Park>, <Bank>, and <{Park, Bank}>. In this case, we use the maximum of the support values of the patterns as the support value of the node. Moreover, the path with only one node will be eliminated from the pattern tree, e.g., the pattern <Hospital> is not shown in the pattern tree.

**Table 2. An example of semantic trajectory pattern set**

| Semantic Trajectory Pattern | Support |
|---|---|
| <Unknown> | 2/3 = 0.667 |
| <School> | 3/3 = 1.0 |
| <Park> | 3/3 = 1.0 |
| <Hospital> | 2/3 = 0.667 |
| <Bank> | 2/3 = 0.667 |
| <{Park, Bank}> | 2/3 = 0.667 |
| <Unknown, School> | 2/3 = 0.667 |
| <Unknown, Park> | 2/3 = 0.667 |
| <Unknown, Hospital> | 2/3 = 0.667 |
| <School, Park> | 3/3 = 1.0 |
| <School, Bank> | 2/3 = 0.667 |
| <School, Hospital> | 2/3 = 0.667 |
| <School, {Park, Bank}> | 2/3 = 0.667 |
| <Park, Hospital> | 2/3 = 0.667 |
| <Unknown, School, Park> | 2/3 = 0.667 |
| <Unknown, School, Hospital> | 2/3 = 0.667 |
| <Unknown, Park, Hospital> | 2/3 = 0.667 |
| <School, Park, Hospital> | 2/3 = 0.667 |
| <Unknown, School, Park, Hospital> | 2/3 = 0.667 |

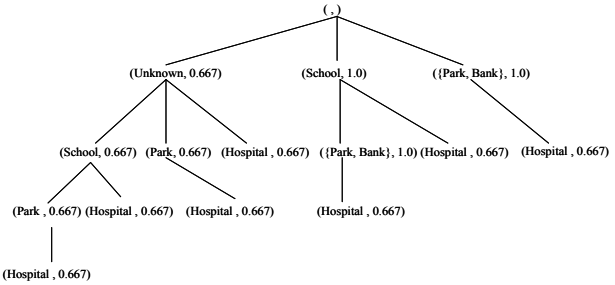



**Figure 8. An example of semantic trajectory pattern tree**

### *4.2.2 Similar User Clustering*

Next we describe the clustering process in semantic mining that clusters mobile user based on their semantic trajectory patterns. We argue that each user's pattern set represents his/her semantic behavior which, i.e., the mobile user's frequent activity behavior. By clustering users with similar semantic behaviors together, the next location of a mobile user can be predicted not only from his/her own past movement behavior but also from that of other mobile users exhibiting similar semantic behaviors.

We measure the similarity between two mobile users by the notion of *Maximal Semantic Trajectory Pattern Similarity* (*MSTP-Similarity*) [14]. Based on *MSTP-Similarity,* two trajectories are more similar when they have more common parts. Given two semantic trajectory patterns, thus, we use the Longest Common Sequence (LCS) of these two patterns to represent their longest common part. For example, given a pattern *P* =

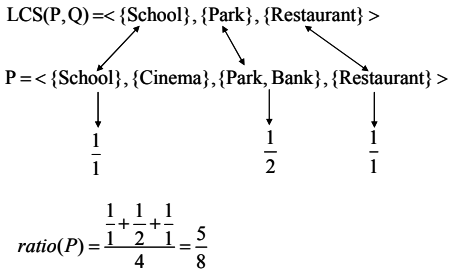



**Figure 9. An example of semantic trajectories.**

<{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, the *participation ratio* of the common part to a pattern $P$ is illustrated in Figure 9. As shown, 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, the ratio of {Park} to {Park, Bank} is 1/2. Similarly, {School} to {School} is 1, and {Restaurant} to {Restaurant} 1. Thus the participation ratio of *LCS(P,Q)* to $P$ will be (1 + 1/2 + 1)/4 = 0.625. The similarity of two patterns, *MSTP-Similarity(P,Q)*, is calculated by averaging the participation ratios of their common part to them. A weighted average of all possible *MSTS-Similarities* between patterns of two users is used to measure their similarity.

Based on mobile users' *MSTP-Similarity*, we then cluster mobile users. Since we only take into account the users' similarity, partition-based clustering methods, such as k-means, fuzzy c-means, are not applicable. Moreover, to our best knowledge, density-based clustering techniques may results in noises not belonging to any cluster. Consequently, a 'noisy mobile user' cannot be processed in the following steps. Therefore we use a hierarchical clustering method, namely, *complete linkage clustering*, to cluster mobile users. This clustering method does not generate noises, which ensures that all the mobile users are supported by our prediction technique.

## 4.3 Geographic Mining

Although semantic mining discovers users' semantic trajectory patterns, they can not be used directly for location prediction since locations are not deductable from the semantic labels. To overcome this problem, we mine the geographic information from users' stay location sequences. Figure 10 shows the flow of data processing within the Geographic Mining step. While we aim to take into account the common frequent behaviors of mobile users, considering the frequent behavior of all general users may cause imbalanced data problem. Hence, we consider the clusters resulted from the semantic mining to aggregate the stay location sequences of mobile users. As shown in Figure 10. We then perform a sequential pattern mining algorithm *Prefix-Span* [10] on each cluster's semantic stay location sequences to mine the frequent stay location sequence, called *stay location pattern*. Similarly, the longer patterns we discover the more subsequences are generated due to the downward closure property [10]. It leads to a loss of efficiency because all the subsequences of a long pattern are to be checked in the next location prediction. Therefore, we also adopt a prefix tree, called stay location pattern tree (*SLP-Tree*), to compactly represent a collection of stay location patterns. Consider the example in Figure 6. Suppose that we set the minimum support of *Prefix-Span* algorithm [10] as 50%, the patterns we mine are shown in Table 3. We also perform the *STP-Tree*Building algorithm, shown in Figure 7, on each stay location pattern set of each cluster to build an *SLP-Tree*. Figure 11 shows the corresponding stay location pattern tree. Similarly, the paths with only one node are not included in the pattern tree.

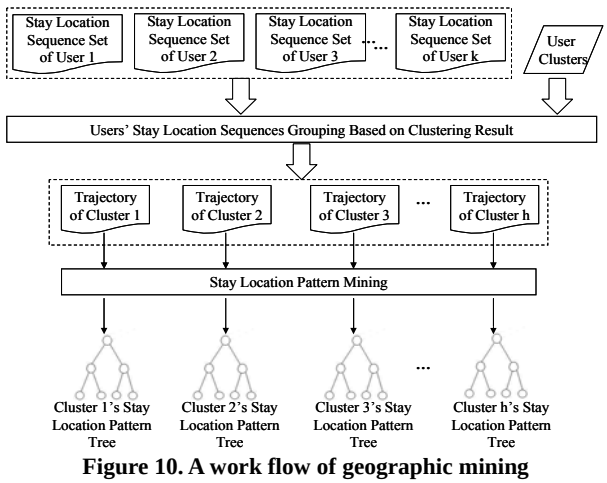



**Figure 10. A work flow of geographic mining**

**Table 3. An example of stay location pattern set**

| Stay Location Pattern | Support |
|---|---|
| <*Stay Location*$_0$> | 0.667 |
| <*Stay Location*$_1$> | 1.0 |
| < *Stay Location*$_3$> | 0.667 |
| <*Stay Location*$_0$, *Stay Location*$_1$> | 0.667 |
| <*Stay Location*$_1$, *Stay Location*$_3$> | 0.667 |
| <*Stay Location*$_0$, *Stay Location*$_3$> | 0.667 |
| < *Stay Location*$_0$, *Stay Location*$_1$, *Stay Location*$_3$> | 0.667 |

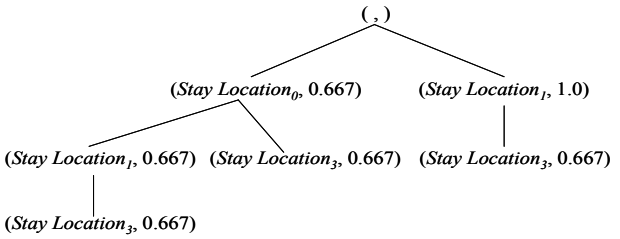



**Figure 11. An example of stay location pattern tree**

# 5. ON-LINE PREDICTION MODULE

Given a mobile user, the on-line prediction module predicts her next stay location based on the stay location pattern tree of her cluster and her own semantic trajectory pattern tree. Given these two pattern trees, the geographic information (i.e., the stay location patterns) of the cluster which the mobile user belongs to and the semantic information (i.e., the semantic trajectory patterns) of the mobile user herself can be incorporated in the prediction. Thus, given the trajectory of a user's recent moves, we compute the best matching scores of candidate paths in these two pattern trees. The matching scores are computed by *a weighted average* of *GeographicScore* and *SemanticScore*, as defined in Equation (1) below.

$$Score = \beta \times GeographicScore + (1-\beta) \times SemanticScore, \quad where\ 0 < \beta \le 1 \qquad (1)$$

Here the score of geographic behavior (*GeographicScore*) measures how well the current geographic behavior of the user matches the stay location patterns in the user's cluster (i.e., paths in the stay location pattern tree (*SLP-Tree*). A path with *GeographicScore* greater than 0 is a *candidate path*. We further transform all the candidate paths into semantic sequences in order to measure their score of semantic behavior (*SemanticScore*) matching the semantic behavior of the user (using the user's personal semantic trajectory pattern tree).

## 5.1 Score of Geographic Behavior

In order to simplify the matching process, the current user's recent moves are transformed into a stay location sequence. Moreover, since the stay location sequence may consist of too many stay locations, it is very time consuming to consider all possible subsequences of the stay location sequence in the matching step. Therefore, we propose a *partial matching* strategy which does not consider all the possible subsequences of the stay location sequence. Instead, the score of geographic behavior (*GeographicScore*) captures three heuristics: 1) outdated moves may potentially deteriorate the precision of predictions; 2) more recent moves potentially have more important impacts on predictions; and 3) the matching path with a higher support and a higher length may provide a greater confidence for predictions. Given a mobile user's stay location sequence $S$ and a matching path $P$ in *SLP-Tree*, we propose a weighted scoring function, *mScore(P, S),* as defined in Equation (2).

$$GeographicScore(P,S)=\sum_{i=1}^{|P|}\sum_{j=k}^{|S|}\alpha^{|S|-j}\times mScore(P_i,S_j),$$
$$\text{where } mScore(P_i,S_j)=\begin{cases}P_i.\text{support} & \text{, if } S_j \text{ is matching to } P_i\\ 0 & \text{, otherwise}\end{cases} \quad (2)$$

In Equation (2), the parameter $\alpha$ is used to exponentially decay the importance of each matched location in the pattern over time. If we set $\alpha$ as 1, the importance of each matched location will not be decayed over time. Based on Equation (2), we need to traverse all paths in *SLP-Tree* which is quite time-consuming. Therefore, we develop a depth-first search algorithm to calculate the *GeographicScore* at the same time (as shown in Figure 13). In the algorithm, a stack is used to store the traversed path at each step of the process. Each entry of the stack indicates an element of the path matched with the stay location sequence.

In our depth-first search algorithm, a sequence set, named *CandidateSet*, is used to stores the matched traversal path in the *SLP-Tree* and its *GeographicScore* at each step. Initially, we set *CandidateSet* as empty, *Candidate* as empty, and *Score* as 0 (see Line 1 to 5 of Figure 13). As shown in Figure 12, we first traverse the whole stay location sequence *S*, i.e., the user current movement, in the given stay location pattern tree *SLP-Tree* ($k$=1). Since no path starts with *Stay Location*$_3$, the score of the candidate path will be set as 0 and the matching path will not be stored in *CandidateSet* (see Line 6 to 15 of Figure 13). Then we ignore the first $k$-1 element of stay location sequence and re-traverse the stay location pattern tree *SLP-Tree* (see Line 2 of Figure 13). Take Figure 12 as an example. Suppose we set $\alpha$=0.8. As shown in Table 4, when $k$=1, the mobile user's stay location sequence, i.e., <*Stay Location*$_3$, *Stay Location*$_1$, and *Stay Location*$_0$>, are not matched with any path in the pattern tree, and the score of geographic behavior, *GeographicScore*, will be evaluated as 0. When $k$=2, the stay location sequence is matched with the path (Stay Location$_0$, 1.0) → (Stay Location$_1$, 0.9), and the *GeographicScore* will be evaluated as 0.8 × 1.0 + 0.9. When $k$=3, the stay location sequence is matched with the path (Stay Location$_1$, 1.0), and the *GeographicScore* will be evaluated as 1.0.

**Table 4. An example of candidate path set**

| Candidate paths | *GeographicScore* |
|---|---|
| (*Stay Location*$_3$) | 0 |
| (*Stay Location*$_0$) → (*Stay Location*$_1$) | 0.8 × 0.667 + 0.667 = 1.2 |
| (*Stay Location*$_1$) | 1.0 |

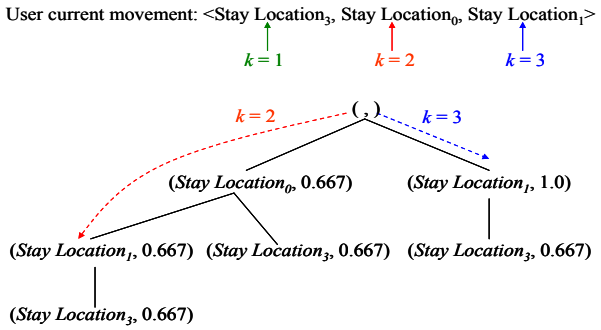



**Figure 12. the score of geographic behavior.**

## 5.2 Score of Semantic Behavior

As mentioned earlier, merely using the geographic information in the *SLP-Tree* to find a user's possible next location is not sufficient. Therefore, we use the user's semantic trajectory pattern tree, *STP-Tree*, to adjust the prediction result. First, we transform each candidate path obtained by traversing the *SLP-Tree* to a semantic path (as shown in Table 5). For example, suppose that for some mobile user, the semantic label of *Stay Location*$_0$, *Stay Location*$_1$, and *Stay Location*$_3$ are "Unknown", "School", and "Hospital", respectively. The candidate paths in Table 4 are

```
Input:  A stay location pattern tree  SLP-Tree
        A stay location sequence S
        Discount parameter α
Output: A set of candidate path along with GeographicScore
 1  CandidateSet ← ∅
 2  for k ← 1 to |S|
 3      node ← SLP-Tree.root
 4      Candidate.Sequence ← ∅
 5      Candidate.Score ← 0
 6      for j ← k to |S|
 7          if ∃ a child nc of node s.t. Sj=nc.location then
 8              node ← nc
 9              Candidate.Sequence.append (nc.location)
10              Candidate.Score ← Candidate.Score +
11                                  (α^(|S|-j) × node.support)
12          end
13      end
14      if j=|S| and Candidate.Score>0 then
15          CandidateSet.add(Candidate)
16      end
17  end
18  return CandidateSet
```

**Figure 13. Depth-first search algorithm.**

transformed into semantic candidate paths as shown in Table 5.

**Table 5. An example of transforming candidate path set**

| Candidate Paths | Semantic Candidate Paths |
|---|---|
| (*Stay Location*$_0$) → (*Stay Location*$_1$) | (Unknown) → (School) |
| (*Stay Location*$_1$) | (School) |

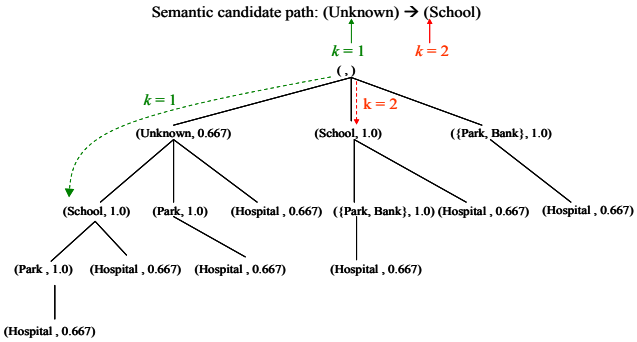



**Figure 14. The score of semantic behavior.**

We consider each semantic candidate path and semantic trajectory pattern tree, *STP-Tree*, as inputs to our depth-first search algorithm. As shown in Figure 14, we first traverse the semantic candidate path (i.e., $k$=1) in the given semantic trajectory pattern tree *STP-Tree*. Then we ignore the first $k$-1 element of the semantic candidate path and re-traverse the semantic trajectory pattern tree *STP-Tree*. Take Figure 14 as an example. Suppose that we set $\alpha$=0.8 and the semantic candidate path is (Unknown) → (School). When $k$=1, the semantic candidate path (Unknown) → (School) matches with the path (Unknown, 0.667) → (School, 0.667), and the score of semantic behavior, *SemanticScore*, is evaluated as 0.8 × 0.667 + 0.667. When $k$=2, the semantic candidate path matches with the path (School, 1.0), and the *SemanticScore* is evaluated as 1.0. Here, we use the highest score to estimate the *SemanticScore* of the semantic candidate path. For example, the *SemanticScores* of all semantic candidate paths are shown in Table 6.

Finally, we use Equation (2) to evaluate the *score* of each candidate path. Consider Table 4 and Table 6 which store the *GeographicScore* and *SemanticScore* of each candidate path for a mobile user, respectively. Suppose we set $\beta$=0.4. The *score* of candidate path (*Stay Location*$_0$) → (*Stay Location*$_1$) is evaluated as 0.4 × 1.2 + 0.6 × 1.2, and the *score* of (*Stay Location*$_1$) is evaluated as 0.4 × 1.0 + 0.6 × 1.0. Thus we predict the children of the candidate path with the highest *score* as the answer. If the candidate path with the highest *score* has no children, we predict the children of the candidate path with the second highest score, and so on.

**Table 6. An example for transforming a candidate path set**

| Semantic Candidate Seq. | *SemanticScore* |
|---|---|
| (Unknown) → (School) | 0.8 × 0.667 + 0.667 = 1.2 |
| (School) | 1.0 |

# 6. EXPERIMENTS

In this section, we conduct a series of experiments to evaluate the performance for the proposed location prediction technique using the MIT reality mining dataset [3]. 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 experimental results followed by discussions.

## 6.1 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 of continuous daily activities. The dataset contains cell trajectories as shown in Figure 15. As shown, the stay time in a cell can be derived by calculating the difference in timestamp when a user arrives in and leaves the cell. Thus, we can easily discover the stay cells of each cell trajectory.

| 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 15. An example of cell sequences of a mobile user.**

Since this dataset contains user annotated cell names, they inherently are semantic trajectories as shown in Figure 16. However, the annotation terms are very diverse. For example, one may annotate a cell as "ML" while someone else may annotate 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 terms to find suitable semantic labels near them. Although we make a lot of efforts to figure out the semantics of the annotation terms in the log, there are unfortunately still some terms which we can not be sure of their meanings. As a consequence, we stem such term as "Unknown".

| oid | name | person_oid | celltower_oid |
|---|---|---|---|
| 643 | ML | 29 | 3393 |
| 644 | Office | 29 | 19290 |

**Figure 16. An example of the annotation of cells by a user.**

Among the 106 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 89 mobile users are used in our experiments. For each mobile user, we randomly select 80% of his/her cell trajectories as the training dataset. The remaining trajectories form the testing dataset. Then, we use the training dataset to obtain 1) *semantic trajectory pattern tree* for each mobile user, and 2) *stay location pattern tree* for each user cluster. Finally, we use Equation (2) to evaluate the score of next location of each trajectory in testing dataset based on their *semantic trajectory pattern tree* and *stay location pattern tree*.

## 6.2 Evaluation Methodology

The followings are the main measurements for the experimental evaluation. The Precision, Recall, and F-measure are defined as Equations (3), (4), and (5), where $p^+$ and $p^-$ indicate the number of correct predictions and incorrect predictions, respectively, and $|R|$ indicates the total number of trajectories. In addition, we use the average improvement rate to measure the percentage our proposed method outperforms other methods. The average improvement rate is defined as (6), where $m_{ours}$ and $m_{baseline}$ are the measured

result of our proposed method and that of the compared baseline method, respectively.

$$\text{Precision} = \frac{p^+}{p^+ + p^-} \quad (3)$$

$$\text{Recall} = \frac{p^+ + p^-}{|R|} \quad (4)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$\text{Average improvement rate} = \frac{m_{ours} - m_{baseline}}{m_{baseline}} \quad (6)$$

The experiments are divided into two parts: i) sensitivity tests; and ii) framework evaluation. The sensitivity tests evaluates the proposed techniques within the *SemanPredict* framework under various parameter settings (i.e., $\alpha$ and $\beta$). In this research study, we claim that 1) the semantic information is a critical factor for location prediction, and 2) the partial-matching strategy could improve the recall of prediction. Hence, we obtain two prediction strategies as baselines for comparison with our SemanPredict framework. First, we adapt the *SemanPredict* framework by skipping the semantic mining step to generate one baseline, called Geographic Only (GO), which logically represents the conventional general-based prediction methods. The other baseline, called full-matching (FM), is generated by using traditional matching method instead of partial-matching. Beside, we provide an efficiency evaluation.

## 6.3 Sensitivity Tests

The sensitivity tests evaluate our approach under various parameter settings in terms of Precision. As shown in Figure 17, the Precision of our method is improved when $\alpha$ is increased, i.e., higher precision is achieved when we give more weight on recent mobile moves. It validates our assumption that more recent mobile moves potentially have a greater effect on predicting the next move. However, we also observe that the improvement is not significant since we adopt a partial matching strategy. As a result, the outdated mobile move may be rarely matched with a pattern. We also can observe that the Precision deteriorates as $\beta$ increases, i.e., as more weight is assigned to *semanticScore,* the precision gets lower. This contradicts our assumption that the semantic information improves the prediction precision. We believe that this is because the clusters of users are generated based on semantic trajectory. As a result, the geographic features become more discriminative than the semantic features for a cluster of users with similar semantic behaviors.

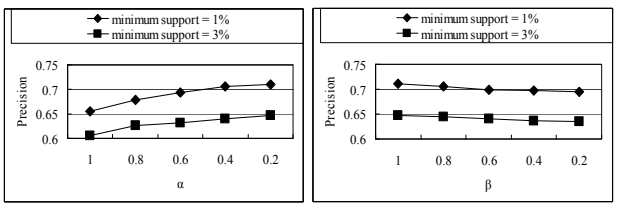



**Figure 17. Precision in various parameter settings.**

Then, we evaluate the impact of the semantic clustering on our prediction model. In Figure 18, we observe that our approach outperforms none-clustering approach in terms of the Precision, Recall, and F-Measure. The average improvement rate of our approach over the none-clustering approach is 20.09% for the precision, 24.06% for the recall, and 21.82% for the F-measure, respectively. It demonstrates that the semantic clustering strategy is effective in improving the proposed prediction framework. We also can observe that our approach is more stable than the none-clustering approach, because the clustering step groups similar users such that most patterns we discovered for each cluster do not fluctuate.

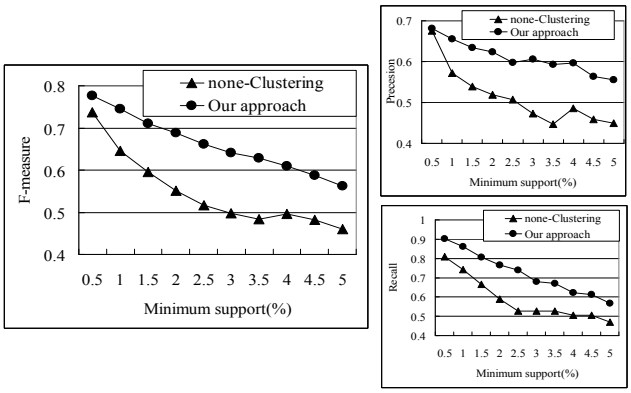



**Figure 18. Impact of the semantic clustering .**

## 6.4 Comparison of Prediction Strategies

This experiment analyzes the precision, recall and F-measure of examined prediction techniques, including Geographic Only (GO), Full-Matching (FM), and our approach (*SemanPredict*). Figure 19 show that *SemanPredict* is not better than FM in terms of precision, but significantly outperforms it in terms of recall and F-measure because *SemanPredict* uses the partial matching strategy. It also leads *SemanPredict* to predict some user moves which are not predictable by other techniques. On the contrary, FM predicts a mobile user's move only if his recent trajectory is a subsequence of the prefix of some patterns. The average improvement rates of *SemanPredict* over FM are 227.72% for the recall, and 96.48% for the F-measure, respectively.

We also can observe that *SemanPredict* is slightly better than GO in terms of precision, but significantly outperforms it in terms of recall and F-measure, because *SemanPredict* considers not only the semantic clustering but also the semantic score of the next location of users' moves in the location prediction. Since the clustering step is based on users' semantic similarities, the recall of *SemanPredict* can achieve 90%. The average improvement rates of *SemanPredict* over GO are 20.02% for the precision, 24.07% for the recall, and 21.78% for the F-measure, respectively.

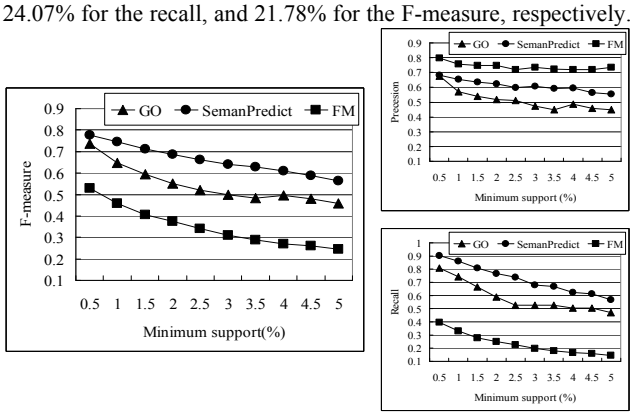
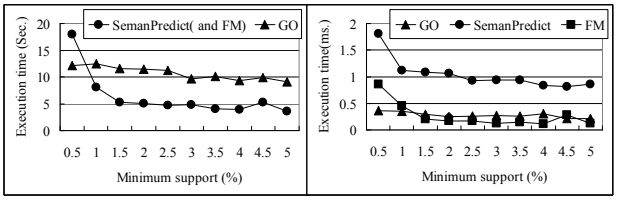



**Figure 19. Comparison of various prediction strategies.**

## 6.5 Efficiency Evaluation

We also conduct experiments to evaluate the efficiency of our approach and other prediction strategies under various minimum supports in offline training module and online prediction module, respectively. Figure 20(a) shows the execution time spent for training our prediction model and GO. As shown, the execution time of our approach consistently outperforms GO under most settings of minimum support. Although our approach needs to deal with the semantic mining in offline training module, the semantic clustering has grouped all users' trajectory logs into small sets. Hence, the execution time for geographic mining in our approach is significantly less than that in GO. We also observe that the execution time of prediction in our approach is longer than that in GO and FM. The reason is that our approach needs to calculate the *SemanticScore* of each candidate path, but GO does not have this overhead. Although FM also needs to calculate the *SemanticScore*, full-matching inherently leads to less candidate paths. However, it is reasonable that the execution time of our approach is limited by 2 mini seconds in online prediction module.




**(a) training** **(b) prediction**

**Figure 20. Execution time.**

# 7. CONCLUSIONS

In this paper, we propose a novel framework, by exploring the semantic trajectories of mobile users, to predict the next location of a mobile user in support of various location-based services. The core of our framework is a novel prediction strategy which evaluates the score of next stay location for a given mobile user. In the *SemanPredict* framework, we propose a novel cluster-based prediction technique to predict the next location of a mobile user. To our best knowledge, this is the first work that exploits both semantic and geographic information in trajectories for location prediction. Through a series of experiments, we validate our proposal and show that the proposed location prediction framework has excellent performance under various conditions.

As for the future work, we plan to design more advanced prediction strategies to enhance the quality of location predictions in location-based services.

# 8. ACKNOWLEDGMENTS

This research was supported by National Science Council, Taiwan, R.O.C. under grant no. NSC100-2631-H-006-002 and NSC100-2218-E-006-001.

# 9. REFERENCES

[1] 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.

[2] 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

[3] 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.

[4] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. ICDE 2008: 70-79.

[5] J. Liu, O. Wolfson, H. Yin. Extracting Semantic Location from Outdoor Positioning Systems. MDM, 2006

[6] E. H.-C. Lu and V. S. Tseng. Mining Cluster-Based Mobile Sequential Patterns in Location-Based Service Environments. MDM, 2009.

[7] M. Morzy. Prediction of moving object location based on frequent trajectories. ISCIS, volume 4263 of LNCS, pages 583–592. Springer, 2006.

[8] M. Morzy. Mining frequent trajectories of moving objects for location prediction. MLDM, volume 4571 of LNCS, pages 667–680. Springer, 2007.

[9] A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti. WhereNext: a location predictor on trajectory pattern mining. KDD 2009: 637-646.

[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] G. Yavas, D. Katsaros, Ö. Ulusoy, and Y. Manolopoulos. A data mining approach for location prediction in mobile environments. D.K.E., 54(2):121–146, 2005.

[12] Yang Ye*, Yu Zheng, Yukun Chen, Jianhua Feng, Xing Xie. Mining Individual Life Pattern Based on Location History. In proceedings of the International Conference on Mobile Data Management 2009 (MDM 2009). IEEE, 1-10.

[13] S.-J. Yen, Y.-S. Lee, C.-H. Lin and J.-C. Ying, Investigating the Effect of Sampling Methods for Imbalanced Data Distributions, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC'2006), pp. 4163-1468, October 2006.

[14] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, V. S. Tseng. Mining User Similarity from Semantic Trajectories. In Proceedings of ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN' 10), San Jose, California, USA, November 2, 2010.

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

[16] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma. Mining interesting locations and travel sequences from GPS trajectories. WWW, 2009.

[17] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma. Mining Correlation Between Locations Using Human Location History. ACM SIGSPATIAL GIS, 2009.