Exploring Regression for Mining User Moving Patterns in a Mobile Computing System

Chih-Chieh Hung, Wen-Chih Peng*, and Jiun-Long Huang

Department of Computer Science, National Chiao Tung University,
Hsinchu, Taiwan, ROC
{hungcc, wcpeng, jlhuang}@csie.nctu.edu.tw

Abstract. In this paper, by exploiting the log of call detail records, we present a solution procedure of mining user moving patterns in a mobile computing system. Specifically, we propose algorithm LS to accurately determine similar moving sequences from the log of call detail records so as to obtain moving behaviors of users. By exploring the feature of spatial-temporal locality, we develop algorithm TC to group call detail records into clusters. In light of the concept of regression, we devise algorithm MF to derive moving functions of moving behaviors. Performance of the proposed solution procedure is analyzed and sensitivity analysis on several design parameters is conducted. It is shown by our simulation results that user moving patterns obtained by our solution procedure are of very high quality and in fact very close to real user moving behaviors.

1 Introduction

User moving patterns refer to the areas where users frequently travel in a mobile computing environment. It is worth mentioning that user moving patterns are particularly important and are able to provide many benefits in mobile applications. A significant amount of research efforts has been elaborated upon issues of utilizing user moving patterns in developing location tracking schemes and data allocation methods [3][5][6]. Clearly, it has been recognized as an important issue to develop algorithms to mine user moving patterns so as to improve the performance of mobile computing systems.

The study in [5] explored the problem of mining user moving patterns with the moving log of mobile users given. Specifically, in order to capture user moving patterns, a moving log recording each movement of mobile users is needed. In practice, generating the moving log of all mobile users unavoidably leads to the increased storage cost and degraded performance of mobile computing systems. Consequently, in this paper, we address the problem of mining user moving patterns from the existing log of call detail records (referred to as CDR) of mobile computing systems. Generally, mobile computing systems generate one call detail record when a mobile user makes or receives a phone call. Table 1 shows an example of selected real call detail records where Uid is the identification of an individual user that makes or receives a phone call and Cellid indicates the corresponding base station that serves that mobile user. Thus, a mobile computing system produces daily a large amount of call detail records which contain hidden valuable information about the moving behaviors of mobile users. Unlike the

* The corresponding author of this paper.

Table 1. An example of selected call detail records

Uid	Date	Time	Cellid
1	01/03/2004	03:30:21	A
1	01/03/2004	09:12:02	D
1	01/03/2004	20:30:21	G
1	01/03/2004	21:50:31	I

moving log keeping track of the entire moving paths, the log of call detail records only reflects the fragmented moving behaviors of mobile users. However, such fragmented moving behaviors are of little interest in a mobile computing environment where one would naturally like to know the complete moving behaviors of users. Thus, in this paper, with these fragmented moving behaviors hidden in the log of call detail records, we devise a solution procedure to mine user moving patterns. The problem we shall study can be best understood by the illustrative example in Fig. 1 where the log of call detail records is given in Table 1. The dotted line in Fig. 1 represents the real moving path of the mobile user and the cells with the symbol of a mobile phone are the areas where the mobile user made or received phone calls. Explicitly, there are four call detail records generated in the log of CDRs while the mobile user travels. Given these fragmented moving behaviors, we explore the technique of regression analysis to generate user moving patterns (i.e., the solid line in Fig. 1). If user moving patterns devised are close to the real moving paths, one can utilize user moving patterns to predict the real moving behaviors of mobile users.

In this paper, we propose a solution procedure to mine user moving patterns from call detail records. Specifically, we shall first determine similar moving sequences from the log of call detail records and then these similar moving sequences are merged into one moving sequence (referred to as *aggregate moving sequence*). It is worth mentioning that to fully explore the feature of periodicity and utilize the limited amount of call detail records, algorithm LS (standing for Large Sequence) devised is able to accurately extract those similar moving sequences in the sense that those similar moving sequences are determined by two adjustable threshold values when deriving the aggregate moving sequence. By exploring the feature of spatial-temporal locality, which refers to the feature that if the time interval between the two consecutive calls of a mobile user is

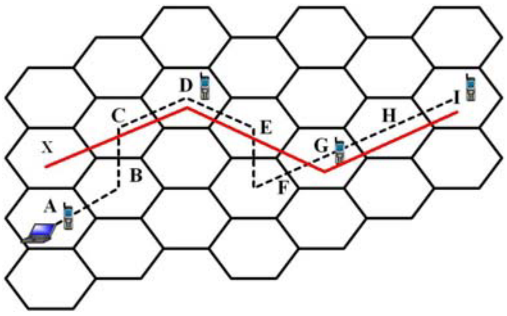

Fig. 1. A moving path and an approximate user moving pattern of a mobile user

small, the mobile user is likely to move nearby, algorithm TC (standing for Time Clustering) developed should group those call detail records into a cluster. For each cluster of call detail records, algorithm MF (standing for Moving Function), a regression-based method, devised is employed to derive moving functions of users so as to generate approximate user moving patterns. Performance of the proposed solution procedure is analyzed and sensitivity analysis on several design parameters is conducted. It is shown by our simulation results that approximate user moving patterns obtained by our proposed algorithms are of very high quality and in fact very close to real moving behaviors of users.

The rest of the paper is organized as follows. Some preliminaries and definitions are presented in Section 2. Algorithms for mining moving patterns are devised in Section 3. Performance results are presented in Section 4. This paper concludes with Section 5.

2 Preliminary

In this paper, assume that the moving behavior of mobile users has periodicity and the consecutive movement of the mobile user is not too far. Therefore, if the time interval of two consecutive CDRs is not too large, the mobile user is likely to move nearby. To facilitate the presentation of this paper, a *moving section* is defined as a basic time unit. A *moving record* is a data structure that is able to accumulate the numbers of base station identifications (henceforth referred to as *item*) appearing in call detail records whose occurring times are within the same moving section. Given a log of call detail records, we will first convert these CDR data into multiple moving sequences where a moving sequence is an ordered list of moving records and the length of the moving sequence is ε. The value of ε depends on the periodicity of mobile users and is able to obtain by the method proposed in [1]. As a result, a moving sequence i is denoted by $<MR_i^1, MR_i^2, MR_i^3, ..., MR_i^{\varepsilon}>$, where MR_i^j is the jth moving record of moving sequence i. We consider four hours as one basic unit of a moving section and the value of ε is six. Given the log data in Table 1, we have the moving sequence $MS_1 = < \{A : 1\}, \{\}, \{D : 1\}, \{\}, \{\}, \{G : 1, I : 1\} >$. *Time projection sequence* of moving sequence MS_i is denoted as TP_{MS_i}, which is formulated as $TP_{MS_i} = < \alpha_1, ..., \alpha_n >$, where $MR_i^{\alpha_j} \neq \{\}$ and $\alpha_1 < ... < \alpha_n$. Explicitly, TP_{MS_i} is a sequence of numbers that are the identifications of moving sections in which the corresponding moving records are not empty. Given $MS_1 =< \{A : 1\}, \{\}, \{D : 1\}, \{\}, \{\}, \{G : 1, I : 1\} >$, one can verify that $TP_{MS_1} =< 1, 3, 6 >$. By utilizing the technique of sequential clustering, a time projection sequence TP_{MS_i} is divided into several groups in which time intervals among moving sections are close. For the brevity purpose, we define a clustered time projection sequence of TP_{MS_i}, denoted by $CTP(TP_{MS_i})$, which is represented as $< CL_1, CL_2, ..., CL_x >$ where CL_i is the ith group and $i = [1, x]$. Note that the value of x is determined by our proposed method.

3 Mining User Moving Patterns

The overall procedure for mining moving patterns comprises three phases, i.e., data collection phase, time clustering phase and regression phase. The details of algorithms in each phases are described in the following subsections.

3.1 Data Collection Phase

As mentioned early, in this phase, we shall identify similar moving sequences from a set of w moving sequences obtained and then merge these similar moving sequences into one *aggregate moving sequence* (to be referred to as AMS). Algorithm LS is applied to moving sequences of each mobile user to determine the aggregate moving sequence that comprises a sequence of large moving records denoted as LMR^i, where $i = [1, \varepsilon]$. Specifically, large moving record LMR^j is a set of items with their corresponding counting values if there are a sufficient number of MR_i^j of moving sequences containing these items. Such a threshold number is called $vertical\_min\_sup$ in this paper. Once the aggregate moving sequence is generated from these recent w moving sequences, we will then compare this aggregate moving sequence with these w moving sequences so as to further accumulate the occurring counts of items appearing in each large moving record. The threshold to identify the similarity between moving sequences and the aggregate moving sequence is named by $match\_min\_sup$. The algorithmic form is given below.

Algorithm LS

```
input: $w$ moving sequences with their lengths being $\varepsilon$, two threshold:$vertical\_min\_sup$
and $match\_min\_sup$
output: Aggregate moving sequence $AMS$
1 begin
2  for $j$ =1 to $\varepsilon$
3    for i=1 to $w$
4      $LMR^j$ =large 1-itemset of $MR_i^j$;
           (by $vertical\_min\_sup$)
5  for $i = 1$ to $w$
6    begin
7      $match = 0$;
8      for $j = 1$ to $\varepsilon$
9        begin
10         $C(MR_i^j, LMR^j) =$
             $|x \in MR_i^j \cap LMR^j| / |y \in MR_i^j \cup LMR^j|$;
11         $match = match + |MR_i^j| * C(MR_i^j, LMR^j)$;
12     end
13     if $match \geq match\_min\_sup$ then
14       accumulate the occurring counts of
         items in the aggregate moving sequence;
15   end
16 end
```

In algorithm LS (from line 2 to line 4), we first calculate the appearing counts of items in each moving sections of w moving sequences. If the count of an item among w moving sequences is larger than the value of $vertical\_min\_sup$, this item will be weaved into the corresponding large moving record. After obtaining all large moving records, AMS is then generated and is represented as $< LMR^1, LMR^2, ...,$

LMR^{ε} $>$, where the length of the aggregate moving sequence is ε. As mentioned before, large moving records contain frequent items with their corresponding counts. Once obtaining the aggregate moving sequence, we should in algorithm LS (from line 5 to line 12) compare this aggregate moving sequence with w moving sequences in order to identify those similar moving sequences and then calculate the count of each item in each large moving record. Note that a moving sequence (respectively, AMS) consists of a sequence of moving records (respectively, large moving records). Thus, in order to quantity how similar between a moving sequence (e.g., MS_i) and AMS, we shall first measure the closeness between moving record MR_i^j and LMR^j, denoted by $C(MR_i^j, LMR^j)$. $C(MR_i^j, LMR^j)$ is formulated as $\frac{|\{x \in MR_i^j \cap LMR^j\}|}{|\{y \in MR_i^j \cup LMR^j\}|}$ that returns the normalized value in $[0, 1]$. The larger the value of $C(MR_i^j, LMR^j)$ is, the more closely MR_i^j resembles LMR^j. Accordingly, the similarity measure of moving sequence MS_i and AMS is thus formulated as $sim(MS_i, AMS) = \sum_{i=1}^{\varepsilon} |MR_i^j| * C(MR_i^j,\ LMR^j)$. Given a threshold value $match\_min\_sup$, for each moving sequence MS_i, if $sim(MS_i, AMS) \geq match\_min\_sup$, moving sequence MS_i is identified as a similar moving sequence containing sufficient moving behaviors of mobile users. In algorithm LS (from line 13 to line 14), for each item in large moving records, the occurring count is accumulated from the corresponding moving records of those similar moving sequences.

3.2 Time Clustering Phase

In this phase, two threshold values (i.e., δ and σ^2) are given in clustering a time projection sequence. Explicitly, the value of δ is used to determine the density of clusters and σ^2 is utilized to make sure that the spread of the time is bounded within σ^2. Algorithm TC is able to dynamically determine the number of groups in a time projection sequence.

Algorithm TC (from line 2 to line 3) first starts clustering coarsely TP_{AMS} into several marked clusters if the difference between two successive numbers is smaller than the threshold value δ. As pointed out before, CL_i denotes the ith marked cluster. In order to guarantee the quality of clusters, a spread degree of CL_i, denoted as $Sd(CL_i)$,

Algorithm TC

```
input: Time projection sequence $TP_{AMS}$, threshold $\delta$ and $\sigma^2$
output: Clustered time projection sequence $CTP(TP_{AMS})$
1 begin
2   group the numbers whose differences are within $\delta$;
3   mark all clusters;
4   while there exist marked clusters and $\delta \geqq 1$
5     for each marked clusters $CL_i$
6       if $Sd(CL_i) \leq \sigma^2$
7         unmark $CL_i$;
8     $\delta = \delta - 1$;
9     for all marked clusters $CL_i$
```

```
10        group the numbers whose differences are
           within $\delta$ in $CL_i$;
11    end while
12    if there exist marked clusters
13      for each marked cluster $CL_i$
14        $k = 1$;
15        repeat
16          $k$++;
17          divide evenly $CL_i$ into $k$ groups ;
18        until the spread degree of each group$\leq \sigma^2$;
19  end
```

is defined to measure the distribution of numbers in cluster CL_i. Specifically, $Sd(CL_i)$ is modelled by the variance of a sequence of numbers. Hence, $Sd(CL_i)$ is formulated as $\frac{1}{m}\sum_{k=1}^{m}(n_k - \frac{1}{m}\sum_{j=1}^{m} n_j)^2$, where n$_k$ is the kth number in CL_i and m is the number of elements in CL_i. As can be seen from line 5 to line 7 in algorithm TC, for each cluster CL_i, if $Sd(CL_i)$ is smaller than σ^2, we unmark the cluster CL_i. Otherwise, we will decrease δ by 1 and with given the value of δ, algorithm TC (from line 8 to line 10) will re-cluster those numbers in unmark clusters. Algorithm TC partitions the numbers of TP_{AMS} iteratively with the objective of satisfying two threshold values, i.e., δ and σ^2, until there is no marked cluster or $\delta = 0$. If there is no marked clusters, $CTP(TP_{AMS})$ is thus generated. Note that, however, if there are still marked clusters with their spread degree values larger than σ^2, algorithm TC (from line 12 to line 18) will further finely partition these marked clusters so that the spread degree for each marked cluster is constrained by the threshold value of σ^2. If the threshold value of δ is 1, a marked cluster is usually a sequence of continuos numbers in which the spread degree of this marked cluster is still larger than σ^2. Given marked cluster CL_i, algorithm TC initially sets k to be 1. Then, marked cluster CL_i is evenly divided into k groups with each group size $\lceil \frac{n}{k} \rceil$. By increasing the value of k each run, algorithm TC is able to partition the marked cluster until the spread degree of each partition in the marked cluster CL_i satisfies σ^2.

3.3 Regression Phase

Assume that AMS is $< LMR^1, LMR^2, ..., LMR^\varepsilon >$ with its clustered time projection sequence $CTP(TP_{AMS})$ = CL_1, CL_2, ..., CL_k, where CL_i represents the ith cluster. For each cluster CL_i of $CTP(\ TP_{AMS})$, we will derive the estimated moving function of mobile users, expressed as $E_i(t) = (\ \hat{x}_i(t), \hat{y}_i(t),\ valid\_time\_interval)$, where $\hat{x}_i(t)$ (respectively, $\hat{y}_i(t)$) is a moving function in x-coordinate axis (respectively, in y-coordinate axis)) and $valid\_time\_interval$ indicates the time interval when the moving function is valid.

Without loss of generality, let CL_i be $\{t_1, t_2, ..., t_n\}$ where t_i is one of the moving section in CL_i. As described before, a moving record has the set of the items with their corresponding counts. Therefore, we could extract those large moving records from AMS to derive the estimated moving function for each cluster. In order to derive moving functions, the location of base stations should be represented in geometry model through a map table provided by tele-companies. Hence, given AMS and a cluster of $CTP(TP_{AMS})$, for each cluster of $CTP(TP_{AMS})$, we could have geometric coordinates of frequent items with their corresponding counts, which are able to represent as (t_1,x_1,y_1,w_1), (t_2,x_2,y_2,w_2), ...(t_n,x_n,y_n,w_n). Accordingly, for each cluster of $CTP(\ TP_{AMS})$, regression analysis is able to derive the corresponding estimated moving function.

Given a cluster of data points (e.g., (t_1, x_1, y_1, w_1), (t_2, x_2, y_2, w_2), ..., (t_n,x_n,y_n,w_n)), we first consider the derivation of $\hat{x}(t)$. If the number of distinct time points in a given cluster is $m + 1$, a m-degree polynomial function $\hat{x}(t) = a_0 + a_1 t + ... + a_m t^m$ will be derived to approximate moving behaviors in x-coordinate axis. Specifically, the regression coefficients $\{\alpha_0, \alpha_1, ...a_m\}$ are chosen to make the residual sum of squares $\epsilon_x = \sum_{i=1}^{n} w_i e_i^2$ minimal, where w_i is the weight of the data point (x_i, y_i) and $e_i = (x_i - (a_0 + a_1 t_i + a_2 (t_i)^2 ... + a_m (t_i)^m))$. To facilitate the presentation of our paper, we define the following terms:

$$T = \begin{bmatrix} 1 \ t_1 \ (t_1)^2 \ ... \ (t_1)^m \\ 1 \ t_2 \ (t_2)^2 \ ... \ (t_2)^m \\ ... \ ... \ ... \ ... \ ... \\ 1 \ t_n \ (t_n)^2 \ ... \ (t_n)^m \end{bmatrix}, \ a^* = \begin{bmatrix} a_0 \\ a_1 \\ ... \\ a_m \end{bmatrix}, \ \boldsymbol{b}_x = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix}, \ e = \begin{bmatrix} e_1 \\ e_2 \\ ... \\ e_n \end{bmatrix}^T, \ W = \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & ... & \\ & & & w_n \end{bmatrix}.$$

The residual sum of squares can be expressed as $\epsilon_x = e^T W e$. Since w_i are positive for all i, W is written as: $W = \sqrt{W}\sqrt{W}$, where $\sqrt{W}$ is a diagonal matrix with its diagonal entries to be $[\sqrt{w_1}, \sqrt{w_2}, ..., \sqrt{w_n}]$. Thus, $\epsilon_x = e^T W e = (\boldsymbol{b}_x - Ta^*)^T \sqrt{W}\sqrt{W}(\boldsymbol{b}_x - Ta^*) = (\sqrt{W}\boldsymbol{b}_x - \sqrt{W}Ta^*)^T(\sqrt{W}\boldsymbol{b}_x - \sqrt{W}Ta^*)$. Clearly, ϵ_x is minimized w.r.t. a^* by the normal equation $(\sqrt{W}T)^T(\sqrt{W}T)a^* = (\sqrt{W}T)^T\sqrt{W}\boldsymbol{b}_x$[2]. The coefficients $\{\alpha_0, \alpha_1, ...a_m\}$ can hence be obtained by solving the normal equation: $a^* = [(\sqrt{W}T)^T(\sqrt{W}T)]^{-1}\ (\sqrt{W}T)^T\sqrt{W}\boldsymbol{b}_x$. Therefore, $\hat{x}(t) = a_0 + a_1 t + ... + a_m t^m$ is obtained. Following the same procedure, we could derive $\hat{y}(t)$. As a result, for each cluster of $CTP(TP_{AMS})$, the estimated moving function $E_i(t) = (\hat{x}(t), \hat{y}(t), [t_1, t_n])$ of a mobile user is devised.

Algorithm MF

```
input: $AMS$ and clustered time projection sequence $CTP(TP_{AMS})$
output: A set of moving functions $F(t) = \{E_1(t), U_1(t), E_2(t), ..., E_k(t), U_k(t)\}$
1 begin
2  initialize $F(t)$=empty;
3   for i= 1 to k-1
4     begin
5       doing regression on $CL_i$ to generate $E_i(t)$;
6       doing regression on $CL_{i+1}$ to generate $E_{i+1}(t)$;
7       $t_1$ =the last number in $CL_i$;
```

```
8      $t_2$ =the first number in $CL_{i+1}$;
9       using inner interpolation to generate
          $U_i(t) = (\hat{x}_i(t), \hat{y}_i(t), (t_1, t_2))$;
10      insert $E_i(t), U_i(t)$ and $E_{i+1}(t)$ in $F(t)$;
11     end
12   if($1 \notin CL_1$)
13     generate $U_0(t)$ and Insert $U_0(t)$ into the head of $F(t)$;
14   if($\varepsilon \notin CL_k$)
15     generate $U_k(t)$ and Insert $U_k(t)$ into the tail of $F(t)$;
16  return $F(t)$;
17 end
```

4 Performance Study

In this section, the effectiveness of mining user moving patterns by call detail records is evaluated empirically. The simulation model for the mobile system considered is described in Section 4.1. Section 4.2 is devoted to experimental results and comparison with the original algorithm of mining moving patterns [5].

4.1 Simulation Model for a Mobile System

To simulate the base stations in a mobile computing system, we use a eight by eight mesh network, where each node represents one base station and there are hence 64 base stations in this model [4][5]. A moving path is a sequence of base stations travelled by a mobile user. The number of movements made by a mobile user during one moving section is modeled as a uniform distribution between mf-2 and mf+2. Explicitly, the larger the value of mf is, the more frequently a mobile user moves. To model user calling behavior, the calling frequency is employed to determine the number of calls during one moving section. If the value of cf is large, the number of calls for a mobile user will increase. Similar to [5], the mobile user moves to one of its neighboring base stations depending on a probabilistic model. To make sure the periodicity of moving behaviors, the probability that a mobile user moves to the base station where this user came from is modeled by P$_{back}$ and the probability that the mobile user routes to the other base stations is determined by (1-P$_{back}$)/(n-1) where n is the number of possible base stations this mobile user can move to. The method of mining moving patterns in [5], denoted as UMP, is implemented for the comparison purposes. For interest of brevity, our proposed solution procedure of mining user moving patterns is expressed by $AUMP$ (standing for approximate user moving patterns). The location is represented as the identification of a base station. To measure the accuracy of user moving patterns, we use the hop count (denoted as hn), which is measured by the number of base stations, to represent the distance from the location predicted by moving functions derived to the actual location of the mobile user. Intuitively, a smaller value of hn implies that the more accurate prediction is achieved.

4.2 Experiments of UMP and AUMP

To conduct the experiments to evaluate UMP and $AUMP$, we set the value of w to be 10, the value of cf to be 3 and the value of ε to be 12. In order to reduce the

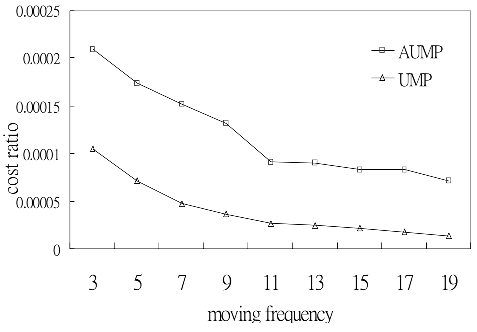

Fig. 2. The cost ratios of AUMP and UMP with the moving frequency varied

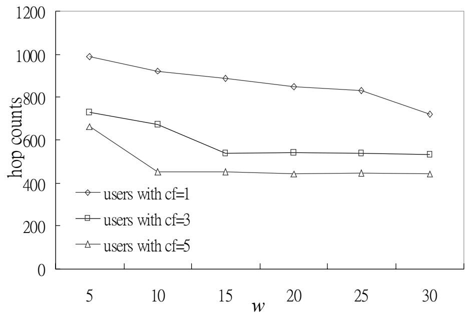

Fig. 3. The performance of AUMP with the value of w varied

amount of data used in mining user moving patterns, $AUMP$ explores the log of call detail records. The cost ratio for a user, i.e., $\frac{\frac{1}{hn}}{amount\ of\ log\ data}$, means the prediction accuracy gained by having the additional amount of log data. Fig. 2 shows the cost ratios of UMP and $AUMP$. Notice that $AUMP$ has larger cost ratios than UMP, showing that $AUMP$ employs the amount of log data more cost-efficiently to increase the prediction accuracy.

The impact of varying the values of w for mining moving patterns is next investigated. Without loss of generality, we set the value of ε to be 12, that of mf to be 3 and the values of cf to be 1, 3 and 5. Both $vertical\_min\_sup$ and $match\_min\_support$ are set to 20% , the value of δ is set to be 3 , and σ^2 is set to be 0.25. With this setting, the experimental results are shown in Fig. 3.

As can be seen from Fig. 3, the hop count of AUMP decreases as the value of w increases. This is due to that as the value of w increases, meaning that the number of moving sequences considered in AUMP increases, AUMP is able to effectively extract more information from the log of call detail records. Note that with a given the value of w, the hop count of AUMP with a larger value of cf is smaller, showing that the log of data has more information when the value of cf increases. Clearly, for mobile users having high call frequencies, the value of w is able to set smaller in order to quickly obtain moving patterns. However, for mobile users having low call frequencies, the value of w should be set larger so as to increase the accuracy of moving patterns mined by $AUMP$.

5 Conclusions

In this paper, without increasing the overhead of generating the moving log, we presented a new mining method to mine user moving patterns from the existing log of call detail records of mobile computing systems. Specifically, we proposed algorithm LS to capture similar moving sequences from the log of call detail records and then these similar moving sequences are merged into the aggregate moving sequence. By exploring the feature of spatial-temporal locality, algorithm TC proposed is able to group call detail records into several clusters. For each cluster of the aggregate moving sequence, algorithm MF devised is employed to derive the estimated moving function, which is able to generate user moving patterns. It is shown by our simulation results that user moving patterns achieved by our proposed algorithms are of very high quality and in fact very close to real user moving behaviors.

Acknowledgment

The authors are supported in part by the National Science Council, Project No. NSC 92-2211-E-009-068 and NSC 93-2213-E-009-121, Taiwan, Republic of China.

References

1. J. Han, G. Dong, and Y. Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. In *Proceeding of the 15th International Conference on Data Engineering*, March 1999.
2. R. V. Hogg and E. A. Tanis. *Probability and Statistical Inference*. Prentice-Hall International Inc., 1997.
3. D. L. Lee, J. Xu, B. Zheng, and W.-C. Lee. Data Management in Location-Dependent Information Services. In *Proceeding of IEEE Pervasive Computing*, pages 65–72, 2002.
4. Y.-B. Lin. Modeling Techniques for Large-Scale PCS Networks. *IEEE Communications Magazine*, 35(2):102–107, February 1997.
5. W.-C. Peng and M.-S. Chen. Developing Data Allocation Schemes by Incremental Mining of User Moving Patterns in a Mobile Computing System. In *Proceeding of IEEE Transactions on Knowledge and Data Engineering, Volume 15*, pages 70–85, 2003.
6. H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke. Personal Paging Area Design Based On Mobile's Moving Behaviors. In *Proceeding of IEEE INFOCOM*, 2001.