A Method for Mobility Management in Cellular Networks Using Data Mining

Giang Minh Duc[1(✉)], Le Manh[2], and Do Hong Tuan[1]

[1] HCM City University of Technology, Ho Chi Minh City, Vietnam
ducgm.bdg@vnpt.vn, do-hong@hcmut.edu.vn
[2] Van Hien University, Ho Chi Minh City, Vietnam
manhle@uit.edu.vn

Abstract. The Mobility prediction is one of the important issues in mobile computing systems. The moving logs of mobile users in mobile computing environment are stored in the Home Location Registry (HLR). The generated moving logs are used for mining mobility patterns. The discovered location patterns can be used to provide various location based services to the mobile user by the application server in mobile computing environment. Currently, some papers have written about mobility data mining methods of mobile users in cellular communications networks. In this paper, we propose a method which decrease time to compute the mobility patterns.

Keywords: Data mining · Mobility rules · Mobility prediction · Cellular networks

1 Introduction

Currently, with rapid development of cellular communication networks, many people use their mobile devices to search for information on the internet. Almost everyone has a mobile device such as mobile phones, mobile tablets, notebook, etc. Many people also search for information as traveling all over the world. At about 7 billion mobile phones are used around the world in 2015 at the rate of 95, 56% of the world population[1]. Therefore, the aim of the issue is how to ensure the quality of service of mobile networks.

In cellular communication networks, a mobile user can move from one location to another one, which neighbors' cell in the network. When mobile users move like that, the location of mobile users will be constantly updated to Visitor Location Register (VLR) of the system. VLR is an intermediate database to store temporary information about mobile users in the service area of Mobile Switching Center (MSC). Mobile users' location information then is transferred to home location register (HLR). The HLR is a database which is a long-term storage of mobile users' information. The movement history of mobile users is extracted from the log files and stored in the HLR of the MSC. The historical data is used to predict the mobility of mobile users [1, 7].

[1] B. Sanou. (2015, May) www.itu.int/ict. [Online]. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf.

P. Cong Vinh et al. (Eds.): ICCASA 2016, LNICST 193, pp. 193–204, 2017.
DOI: 10.1007/978-3-319-56357-2_20

In addition, the mobility prediction of the mobile users is used to increase the efficiency of the cellular networks [10, 11]. Many application areas including:

- The service providers can calculate in an optimal way when they design structure and bandwidth of mobile networks [12].
- The telecommunications service providers can reduce the number of unnecessary handover in hierarchical macro/femto-cell networks [13].
- Location Based Services (LBS).
- Etc…

2 Problem Definition

Some researchers applied the data mining techniques and other methods with the aim of solving the problem of cell communication networks, such as mobility, disconnect, long delay time, handover, bandwidth continuously changing... However, these methods have a long execution time. Therefore, to improve further the quality of service of the mobile networks, we propose a new contribution as follows: we redefine the CandidateGeneration() function in the UMPMining algorithm in [7] to reduce running time of the algorithm. Results of our experiment show that our proposed algorithm outperforms the UMPMining algorithm in terms of the execution time.

3 Related Work

The techniques which mine the movement patterns of mobile users is mentioned in the article [3, 4].

Mobility Prediction Method based on Transition Matrix (TM) [5] predicted location according to the ability could occur transition “cell-to-cell” of a mobile user is calculated by the previous move. Relying on this basis, the allocation of resources is done in k cells most likely in the neighboring cell. The parameter k is a parameter defined by the user.

In [6], Katsaros et al. used for discovering user mobility patterns from collections of recorded mobile trajectories, and then these patterns are used for the prediction of location and dynamic allocation of resources.

In [7], Yavas et al. proposed an algorithm for predicting the next inter-cell movement of mobile user in PCSs.

The UMPMining algorithm in [7] predicts the next location of mobile users using data mining techniques. Yavas et al presented an AprioriAll based sequential pattern mining algorithm to find the frequent sequences and to predict the next location of mobile users. They compared their algorithm’s results with Mobility Prediction based on Transition Matrix (TM).

The algorithm in [9] is also the same as [7], but the paths storage file of mobile users is stored in the grid node placed at different locations. Data grid [8] provides a geographic distributing database for computational Grid and executes by an algorithm called KMPM (Knowledge Grid Based Mobility Pattern Mining). If the number of nodes increases, the computation time of the algorithm decreases.

In our paper, we propose a method which refines the algorithm of [7] to reduce running time of this algorithm.

4 Implementation

- **Get data from the logs of HLR:**
  The movement of a mobile user from his current cell to another cell will be recorded in a database which is called Home Location Register (HLR). The HLR stores the permanent subscriber information in a mobile network. Every base station keeps a database in which the profiles of the users located in this cell are recorded which is called Visiting Location Register (VLR). The VLR maintains temporary user information like current location for managing requests from subscriber who are out of the home area. The movement history of a mobile user is extracted from the logs on its home location register (Fig. 1).

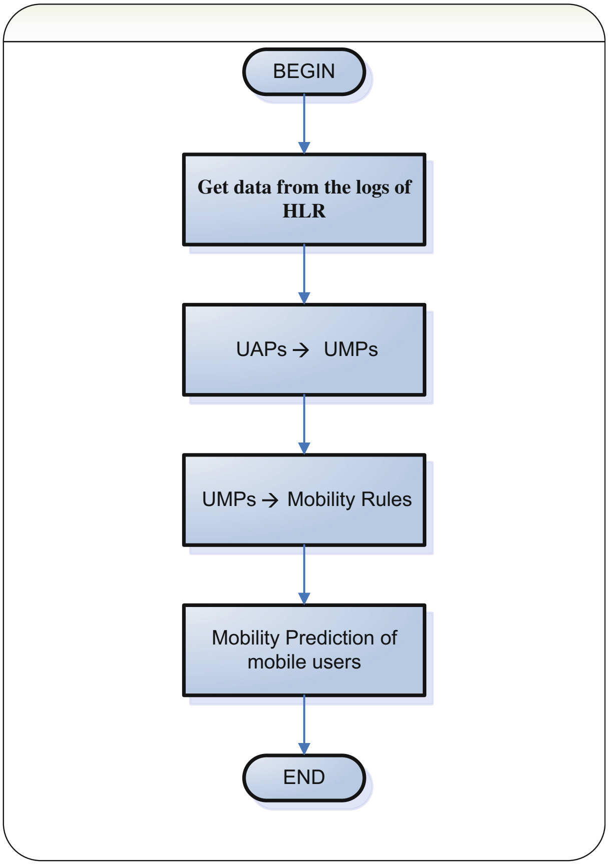

Fig. 1. A method for location prediction in mobile environment

- **Mining the user mobility patterns (UMPs) from user actual paths (UAPs):** A user mobility pattern (UMP) is a sequence of neighboring cells in the coverage region network [9]. The consecutive cells of a UMP should be neighbors because the users cannot travel between non neighbor cells. In order to mine the UMPs from user actual paths (UAPs), sequential pattern mining [7] can be used.

Assume that we have UAPs which have form U = ($c_1, c_2, \ldots, c_n$). Each c_k denotes the ID number [14] of the k_{th} cell in the coverage region.

Mobility Prediction can be defined as the prediction of a mobile user's next movement where the mobile user is traveling between the cells of a cellular network.

Suppose that there are two UAPs, A = $\{a_1, a_2, \ldots a_n\}$ and B = $\{b_1, b_2, \ldots b_m\}$. B is a substring of A, if exist: $1 \leq i_1 < \ldots < i_m \leq n$, $b_k = a_{ik}$, $\forall k$, and $1 \leq k \leq m$.

In addition, B is a substring of A, if all cells of B exist in A (not need sequent in A).

The UMPMining algorithm is a sequence pattern mining algorithm which applied in the movement predict of the cellular networks [7, 9].

UMPMining algorithm

```
Input:   UAPs of database D, min_supp, graph G
Output:  L(UMPs)

C1  ← the length-1 patterns
k = 1
L = ∅      // initially the set is empty
while Ck ≠ ∅
   for each (UAP a ∈ D) do
      S = {s | s ∈ Ck and s is subsequence of a}
      for each s  ∈  S do
          s.count = s.count + s.suppInc
      endfor
   endfor
   Lk = {s | s ∈ Ck, s.count ≥ min_supp }
   L = L ∪ Lk
   Ck+1←CandidateGeneration(Lk,G),∀c∈Ck+1,, c.count=0
   k = k + 1
endwhile
return L
```

At line 13, the CandidateGeneration () function is written as follows:

CandidateGeneration()

```
Input:  Length-k large pattern, L_k
        Coverage Region Graph, G
Output: Length-(k + 1) candidate patterns, Candidates

Candidates = ∅ //Initially the candidates set is empty
For each L = (l_1, l_2, …, l_k), L ∈ L_k {
         {N+ = {v | there is an edge in G such as l_k → v}
          //for each of these neighbor cells, v generate
          a candidate by attaching v to end of L
          For each v ∈ N+ (l_k ) {
             C' = (l_1, l_2, …, l_k, v)
             // Add C' to the candidates' set
             Candidates ← Candidates ∪ C'
          }
        }
Return Candidates
```

The above algorithm is a candidate generation algorithm which proposes in [7] and our algorithm as follows (CandidateGeneration_New) is refined in order to decrease running time of candidate patterns.

CandidateGeneration_New()

```
Input:  Length-k large pattern, L_k
        Coverage Region Graph, G
Output: Length-(k + 1) candidate patterns, Candidates

Rule: set of length(k) is not large which cannot be
subset of large(k+1).

Candidates = ∅
For each L = (l_1, l_2, …, l_k), L ∈ L_k {
  N+={v |there is an edge in G such as l_k → v and v∈L}
  For each v ∈ N+ (l_k) {
     C' = (l_1, l_2, …, l_k, v)
     Candidates ← Candidates ∪ C'
   }
 }
return Candidates
```

At line 8 of the UMPMining algorithm, value 's' is calculated as follows.

For instance, consider UAPs (4, 6, 8, 0, 5), (2, 4, 8, 0, 6) and (1, 2, 4, 6) where the number 4 represents location of mobile user. The support count of the subsequence (4, 6) can be calculated: s.count = s.count + suppInc and suppInc = $\frac{1}{1+totdis}$ where totdis is number of location between 4 and 6. s.count value is 2 because it appears in 1st and 3rd UAP. In 2nd UAP, there are two locations between 4 and 6. Therefore the support value for 4 and 6 is (4, 6).count = 2 + $\frac{1}{1+2}$ = 2.33. It will increase the accuracy of the support counting.

Result with the actual database as follows (Table 1):

Table 1. Result with the actual database

Cn = 2 CANDIDATE	SUPPORT	Ln = 2 PATTERN	SUPPORT
1,7	15.5	1,7	15.5
1,12	15.75	1,12	15.75
1,16	1	1,17	33.5
1,17	33.5	2,21	3
1,18	0.83	3,5	10.34
2,15	0.5	3,17	8.46
2,21	3	3,26	27.67
2,24	0	4,9	20.62
2,45	0	4,23	5
2,47	1.5	4,24	4.5
3,5	10.34	4,25	4.83
3,17	8.46	5,3	11
3,26	27.67	5,16	8.33
4,9	20.62	7,1	11
4,22	0	7,17	3
4,23	5	8,18	6.33
4,24	4.5	8,41	30.33
4,25	4.83	9,4	19
5,3	11	9,21	3.58
5,6	0	10,12	6.98
5,16	8.33	12,1	13.33
5,17	0	12,10	11.83

………………………..

Cn = 3 CANDIDATE	SUPPORT	Ln = 3 PATTERN	SUPPORT
1,7,1	3	1,7,1	3
1,7,17	0	1,12,1	5.83
1,12,1	5.83	1,17,1	10.83
1,12,10	0	3,5,3	2.83
1,12,16	0	3,17,3	1.38
1,17,1	10.83	3,17,26	1.5
1,17,3	0.33	3,26,3	7.5
1,17,7	0.5	4,9,4	5.58
1,17,21	0	4,9,21	1.75
1,17,26	1	4,23,4	2.5
2,21,2	1	4,24,4	2.5
2,21,9	1	4,25,4	1.33
2,21,11	0	5,3,5	3.2
2,21,15	1	5,16,5	4.08
2,21,17	0	7,1,7	5.08
3,5,3	2.83	7,17,7	1.5
3,5,16	0	8,18,8	1.58

……………………….

- **Generation of mobility rules:**
  We can now produce the set of the mobility rules from these UMPs [15]. Assume that we have a UMP C = (c_1, c_2, …, c_k), where k > 1. All the possible mobility rules which can be derived from such a pattern are:

$$
\begin{aligned}
&(c_1) \rightarrow (c_2, \ldots, c_k) \\
&(c_1, c_2) \rightarrow (c_3, \ldots, c_k) \\
&\ldots \\
&(c_1, c_2, \ldots, c_{k-1}) \rightarrow (c_k)
\end{aligned}
$$

For example, we have a form UMP is (3, 4, 5). The mobility rules as follows:

$$
\begin{aligned}
(3) &\rightarrow (4, 5) \\
(3, 4) &\rightarrow (5)
\end{aligned}
$$

For a mobility rule, we call the part of the rule before the arrow the *head* of the rule, and the part after the arrow the *tail* of the rule. Moreover, when these rules are generated, a confidence value is calculated for each rule. For a mobility rule R: $(c_1, c_2, \ldots, c_{i-1}) \rightarrow (c_i, c_{i+1}, \ldots, c_k)$, the confidence is determined by using the following formula:

$$\text{Confidence}\,(\text{R}) = \frac{(c_1, c_2, \ldots, c_k) \cdot count}{(c_1, c_2, \ldots, c_{i-1}) \cdot count} \times 100$$

By using the mined UMPs, all possible mobility rules are generated and their confidence values are calculated. Then the rules which have a confidence higher than a predefined confidence threshold (min_conf) are selected [15, 16].

we have the results table from actual data as follows (min_conf = 5%):

Table 2. The rules result

The rules:

SN	Head	Tail	Confidence
0	1	7	11.8
1	1	12	18.8
2	1	17	36.6
3	3	5	10.5
4	3	17	7.3
5	3	26	25.3
6	4	9	28.5
7	4	23	6.6
8	5	3	13.9
9	5	16	12.1
10	7	1	19.4
11	7	17	8.9
....			
917	53,61	88	9.6
918	53,66	52	75.8
919	54,63	56	6.5
920	54,63	59	5.4
921	54,63	79	36.1
922	54,63	88	8.9
923	54,65	69	12.5
924	54,79	63	64.7
925	56,63	54	64.6
926	56,63	88	50.2
927	56,64	54	99.3
928	57	99,85	6.5

- **The movement prediction of mobile users:**
  From the above rules results (set of mobility rules R), we find out the set of predicted cells as follows (in our work [16]):

Pre_Mov algorithm

```
Input: Current movement of the user: P = (c1, c2, …, ci)
       Set of mobility rules: R.
Output: Set of predicted cells: Pre_Cells.

1. Pre_Cells = ∅    // assign set Pre_Cells = ∅
2. n = 1             // n: cardinal number of Rule_Array
3. For each r: (i1, i2,…, ij)→(ij+1,…, ik)∈ R do
4.    if (i1, i2, …, ij) ⊆ P and ij = ci  then
5.       Pre_Rule ← r
6.       Rule_Array[n] ← (Pre_Rule, confidence value)
7.       n = n + 1
8.    endif
9. Endfor
   // in descending order with respect to confidence
   value.
10. Sort (Rule_Array)
11. for i = 1 to n do
       //get the first cell that is on the right side of
         each rule in Rule_Array
12.    Pre_Cells ← Right_cell
13.    n = n + 1
14. Endfor
15. Return Pre_Cells
```

In this part, the next movement of user is predicted. Suppose that the movement of a user (up to now) is P = (64, 56, 63). Current this user is being cell 63 of the coverage region. The algorithm finds out the rules as follows: (56, 63) → (54) and (56, 63) → (88) (line 925 and 926 of Table 2). The set of predicted cells is {54, 88} (both cell 54 and cell 88 are selected). The cell 54 is selected first because it has the confidence value more than the confidence value of the cell 88 (64.6 and 50.2) (Fig. 3).

5 Experimental Results

We extracted some of data of logs from HLR and the data is:

- The total number of base stations is: 351.
- The total number of user actual paths (UAPs) is: 31415.

C_n, L_n	unrefined		refined	
	quantity	running time	quantity	running time
C_1, L_1	352, 347	27	352, 347	25
C_2, L_2	1493, 1028	125	1489, 1028	69
C_3, L_3	5170, 894	384	4310, 894	184
C_4, L_4	4488, 402	195	1265, 402	56
C_5, L_5	2036, 189	85	487, 189	23
C_6, L_6	931, 64	39	174, 64	9
C_7, L_7	311, 25	13	56, 25	4
C_8, L_8	118, 15	6	21, 15	1
C_9, L_9	74, 9	3	13, 9	1
C_{10}, L_{10}	42, 6	2	7, 6	0
C_{11}, L_{11}	27, 4	1	7, 4	1
C_{12}, L_{12}	20, 2	1	3, 2	0
C_{13}, L_{13}	10, 2	1	3, 2	1
C_{14}, L_{14}	10, 2	0	3, 2	0
C_{15}, L_{15}	10, 0	1	3, 0	0
Total number of running time		883		374

Fig. 2. Experimental results

Where:

- C_1: set of length-1 candidate patterns.
- L_1: set of length-1 large patterns.
- …………….
- C_n: set of length-n candidate patterns.
- L_n: set of length-n large patterns.

Time comparing chart:

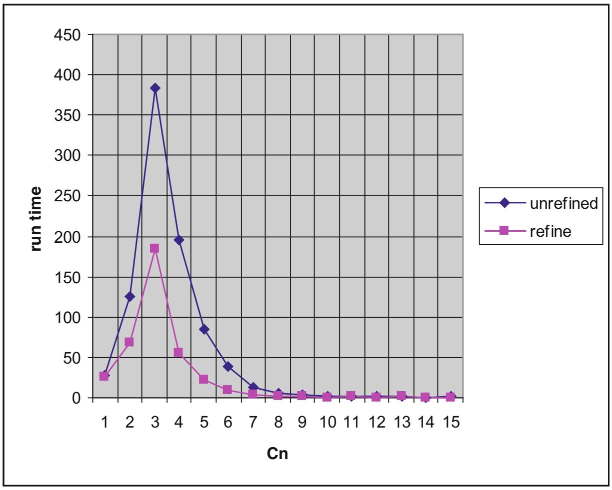

Fig. 3. Comparing time of two algorithms

Our result as follow:

– The number of candidate patterns is reduced:

$C_2 = 1493 - 1489 = 4$
$C_3 = 5170 - 4310 = 860$
$C_4 = 4488 - 1265 = 3223$
$C_5 = 2036 - 487 = 1549$
$C_6 = 931 - 174 = 757$
$C_7 = 311 - 56 = 255$
$C_8 = 118 - 21 = 97$
$C_9 = 74 - 13 = 61$
$C_{10} = 42 - 7 = 35$
$C_{11} = 27 - 7 = 20$
$C_{12} = 20 - 3 = 17$
$C_{13} = 10 - 3 = 7$
$C_{14} = 10 - 3 = 7$
$C_{15} = 10 - 3 = 7$

The total number of candidate patterns is reduced from $C_2 \div C_{15}$: 6899.

– $L_1, L_2, L_3, L_4, L_5, \ldots, L_{15}$ of two algorithms are equal (Fig. 2)
– The total number of running time our algorithm is reduced as follow:

The total number of running time of the algorithm in [7]: 883 s.
The total number of running time of our algorithm: 374 s.
The total number of time reduced: 509 s (57.64%).

6 Conclusion

In this paper, we propose a method which decreases the executing time of the algorithm [7]. In our experimental results, we get data from HLR with 351 base stations and 31415 records of user actual paths (UAPs). The total number of running time of algorithm [7] is 883 s and the total number of running time of our algorithm is 374 s (reduce 57.64%).

References

1. Gok, G., Ulusoy, O.: Transmission of continuous query results in mobile computing sysyems. Inform. Sci. **125**(1–4), 37–63 (2000)
2. Mohan, S., Jain, R.: Two user location strategies for personal communication systems. IEEE Pers. Commun. Mag. **1**, 42–50 (1994)
3. Nanopoulos, A., Katsaros, D., Manolopoulos, Y.: Effective prediction of web user accesses: a data mining approach. In: Proceedings of the WebKDD Workshop (WebKDD 2001) (2001)
4. Nanopoulos, A., Katsaros, D., Manolopoulos, Y.: A data mining algorithm for generalized web prefetching. IEEE Trans. Knowl. Data Eng. **15**(5), 1155–1169 (2003)
5. Rajagopal, S., Srinivasan, R.B., Narayan, R.B., Petit, X.B.C.: GPS-based predictive resource allocation in cellular networks. In: Proceedings of the IEEE International Conference on Networks (IEEE ICON 2002), pp. 229–234 (2002)
6. Katsaros, D., Nanopoulos, A., Karakaya, M., Yavas, G., Ulusoy, Ö., Manolopoulos, Y.: Clustering mobile trajectories for resource allocation in mobile environments. In: R. Berthold, M., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) IDA 2003. LNCS, vol. 2810, pp. 319–329. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45231-7_30
7. Yavas, G., Katsaros, D., Ulusoy, O.: A data mining approach for location prediction in mobile environments. Data Knowl. Eng. **54**, 121–146 (2005)
8. Sakthi, U., Hemalatha, R., Bhuvaneswaran, R.S.: Parallel and distributed mining of association rule on knowledge grid. World Acad. Sci. Eng. Technol. **42**, 316–320 (2008)
9. Sakthi, U., Bhuvaneswaran, R.S.: Mobility prediction of mobile users in mobile environment using knowledge grid. J. Comput. Sci. Netw. Secur. **9**(1), 303–309 (2009)
10. Wu, C.-F., et al.: A novel call admission control policy using mobility prediction and throttle mechanism for supporting QoS in wireless cellular networks. J. Control Sci. Eng. **2011**, 21–31 (2011)
11. Nadembega, A., et al.: An integrated predictive mobile-oriented bandwidth-reservation framework to support mobile multimedia streaming. IEEE Trans. Wireless Commun. **13**(12), 6863–6875 (2014)
12. Aljadhai, A., Znaiti, T.: Predictive mobility support for QoS provisioning in mobile wireless environments. IEEE J. Select. Area Commun. **19**(10), 1915–1930 (2001)

13. Jeong, B., Shin, S., Jang, I., Sung, N.W., Yoon, H.: A smart handover decision algorithm using location prediction for hierarchical macro/femto-cell networks. In: 2011 IEEE 74th Vehicular Conference (VTC Fall), San Francisco, CA, September 2011, pp. 1–5 (2011)
14. Manh, L., Duc G.-M.: Transactions in mobile communication. In: Sixth International Conference on Information Technology for Education and Research in HCM City, pp. 120–126 (2010)
15. Duc, G.M., Manh, L., Tuan, D.H.: A novel location prediction algorithm of mobile users for cellular networks. J. Inf. Commun. Technol. (Res. Dev. Inf. Commun. Technol.) E-3, **8**(12), 58–66 (2015)
16. Duc, G.M., Manh, L., Tuan, D.H.: Mobility patterns mining algorithms with fast speed. Trans. Context Aware Syst. Appl. **2**(6), e2 (2015). http://dx.doi.org/10.4108/eai.5-11-2015.150603