Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L12

Mining Group Moving Patterns in a Mobile Environment

Yin-Fu Huang and Li-Kai Lin*

School of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Touliu, Yunlin, 640 Taiwan

Received: 18 Apr. 2013, Revised: 13 Aug. 2013, Accepted: 14 Aug. 2013
Published online: 1 Apr. 2014

Abstract: In recent years, wireless networks and mobile applications grew rapidly. Mobile users not only request various kinds of interesting information, but also demand on the quality of services. In this paper, we use an efficient algorithm called graph search technique (GST) to mine the frequent moving patterns of each mobile user. For found frequent moving patterns (FMPs), we further apply the Apriori algorithm to find the common FMPs among users. Finally, for each group of users, a characterizing method is used to discover the relevant attribute values of the group. In the experiments, we observed that the GST algorithm has better performance in execution time and is more stable than the AprioriAll algorithm. Furthermore, we also observed the number of found patterns under different minimum support values and relevant percentages.

Keywords: Data Mining, Sequential Pattern Mining, Mobile Networks

1 Introduction

In recent years, due to the rapid growth in mobile and wireless networks, more and more people are used to searching the information on mobile devices through the Internet. Because of the popularity of mobile and wireless networks, almost everyone has his/her own mobile device, such as cell phone, PDA, and notebook. Furthermore, lots of users would like to request various kinds of information when they traveled around different places. However, they always receive the irrelevant information, or spend much time on requesting or downloading the information they are interested in. This not only wastes user time, but also consumes network bandwidth. Therefore, some researchers attempt to mine users' moving patterns in order to get better performance [5,6,7,8,9,10]. After getting the frequent moving patterns of users, we can predict their next stops and send useful information and added value services to them in advance, thereby reducing communication time and saving network bandwidth. Since the amounts of data in databases grew very large, data mining has become an interesting area in recent years. Different kinds of knowledge such as association rules, sequential patterns, et al. could be mined from a large database, and also lots of methods were proposed to solve these problems. The association rule mining is to find a set of items frequently occurring together. The most well-known method to find association rules is called Apriori algorithm [1], which is also the basic of the sequential pattern mining. Different from the association rule mining, the sequential pattern mining takes the temporal relationship of patterns into consideration, and is first proposed by Agrawal and Srikant in 1995 [2]. Most of the previous researches in the sequential pattern mining such as AprioriAll [2,5,9], DSG algorithm [11], fuzzy algorithm [4], and mining path traversal patterns algorithm [3], adopted an Apriori-based method to find sequential patterns. Besides, some of other research used the tree structure [7,8,10,12] to find sequential patterns, and showed that they have better performance than the Apriori-based methods. The reason is that the Apriori-based methods scan the whole database frequently and repetitively to generate candidate and large itemsets, whereas the tree structure methods scan the database only once. Another efficient method for mining sequential patterns is the graph search technique [6]. Different from the above methods, the GST algorithm can out of order find large k-sequences (k>=3); in other words, it can find large k-sequences not directly through large (k-1)-sequences. This leads the GST algorithm has much better performance than the above methods. The remainder of the paper is organized as follows. Section 2 presents the related work and some preliminaries. In Section 3, we proposed a method not only finding users' frequent moving patterns, but also characterizing a

* Corresponding author e-mail: huangyf@yuntech.edu.tw

specific group of mobile users with the same frequent moving patterns. Then, experimental results are shown in Section 4. Finally, we make conclusions and future work in Section 5.

2 Related Work

As shown in Figure 1, the typical architecture of a third-generation (3G) mobile radio system is composed of cells each of which has a base-station. Several neighbor base-stations are grouped into a location area, which are managed by a base-station controller (BSC). Beyond the BSC, a mobile switch center (MSC) maintains a visitor location register (VLR) that records the location area identifier and relevant information of current mobile users in the location area. Further, all the mobile users' information in visitor location registers would be sent to a home location register (HLR). When a mobile user moves from one location area to another one, MSC would update his/her information in the corresponding VLR and HLR. The data recorded in the HLR are also called log data, which can be used to mine the frequent moving patterns of each mobile user, and further find the moving patterns of a specific group of mobile users. A moving pattern means a spatial moving pattern of a mobile user that moves sequentially during a time period. In the previous researches, the mined moving patterns indicate that a mobile user moves from one location area to another. However, in our research, the moving patterns are mined finer so that they involve sites, even further site categories. For example, given each city/county in Taiwan as one location area, if a mobile user living Touliu city would go to a movie theater located in Taichung city, the moving sequence could be 1) walking from his home to the railway station, 2) then taking a train to Taichung, and 3) finally taking a bus to the movie theater. For the previous researches, the mined moving pattern of the user could be Touliu→Taichung, whereas the sequence would be home→railway station→movie theater in our research.

Due to privacy issues, the real log data of mobile users, in the HLR, are not easily acquired, so that most previous researches used simulation methods to generate moving patterns. In this paper, we also develop a simply mobile data generator to produce log data. The contents of produced log data consist of user ID, date, and moving sequence, as shown in Table 1. In the data generator, each mobile user is assigned five attributes such as age, gender, marital status, occupation, and hobby. Also, visiting sites are classified into eight categories such as leisure, entertainment, sports, shopping, foods, arts and culture, education, and finance, as shown in Table 2. Besides, to mine the moving patterns with only considering site categories, if a mobile user goes to a department store and then a supermarket, and both sites belong to the same category "shopping", these two successive sites should be regarded as one category "shopping". In other words, the same symbol would not appear at two successive positions in a moving sequence. Finally, for sequential pattern mining problems, the AprioriAll method was the first one proposed by Agrawal and Srikant in 1995, which is derived from association rule mining methods by adding temporal relationships. That is, <A, B> and <B, A> are two different patterns for sequential pattern mining methods, but are the same ones for association rule mining methods. In sequential pattern mining, a sequence with k items (k>0) is called k-sequence, and a sequence with the support count larger than or equal to the minimum support value is called a large sequence. Furthermore, a large sequence is maximal if it is not contained in any other large sequences.

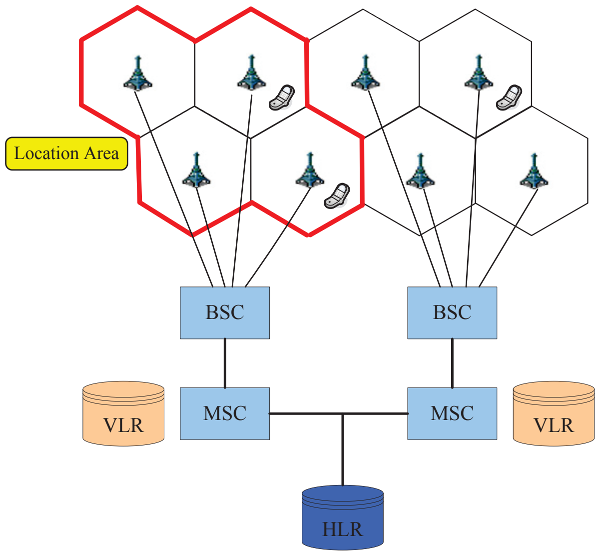

Fig. 1: Typical architecture of a 3G mobile radio system

Table 1: Log data of mobile users

User ID	Date	Moving Sequence
1	2010/01/01	B, D, G, B, C, E
1	2010/01/02	B, G, C, D, C
1	2010/01/10	G, D, B, C, G, B, A, C
:	:	:

Table 2: Corresponding symbols for site categories

A	Leisure
B	Entertainment
C	Sports
D	Shopping
E	Foods
F	Arts and Culture
G	Education
H	Finance

The graph search technique algorithm [6] used in this paper is different from the Apriori-like algorithms, and its finding large k-sequences (k>=3) can be out of order. First, it produces a set of candidate 1-sequences (C_1) and

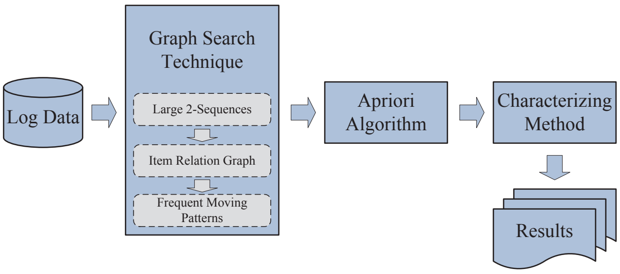

Fig. 2: Architecture of the proposed method

their support counts. If the support count of a 1-sequence in C_1 is larger than or equal to the minimum support value, the 1-sequence is put into a set of large 1-sequences (L_1). Then, it produces a set of candidate 2-sequences (C_2) by self-joining L_1, and their support counts. Following the same procedure, a set of large 2-sequences (L_2) is obtained to construct the item relation graph. Finally, through searching the graph, all the other large k-sequences (k>=3) are found, and also the maximal large sequences.

3 Mining Method

In this section, we describe the proposed method used to mine the moving patterns for a specific group of mobile users. For the architecture of the proposed method as shown in Figure 2, first, the graph search technique (GST) algorithm is used to find the frequent moving patterns of each mobile user. Then, we use the Apriori algorithm to further find the common frequent moving patterns among users. Finally, we apply a characterizing method to discover the relevant attribute values corresponding to a specific group of mobile users.

3.1 Graph Search Technique

The finding large k-sequences (k>=3) in the GST algorithm can be out of order. First, it produces a set of large 2-sequences (L_2) and uses them to construct the item relation graph (IRG). Then, through searching the graph, all the frequent moving patterns can be found.

3.1.1 Scanning the Database and Generating the Large 1-sequences (L_1)

First, we extract the log data of each mobile user, which can be regarded as one database for mining frequent moving patterns, as shown in Table 3. Then, we scan the database to obtain the candidate 1-sequences (C_1) and their support counts. For each 1-sequence in C_1, if its

Table 3: Log data for user 1

User ID	Date	Moving Sequence
1	2011/01/01	B, C, B, F, G
1	2011/01/02	G, A, B, C, G, B, D, C
1	2011/01/05	C, F, E, D, B, C, A
1	2011/01/10	G, A, C, H, B
1	2011/01/15	B, A, G, B, C, E
1	2011/01/16	A, G, D, C, B
1	2011/01/21	B, G, C, B, A, C
1	2011/01/22	E, G, B, A, G, C, D
1	2011/01/23	G, B, C, B, G
1	2011/01/27	C, E, B, F, B, G, H

Table 4: Candidate 1-sequences (C_1)

Sequence	Support
A	7
B	10
C	10
D	4
E	4
F	3
G	9
H	2

Table 5: Large 1-sequences (L_1)

Seq	Date	Pos
A	2011/01/02	2
A	2011/01/05	7
A	2011/01/10	2
A	2011/01/15	2
A	2011/01/16	1
A	2011/01/21	5
A	2011/01/22	4

Seq	Date	Pos
B	2011/01/01	1, 3
B	2011/01/02	3, 6
B	2011/01/05	5
B	2011/01/10	5
B	2011/01/15	1, 4
B	2011/01/16	5
B	2011/01/21	1, 4
B	2011/01/22	3
B	2011/01/23	2, 4
B	2011/01/27	3, 5

Seq	Date	Pos
C	2011/01/01	2
C	2011/01/02	4, 8
C	2011/01/05	1, 6
C	2011/01/10	3
C	2011/01/15	5
C	2011/01/16	4
C	2011/01/21	3, 6
C	2011/01/22	6
C	2011/01/23	3
C	2011/01/27	1

Seq	Date	Pos
G	2011/01/01	5
G	2011/01/02	1, 5
G	2011/01/10	1
G	2011/01/15	3
G	2011/01/16	2
G	2011/01/21	2
G	2011/01/22	2, 5
G	2011/01/23	1, 5
G	2011/01/27	6

support count is larger than or equal to the minimum support value, it would be put into a set of large 1-sequences (L_1). Given the example as shown in Table 3 and the minimum support value 5, C_1 and L_1 can be obtained as shown in Table 4 and Table 5, respectively. For each L_1, we record the dates of its associated moving sequences and the relative positions in these moving sequences.

3.1.2 Generating the Large 2-sequences (L_2) by Self-joining L_1

Then, we produce the candidate 2-sequences (C_2) by self-joining L_1 with dates and positions. For two large

Table 6: Large 2-sequences (L_2)

Seq	Date	Pos(1, r)
$< A,C >$	2011/01/02	(2, 4) (2, 8)
$< A,C >$	2011/01/10	(2, 3)
$< A,C >$	2011/01/15	(2, 5)
$< A,C >$	2011/01/16	(1, 4)
$< A,C >$	2011/01/21	(5, 6)
$< A,C >$	2011/01/22	(4, 6)

Seq	Date	Pos(1, r)
$< B,C >$	2011/01/01	(1, 2)
$< B,C >$	2011/01/02	(3, 4) (3, 8) (6, 8)
$< B,C >$	2011/01/05	(5, 6)
$< B,C >$	2011/01/15	(1, 5) (4, 5)
$< B,C >$	2011/01/21	(1, 3) (1, 6) (4, 6)
$< B,C >$	2011/01/22	(3, 6)
$< B,C >$	2011/01/23	(2, 3)

Seq	Date	Pos(1, r)
$< B,G >$	2011/01/01	(1, 5) (3, 5)
$< B,G >$	2011/01/02	(3, 5)
$< B,G >$	2011/01/15	(1, 3)
$< B,G >$	2011/01/21	(1, 2)
$< B,G >$	2011/01/22	(3, 5)
$< B,G >$	2011/01/23	(2, 5) (4, 5)
$< B,G >$	2011/01/27	(3, 6) (5, 6)

Seq	Date	Pos(1, r)
$< C,B >$	2011/01/01	(2, 3)
$< C,B >$	2011/01/02	(4, 6)
$< C,B >$	2011/01/05	(1, 5)
$< C,B >$	2011/01/10	(3, 5)
$< C,B >$	2011/01/16	(4, 5)
$< C,B >$	2011/01/21	(3, 4)
$< C,B >$	2011/01/23	(3, 4)
$< C,B >$	2011/01/27	(1, 3) (1, 5)

Seq	Date	Pos(1, r)
$< G,B >$	2011/01/02	(1, 3) (1,6) (5, 6)
$< G,B >$	2011/01/10	(1, 5)
$< G,B >$	2011/01/15	(3, 4)
$< G,B >$	2011/01/16	(2, 5)
$< G,B >$	2011/01/21	(2, 4)
$< G,B >$	2011/01/22	(2, 3)
$< G,B >$	2011/01/23	(1, 2) (1, 4)

Seq	Date	Pos(1, r)
$< G,C >$	2011/01/02	(1, 4) (1, 8) (5, 8)
$< G,C >$	2011/01/10	(1, 3)
$< G,C >$	2011/01/15	(3, 5)
$< G,C >$	2011/01/16	(2, 4)
$< G,C >$	2011/01/21	(2, 3) (2, 6)
$< G,C >$	2011/01/22	(2, 6)
$< G,C >$	2011/01/23	(1, 3)

1-sequences p and q, if p.Date is equal to q.Date and p.Position is less than q.Position, they would form a candidate 2-sequence. For the sequential pattern mining, 2-sequences <A, B> and <B, A> are different and should be put in C_2. Also, the 2-sequences in C_2 with the support counts larger than or equal to the minimum support value 5 would be put into a set of large 2-sequences (L_2), as shown in Table 6.

3.1.3 Constructing the Item Relation Graph (IRG)

Next, we use the large 2-sequences in L_2 to construct the IRG. In the IRG, each node represents an item of a 2-sequence, and each directional edge indicates the order of a 2-sequence. Besides, an edge is labeled with some information such as Date and Position recorded in L_2. Based on the large 2-sequences in L_2 as shown in Table 6, the IRG can be constructed as shown in Figure 3 where all labels in edges are omitted.

3.1.4 Searching the IRG to Obtain Maximal Large Sequences

Finally, we search all the other large k-sequences (k>=3) rooted from each node in the IRG, and then find the maximal large sequences. While traversing the IRG from one edge to the next edge, some conditions must be met. For edge e and the next edge w, if e.Date is equal to w.Date and e.Position(r) is equal to w.Position(l), a candidate 3-sequence is found. Then, traversing the IRG continues till a longest path is found. For the IRG as shown in Figure 3, the maximal large k-sequences (k>=3) <G, B, C> and <G, C, B> are found as shown in Table 7.

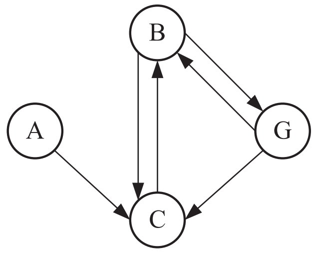

Fig. 3: Item relation graph

Table 7: Maximal large sequences

Sequence
$< G,B,C >$
$< G,C,B >$

Table 8: Frequent moving patterns of each user

User ID	Frequent Moving Patterns
1	$< G,B,C >, < G,C,B >$
2	$< H,A,B >$
3	$< A,C,B >, < G,B,C >, < B,G,C,B >$
:	:
999	$< E,D,F >, < D,C,E,F >$
1000	$< C,A,E >, < C,B,E,A >$

Table 9: Grouping members with common FMPs

Common FMP	Member ID
$< G,B,C >$	1, 3, 10, ...
$< A,C,B >$	3, 17, ...
$< B,G,C,B >$	1, 3, 8, ...
:	:
$< A,C,B >, < B,G,C,B >$	3, ...
:	:

3.2 Apriori Algorithm

After mining the frequent moving patterns (FMP) of each mobile user, we collect them in a new dataset as shown in Table 8. In this step, we regard each FMP as one item and further apply the Apriori algorithm to find the common FMPs among users, based on another minimum support value. That is, if the support count of an FMP is larger than or equal to the minimum support value, it means that there are a significant number of users with a common FMP. For the example as shown in Table 9, each row lists a specific group of users with common FMPs.

Table 10: Attributes of users with common FMP $< G,B,C >$

User ID	Gender	Age	Marital status	Occupation	Hobby
1	Male	20~29	Not	Students	{1,4,6}
3	Male	20~29	Not	Tech. and Info.	{2,3,8}
10	Male	20~29	Not	Students	{3,7,9}
51	Male	20~29	Not	Students	{2,5,11}
122	Male	20~29	Not	Service Industry	{1,3,12}
175	Female	30~39	Not	Others	{1,2,9}
301	Male	40~49	Not	Service Industry	{1,4,12}
558	Male	20~29	Not	Entertainment	{2,8,9}
610	Female	20~29	Not	Students	{4,8,12}
727	Male	20~29	Married	Manufacturing Industry	{2,3,10}
:	:	:	:	:	:

Table 11: Results of the proposed method

FMP	Attributes
$< G,B,C >$	Male, 20~29 years old, not married, students, traveling and playing ball
$< G,D,E >$	Female, under 19 years old, not married, students, reading books
$< A,C,B >$ $< B,G,C,B >$	Male, 20~29 years old, not married, manufacturing industry, traveling
:	:

3.3 Characterizing Method

In this step, we apply a characterizing method to discover the relevant attribute values corresponding to a specific group of mobile users. For the example with common FMP <G, B, C>, all attribute values of the specific group of users are shown in Table 10. In the characterizing method, we can discover what attribute values are to represent a specific group of users. For each attribute, if a value appears higher than the average value, this value is marked as a candidate value. Thus, there could be more than one candidate value for each attribute. After finding the candidate attributes of five attributes, we combine them into several attribute-combined values. For a specific group of users with a common FMP, if an attribute-combined value appears higher than the relevant percentage, this attribute-combined value can be viewed as a relevant attribute-combined value of this group. For the example as shown in Table 11, the users who are males, between 20 29 years old, not married, students, and with the hobby "traveling" and "playing ball" would probably have the common frequent moving pattern <G, B, C>.

4 Experimental Results

4.1 Simulation Environment and Data Sets

We implemented the proposed method using C++ language on a Pentium 4, 3.0G CPU platform with hyper threading and 2GB DDR RAM. In the experiments, a mobile data generator produces simulation data according to the six attributes of a mobile user, which are user ID, gender, age, marital status, occupation, and hobby. Each record of the simulation data consists of user ID, date,

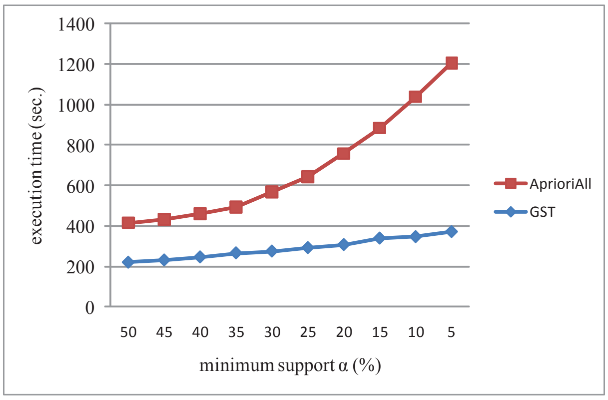

Fig. 4: Execution time in finding FMPs

and moving sequences. If a site is generated sequentially in the moving sequence of a mobile user, it would appear only once in the moving sequence; this implies that the user stays at this site for a long period of time. But, the same site (e.g., site A) can appear more than once in a moving sequence, when a user leaves from site A to site B and then moves back to site A again. In the experiments, we first created the six-attribute records of 5000 mobile users and then, according to them, produced 100 log data records for each user. For the architecture of the proposed method as shown in Figure 2, we have three parameters α, β, and γ respectively used in the graph search technique, the Apriori algorithm, and the characterizing method. The parameter α indicates the minimum support value used in the GST algorithm, and the parameter β indicates another minimum support value used in the Apriori algorithm. In the characterizing method, there is the relevant percentage γ for the attribute-combined values of mobile users.

4.2 Experiments

In this section, two experiments are conducted to evaluate the performance of the GST and/or AprioriAll algorithm. In the first experiment, we compare the execution time of both algorithms in finding frequent moving patterns. Then, in the second experiment, we analyze how many patterns can be found under different minimum support values and relevant percentages. As shown in Figure 4, we observed that the GST algorithm has much better performance than the AprioriAll algorithm in execution time. Also, when the value α is varied from 50% to 5%, the execution time of the GST algorithm is still stable. However, since the AprioriAll algorithm always scans the whole database, it needs more execution time to generate candidate sequences and large sequences under small minimum support values.

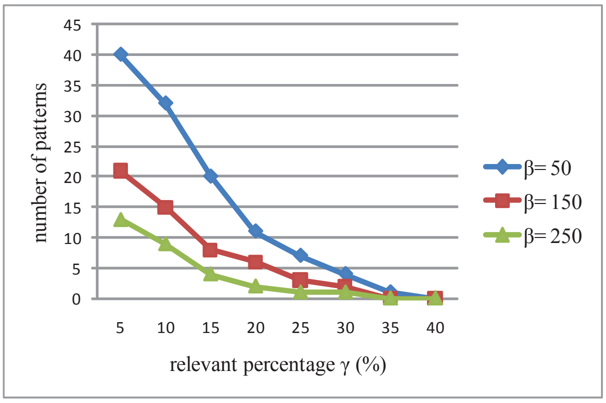

Fig. 5: Number of found patterns with different β and γ

In the second experiment, the value α is set to 25% and we observed the number of found patterns under different minimum support values β and relevant percentages γ, as shown in Figure 5. The value β represents the minimum number of mobile users in a group. We found when the value β increases, the number of found patterns would decrease. As well, higher relevant percentages γ make the number of found patterns decrease.

Finally, we analyze the partially found patterns under different β and γ, as shown in Table 12. We found that for the same relevant percentage γ, some patterns would not appear when the value β increases. Also, for the samefda β, some patterns would not be found under higher γ.

Table 12: Analyses of the partially found patterns under different β and γ

β	γ	Patterns: characterized attribute values
50	15	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book {H, A, F}: Female, 50~59 years old, married, financial industry, investment {A, C, B}, {B, G, C, B}: Male, 20~29 years old, not married, manufacturing industry, traveling
150	15	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book {A, C, B}, {B, G, C, B}: Male, 20~29 years old, not married, manufacturing industry, traveling
250	15	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book {B, G, C, B}: Male, 20~29 years old, not married, manufacturing industry, traveling
50	30	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book
150	30	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book
250	30	{G, B, C}: Male, 20~29 years old, not married, students, traveling and playing ball {G, D, E}: Female, under 19 years old, not married, students, reading book

5 Conclusions and Future Work

In this paper, we attempt to find the frequent moving patterns of a specific group of mobile users. First, an efficient sequential pattern mining method called graph search technique is used to find the frequent moving patterns of each mobile user. Then, the association rule mining and the characterizing method are applied to discover the relevant attribute values corresponding to a specific group of mobile users. Through the experiments, we observed that the GST algorithm not only has better performance in execution time, but also is more stable than the AprioriAll algorithm. Furthermore, we also observed the number of found patterns under different minimum support values and relevant percentages. For the future work, our method can practically apply on the real log data of a telecommunication company to provide more useful information and added value services to a specific group of users, instead of a single user. Furthermore, for the found patterns of users, we can predict their next stops and send relevant information or services to them in time, thereby reducing communication time and saving network bandwidth.

References

[1] R. Agrawal and R. Srikant, Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases, 487-499 (1994).

[2] R. Agrawal and R. Srikant, Mining sequential patterns. Proceedings of the 11th International Conference on Data Engineering, 3-14 (1995).

[3] Ming-Syan Chen, Jong-Soo Park, and P. S. Yu, Efficient data mining for path traversal patterns. IEEE Transactions on Knowledge and Data Engineering, **10**, 209-221 (1998).

[4] Tzung-pei Hong, Chan-Sheng Kuo, and Sheng-Chai Chi, Mining fuzzy sequential patterns from quantitative data. In Proc. of IEEE International Conference on System, Man, and Cybernetics, 962-966 (1999).

[5] Cheng-Ming Huang, Tzung-Pei Hong, and Shi-Jinn Horng, Discovering mobile users' moving behaviors in wireless networks. Expert Systems with Applications, **36**, 10809-10814 (2009).

[6] Yin-Fu Huang and Shao-Yuan Lin, Mining sequential patterns using graph search techniques. In Proc. of the 27th Annual International Conference on Computer Software and Applications, 4-9 (2003).

[7] YonSik Lee and Hyun Ko, Efficient STMPM (spatio-temporal moving pattern mining) using moving sequence tree. In Proc. of the 4th International Conference on Network Computing and Advanced Information Management, 432-437 (2008).

[8] Seung-Cheol Lee, Eunju Lee, Wongil Choi, and Ung Mo Kim, Extracting temporal behavior patterns of mobile user. In Proc. of the 4th International Conference on Network

Computing and Advanced Information Management, 455-462 (2008).

[9] J. W. Lee, O. H. Paek, and K. H. Ryu, Temporal moving pattern mining for location-based service. Journal of Systems and Software, **73**, 481-490 (2004).

[10] V. S. Tseng and K. W. Lin, Mining temporal moving patterns in object tracking sensor networks. In Proc. of IEEE International Workshop on Ubiquitous Data Management, 105-112 (2005).

[11] Show-Jane Yen and Arbee L. P. Chen, An efficient approach for discovering knowledge from large databases. In Proc. of the 4th IEEE International Conference on Parallel and Distributed Information Systems, 8-18 (1996).

[12] Ching-Haung Yun and Ming-Syan Chen, Mining mobile sequential patterns in a mobile commerce environment. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, **37**, 278-295 (2007).

Yin-Fu Huang received the B.S. degree in computer science from National Chiao-Tung University in 1979, and the M.S. and Ph.D. degrees in computer science from National Tsing-Hua University in 1984 and 1988, respectively. He is currently a Professor in the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology. Between July 1988 and July 1992, he was with Chung Shan Institute of Science and Technology as an Assistant Researcher. His research interests include database systems, multimedia systems, data mining, mobile computing, and bioinformatics.

Li-Kai Lin received his B.S. degree in computer sciences from Tamkang University and M.S. degree in computer sciences from National Yunlin University of Science and Technology in 2007 and 2011, respectively. He is currently serving in the army. His major areas of interests are database systems, data mining, and mobile computing.