Southeast-Asian J. of Sciences: Vol 1, No 1(2012), pp. 113-125

MODELING MOBILITY IN WIRELESS NETWORK WITH SPATIOTEMPORAL STATE

Thuy Van T. Duong*, **Dinh Que Tran**†

**Department of Information Technology and Applied Mathematics*
University of Ton Duc Thang
Ho Chi Minh, Vietnam
e-mail: vanduongthuy@yahoo.com

†*Department of Information Technology*
Post and Telecommunication Institute of Technology (PTIT)
Hanoi, Vietnam
e-mail: tdque@yahoo.com

Abstract

In the wireless network, mobile users often change their accessing points to Internet. Then, while the mobile node transfers communication from one access point to another access point - called *handover* or *handoff*, it is not able to send or receive data. Predicting mobility is one of the prominent solutions for reducing the handover latency. Several approaches to the prediction have been considered such as Hidden Markov models, machine learning, data mining and so on. The data mining approach investigates the log file of node mobility history to predict its next move. In such a context, the spatial attributes of a mobile node are changing over time, and therefore time constraints between mobile locations play an crucial role. However, the current studies on mobility prediction are not satisfactory since spatial and temporal attributes of data are not considered simultaneously.

In this paper, we introduce a formal model of mobility based on spatial and temporal attributes of data in the wireless network. Based on this model, we construct mobility patterns and weighted mobility rules and then develop algorithms for discovering these patterns and rules. The

Key words: mobitity modeling, mobility prediction, pattern, rule, spatiotemporality.
2000 AMS Mathematics Subject Classification: Applied Mathematics

discovered rules will be utilized in predicting the location of mobile node in the wireless network.

1. Introduction

In the recent years, the need of Internet connection everywhere and at any time has become more and more popular with IEEE 802.11 wireless local area networks (WLAN) Standards. In 802.11 WLAN components, Access PointsAccess Points (AP) (also known as Base Stations) are stations providing services of distributed systems and they are connected to each other via a fixed wired network. APs with parameters such as name of networks and the channel used by AP, etc., offer wireless connection with the mobile nodes such as laptops, Personal digital assistants (PDAs) or mobile phones.

A handover (handoff) is the transfer of a communication from one AP to another AP as the mobile node moves [1]. During a handover, a mobile node (MN) cannot send or receive any packets thus the packet loss may occur as well [2]. There have been a considerable amount of researches on handover latency to reduce the delay in handover and to achieve a seamless handover with the meaning that the user is unware of such status [3]. In 802.11 WLAN, the mobile nodes gain access to the network through an AP by scanning all available channels with active or passive types to find an AP to associate with [3]. This task takes a lot of time and becomes the main factor contributing to the handover latency [1]. Predicting the next location of the mobile node has become an important approach to reduce the handover delay by detecting the identity of the future AP prior to the actual handover ([3] [4]) and it has attracted several research interests ([1] [3] [5] [6]).

Formally, the wireless networks consist of a number of APs which are called cells. The movement of a MN from his current cell to another cell will be recorded in a database which is called the log file of mobility history [5]. Each mobility history contains the identity (ID) of the cell which the MN is connected and the timestamp of this connection. According to ([1] [3] [5] [6] [7] [8] [9] [10]), data mining has been a useful approach to mobility prediction based on the log file of node mobility history. However, the current studies on mobility prediction are not satisfactory when investigating simultaneously spatial and temporal attributes of data. In the context of wireless network, the spatial attributes of a mobile node are changing over time, therefore time constraints between locations need to be considered in predicting the mobility.

The purpose of this paper is firstly to model mobility based on spatial and temporal attributes of data in the wireless network. Based on this model we determine frequent mobility patterns and weighted mobility rules. Then we introduce algorithms for discovering mobility patterns and mobility rules which are used in mobility prediction problem. The remainder of this paper is constructed as follows. Section 2 describes a formal model of mobility with spatiotemporal state. Section 3 presents our proposed algorithms for discovering mobility patterns and mobility rules. Section 4 is some discussion and related works. Section 5 is conclusions and further work.

2. Formal Model for Mobility in Wireless Network based on Time and Space

2.1. Modeling Mobility and Pattern

Given a directed graph G, where each vertex is a cell in the coverage region and the edges of G are formed as follows: if two cells, say A and B, are neighboring cells in the coverage region then G has a directed and unweighted edge from A to B and also from B to A. These edges demonstrate the fact that a mobile node can move from A to B or B to A directly. The mobile node can travel around the coverage region corresponding G.

Let c be the ID number of the cell to which the mobile node connected at the timestamp t, we define a point as follows:

Definition 1. *Let C and T be two sets of ID cells and timestamps, respectively. The ordered pairs p = (c, t), in which $c \in C$ and $t \in T$, is called a point. Denote P the set of all points $P = C \times T = \{(c,t)|c \in C \text{ and } t \in T\}$.*

Two point $p_i = (c_i, t_i)$ and $p_j = (c_j, t_j)$ are said to be equivalent if and only if $c_i = c_j$ and $t_i = t_j$. Point $p_i = (c_i, t_i)$ is defined to be earlier than point $p_j = (c_j, t_j)$ if and only if $t_i < t_j$, and it is denoted as $(c_i, t_i) < (c_j, t_j)$ or $p_i < p_j$

Definition 2. *The trajectory of the mobile node is defined as a finite sequence of points $\{p_1, p_2, \ldots, p_k\}$ in $C \times T$ space, where point $p_j = (c_j, t_j)$ for $1 \leqslant j \leqslant k$.*

Note that the value of each timestamp t_j is not unique in a trajectory, i.e. t_j may be equal to t_i if they are timestamps of two consecutive points of a trajectory. For example $< (c_1, t_1), (c_2, t_2), (c_3, t_2), (c_4, t_4) >$ is a trajectory. The ascending order of points of trajectory is sorted by using t as the key.

Definition 3. *Given a maximal time gap, gap_{max}, a sequential mobility pattern is a list of temporally ordered points $s =< (c_1, t_1), (c_2, t_2), \ldots, (c_k, t_k) >$, where $t_j - t_{j-1} \leqslant gap_{max}$, for $1 \leqslant j \leqslant k$. A sequence composed of k elements is denoted as a k-pattern.*

For example $< (c_1, t_1), (c_2, t_2), (c_3, t_3), (c_4, t_4) >$ is a 4-pattern.

Definition 4. *A mobility pattern $B =< b_1, b_2, \ldots, b_m >$ is a sub-pattern of another mobility pattern $A =< a_1, a_2, \ldots, a_n >$, where a_i and b_j are points,* *written as $B \subset A$, if and only if there exists integers $1 \leqslant i_1 < \cdots < i_m \leqslant n$ such that $b_k = a_{ik}$, for all k, where $1 \leqslant k \leqslant m$. Here, A is called the super-pattern of B.*

For example, given $A =< (c_4, t_2), (c_5, t_3), (c_6, t_4), (c_8, t_5) >$ and $B =< (c_5, t_3), (c_8, t_5) >$. Then B is a sub-pattern of A and conversely, A is super-pattern of B.

Definition 5. *The transaction database of a mobile node is defined as a temporally ordered set of sequential mobility patterns $D = \{S_1, S_2, \ldots, S_N\}$. Note that the length of each mobility pattern S_i for $1 \leqslant i \leqslant N$ may be not equal to each other, i.e. $\exists i, j : 1 \leqslant i, j \leqslant N$ and S_i is a k-pattern, S_j is a m-pattern then $k \neq m$.*

Definition 6. *([11]) Let $D = \{S_1, S_2, \ldots, S_N\}$ be a transaction database which contains N sequential mobility patterns. The support of pattern S is defined as*

$$support(S) = \frac{|\{S_i | S \subset S_i \text{ and } 1 \leqslant i \leqslant N\}|}{N} \tag{1}$$

Definition 7. *Given a minimum support threshold, $supp_{min}$, a sequential mobility pattern S is defined as a frequent mobility pattern if and only if S has support value satisfying:*

$$support(S) \geqslant supp_{min}$$

2.2. Modeling Rules

Definition 8. *A mobility rule has the form $R : A \rightarrow B$, in which A and B are two frequent mobility patterns and $A \cap B = \varnothing$. Then, A is called the head of the rule, and B is called the tail of the rule.*

For example, given the rule $R :< p_1, \ldots, p_{i-1} > \rightarrow < p_i, \ldots, p_k >$. Then the head and the tail of R are $< p_1, \ldots, p_{i-1} >$ and $< p_i, \ldots, p_k >$, respectively. Because the mobility rule $R : A \rightarrow B$ is generated from the frequent mobility pattern $A \cup B$, the support of the rule is the support of the pattern $A \cup B$, that is:

$$support(R) = support(A \cup B)$$

Definition 9. *([5]) Gien a rule $R : A \rightarrow B$, its confidence value is determined by the formula:*

$$confidence(R) = \frac{support(A \cup B)}{support(A)}$$

$$confidence(R) = \frac{(A \cup B).count}{A.count} \times 100 \tag{2}$$

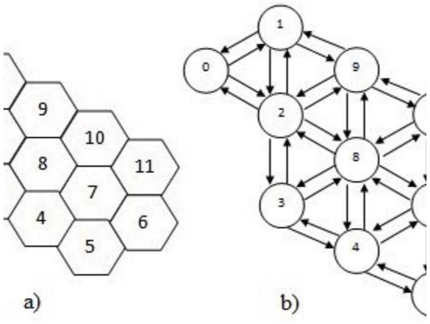

Figure 1: An example coverage region (a) and the corresponding graph G (b)

3. Discovering Mobility Patterns and Rules

3.1. Discovering Frequent Mobility Patterns

This section presents an algorithm, which is the modified version of the Apriori technique [12] [13], to discover all frequent mobility patterns from the transaction database D.

3.1.1. Discovering Frequent Mobility 1-patterns

Assume that the directed graph G has N vertices and the maximal timestamp is T. First, all frequent mobility 1-patterns are extracted from the database D as the following procedure. To discover frequent 1-patterns, for each cell ID c_i for $1 \leqslant i \leqslant N$, we scan the transactional database D to find all points (c_i, t_j) for $1 \leqslant j \leqslant T$. Each point (c_i, t_j) is a 1-pattern. Then their support values are calculated by using the equation (1). All 1-patterns which have support values higher than a predefined minimum support threshold ($supp_{min}$) are selected and called frequent mobility 1-patterns as definition 7.

Table 1: An example transactional database D

Transaction ID	Mobility patterns
1	$\langle (0,t_3),(2,t_4),(8,t_6),(3,t_7),(4,t_8),(7,t_9) \rangle$
2	$\langle (2,t_4),(3,t_5),(8,t_6),(4,t_8),(5,t_9),(7,t_9) \rangle$
3	$\langle (2,t_4),(8,t_6),(3,t_7),(4,t_8),(7,t_9),(6,t_9) \rangle$
4	$\langle (9,t_3),(2,t_4),(8,t_6),(4,t_8),(5,t_9),(7,t_9) \rangle$
5	$\langle (2,t_4),(8,t_6),(4,t_7),(5,t_9),(7,t_9),(6,t_9) \rangle$

Table 2: Mobility 1-patterns and Frequent mobility 1-patterns

C_1		L_1	
Candidates 1-patterns	**Support**	**Frequent 1-patterns**	**Support**
$\langle(0, t_3)\rangle$	1	$\langle(2, t_4)\rangle$	5
$\langle(2, t_4)\rangle$	5	$\langle(3, t_7)\rangle$	2
$\langle(3, t_5)\rangle$	1	$\langle(4, t_8)\rangle$	4
$\langle(3, t_7)\rangle$	2	$\langle(5, t_9)\rangle$	3
$\langle(4, t_7)\rangle$	1	$\langle(6, t_9)\rangle$	2
$\langle(4, t_8)\rangle$	4	$\langle(7, t_9)\rangle$	5
$\langle(5, t_9)\rangle$	3	$\langle(8, t_6)\rangle$	5
$\langle(6, t_9)\rangle$	2		
$\langle(7, t_9)\rangle$	5		
$\langle(8, t_6)\rangle$	5		
$\langle(9, t_3)\rangle$	1		

Let L_1 be a set of frequent mobility 1-patterns

$$L_1 = \{< (c_i, t_j) > |support(< c_i, t_j >) \geqslant supp_{min} \text{ for } 1 \leqslant i \leqslant N \text{ and } 1 \leqslant j \leqslant T\}$$

Example 1: Let the maximal timestamp $T = 11$, $supp_{min} = 2$, transaction database is in Table 1 and directed graph is in Figure 1, then L_1 is in Table 2. Note that the support values of 1-patterns $\langle(0, t_3)\rangle$, $\langle(3, t_5)\rangle$, $\langle(4, t_7)\rangle$ and $\langle(9, t_3)\rangle$ are less than $supp_{min}$ so they are discarded and called outliers (C_1 in Table 2).

3.1.2. Discovering Frequent Mobility k-Patterns for $k \geqslant 2$

First, all 2-patterns C_2 are generated by using the set of frequent 1-patterns L_1. A candidate 2-pattern is created by adding to the end of a frequent 1-pattern a neighbor cell in the consideration of the time constraint. Let $F = \langle(c_i, t_j)\rangle$, for $1 \leqslant i \leqslant N, 1 \leqslant j \leqslant T$, be a frequent 1-pattern, $V(c_i)$ be a set of cells which are neighbors of c_i in G:

$$V(c_i) = \{v| \text{ there is an edge in } G \text{ such as } c_i \rightarrow v\}$$

For each $v \in V(c_i)$, generating all points (v, t_k) for $1 \leqslant t_k \leqslant T$ where $\langle(v, t_k)\rangle$ is a frequent 1-pattern and $(v, t_k) > (c_i, t_j)$ (definition 3)

$$P(c_i) = \{p = (v, t_k)|\langle(v, t_k)\rangle \in L_1 \text{ and } t_k \geqslant t_j\}$$

For each point $p \in P(c_i)$, a candidate 2-pattern C is generated by attaching $p = (v, t_k)$ to the end of $F = \langle (c_i, t_j) \rangle$:

$$C = \langle (c_i, t_j), (v, t_k) \rangle$$

Then adding C to the set of candidates 2-pattern:

$$C_2 = C_2 \cup C$$

This procedure is repeated for all the frequent 1-patterns in L_1. Then all candidates 2-patterns, which have support values higher than $supp_{min}$, are selected and called frequent 2-patterns

$$L_2 = \{C | C \in C_2 \text{ and } supp(C) \geqslant supp_{min}\}$$

Example 2 (continued): For frequent mobility 1-pattern $\langle (2, t_4) \rangle$, we have:
$V(2) = \{0, 1, 3, 8, 9\}$
$P(2) = \{(3, t_7), (8, t_6)\}$
$C_2 = \{\langle (2, t_4), (3, t_7) \rangle, \langle (2, t_4), (8, t_6) \rangle\}$

For frequent mobility 1-pattern $\langle (8, t_6) \rangle$, we have:
$V(8) = \{2, 3, 4, 7, 9, 10\}$
$P(8) = \{(3, t_7), (4, t_8), (7, t_9)\}$
$C_2 = \{\langle (8, t_6), (3, t_7) \rangle, \langle (8, t_6), (4, t_8) \rangle, \langle (8, t_6), (7, t_9) \rangle\}$

All candidates 2-patterns discovered from L_1 in Table 2 are represented by C_2 in Table 3. L_2 in Table 3 shows all frequent mobility 2-patterns.

For $k \geqslant 2$, candidate k-pattern are discovered as follows. Given a frequent $(k-1)$-pattern $F = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_{k-1}, t_{k-1}) \rangle$. Let $V(c_{k-1})$ is a set of cells which are neighbors of c_{k-1} in G:

$$V(c_{k-1}) = \{v | v \text{ is a neighbors of } c_{k-1}\}$$

For each $v \in V(c_{k-1})$, generating all points (v, t_k) for $1 \leqslant t_k \leqslant T$ satisfy $\langle (v, t_k) \rangle$ is a frequent 1-pattern and $t_k \geqslant t_{k-1}$:

$$P(c_{k-1}) = \{p = (v, t_k) | \langle (v, t_k) \rangle \in L_1 \text{ and } t_k \geqslant t_{k-1}\}$$

For each $p \in P(c_{k-1})$, a candidate k-pattern C is generated by attaching $p = (v, t_k)$ to the end of F:

$$C = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_{k-1}, t_{k-1}), (v, t_k) \rangle$$

Then adding C to the set of candidates k-pattern:

$$C_k = C_k \cup C$$

This procedure is repeated for all the frequent $(k-1)$-patterns in L_{k-1}. Then, all candidates k-pattern which have support values higher than $supp_{min}$ are selected:

$$L_k = \{C | C \in C_k \text{ and } support(C) \geqslant supp_{min}\}$$

The pseudo-code for discovering all the frequent mobility patterns is presented in Algorithm 2.

Table 3: Mobility 2-patterns and Frequent mobility 2-patterns

C_2		L_2	
Candidates 2-patterns	**Support**	**Frequent 2-patterns**	**Support**
$\langle(2,t_4),(3,t_7)\rangle$	2	$\langle(2,t_4),(3,t_7)\rangle$	2
$\langle(2,t_4),(8,t_6)\rangle$	5	$\langle(2,t_4),(8,t_6)\rangle$	5
$\langle(3,t_7),(4,t_8)\rangle$	2	$\langle(3,t_7),(4,t_8)\rangle$	2
$\langle(4,t_8),(5,t_9)\rangle$	2	$\langle(4,t_8),(5,t_9)\rangle$	2
$\langle(4,t_8),(7,t_9)\rangle$	4	$\langle(4,t_8),(7,t_9)\rangle$	4
$\langle(5,t_9),(6,t_9)\rangle$	1	$\langle(5,t_9),(7,t_9)\rangle$	3
$\langle(5,t_9),(7,t_9)\rangle$	3	$\langle(7,t_9),(6,t_9)\rangle$	2
$\langle(6,t_9),(5,t_9)\rangle$	0	$\langle(8,t_6),(3,t_7)\rangle$	2
$\langle(6,t_9),(7,t_9)\rangle$	0	$\langle(8,t_6),(4,t_8)\rangle$	4
$\langle(7,t_9),(5,t_9)\rangle$	0	$\langle(8,t_6),(7,t_9)\rangle$	5
$\langle(7,t_9),(6,t_9)\rangle$	2		
$\langle(8,t_6),(3,t_7)\rangle$	2		
$\langle(8,t_6),(4,t_8)\rangle$	4		
$\langle(8,t_6),(7,t_9)\rangle$	5		

3.2. Generating Mobility Rules

This subsection presents the algorithm to generate all the mobility rules using the frequent mobility patterns which are discovered in the previous phase. Let S be a frequent mobility pattern, all the possible mobility rules which can be generated from S are: $A \rightarrow (S - A)$ forall $A \subset S$ and $A \neq \varnothing$. Each rule has a confidence value which is computed by using the equation 2. Then the rules, which have confidence values higher than a predefined confidence threshold $conf_{min}$, are selected and called frequent mobility rules. It means that frequent mobility rules are mobility rules satisfying:

$$conf(A \rightarrow (S - A)) = \frac{support(S)}{support(A)} \geqslant conf_{min}$$

For example, a frequent mobility pattern $P = \langle(c_1,t_1),(c_2,t_2),\ldots,(c_k,t_k)\rangle$, where $k > 1$, has all possible rules:

$R_1 : \langle(c_1,t_1)\rangle \rightarrow \langle(c_2,t_2),\ldots,(c_k,t_k)\rangle$
$R_2 : \langle(c_1,t_1),(c_2,t_2)\rangle \rightarrow \langle(c_3,t_3),\ldots,(c_k,t_k)\rangle$
...
$R_{k-1} : \langle(c_1,t_1),(c_2,t_2),\ldots,(c_{k-1},t_{k-1})\rangle \rightarrow \langle(c_k,t_k)\rangle$

Then the rules R_i, for $1 \leqslant i \leqslant (k-1)$, which satisfy $conf(R_i) \geqslant conf_{min}$ are frequent mobility rules. By using all the mined frequent mobility patterns,

Algorithm 1 Frequent mobility patterns Discovery Algorithm

Input: a transactional database, D
minimum support threshold, $supp_{min}$
coverage region directed graph, G
Output: a set of frequent mobility patterns, L

```
//Let $C_k$ is a set of candidates $k$-patterns
//Let $L_k$ is a set of frequent mobility $k$-patterns
$L_1 \leftarrow$ a set of the frequent mobility 1-patterns
$k = 1$
repeat
    $C_{k+1} \leftarrow$ CandidateGeneration($L_k$)
    for all mobility pattern $F \in D$ do
        $C \leftarrow \{c | c \in C_{k+1}$ and $c \subset F\}$
        for all $c \in C$ do
            $c.count = c.count + 1$
        end for
    end for
    $L_{k+1} = \{c | c \in C_{k+1}$ and $c.count \geqslant supp_{min}\}$
    $L = L \cup L_{k+1}$
    $k = k + 1$
until $L_k = \varnothing$
RETURN $L$
```

all frequent mobility rules are generated and are used in the mobility prediction phase.

Note that the rules, which are extracted from the recent patterns, should be more important than the rules extracted from the older movement patterns.

Definition 10. *Assume that we have two rules:*

$$r_1 : \langle (a_1, t_1), (a_2, t_2), \ldots, (a_j, t_j) \rangle \to \langle (a_{j+1}, t_{j+1}), (a_{j+2}, t_{j+2}), \ldots, (a_k, t_k) \rangle$$
$$r_2 : \langle (b_1, t'_1), (b_2, t'_2), \ldots, (b_i, t'_i) >\to \langle (b_{i+1}, t'_{i+1}), (b_{i+2}, t'_{i+2}), \ldots, (b_l, t'_l) \rangle$$

The rule r_2 is an older rule iff the timestamp of the last position of its tail is smaller than the timestamp of the last position of the tail of r_1 ($t'_l < t_k$). Then, r_1 is called the recent rule.

For an accurate prediction, the recently generated rules should have a greater importance than older rules. and. Each generated rule r_i is assigned a weight value w_i, which is determined through the timestamp of the last position of the rule tail (t_k). Therefore, all the rules extracted from a frequent mobility pattern have the same weight value.

Moreover, if the timestamp of the last position of the head of the rule (t_j) is far away from the query time t, the prediction result can not be accurate.

Algorithm 2 Candidates Generation Algorithm

Input: a set of frequent mobility k-patterns, L_k
coverage region directed graph, G
Output: a set of candidates $(k+1)$-patterns, C_{k+1}

```
for all frequent mobility k-pattern $P_k = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_k, t_k) \rangle \in L_k$ do
    $V(c_k) \leftarrow \{ \ v \mid v$ is a neighbour of $c_k \}$
    for all vertex $v \in V(c_k)$ do
        $P(c_k) \leftarrow \{p = (v, t_{k+1}) | \langle (v, t_{k+1}) \rangle \in L_1$ and $t_{k+1} \geqslant t_k\}$
        for all $p = (v, t_{k+1}) \in P(c_k)$ do
            $C = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_k, t_k), (v, t_{k+1}) \rangle$
            $C_{k+1} = C_{k+1} \cup C$
        end for
    end for
end for
RETURN $C_{k+1}$
```

Thus, T_{diff} is included to incorporate temporal similarity to the query time. The algorithm is extended by matching the current trajectory of a node to the rule head and finding the best matching using the time constraints. The best matched rule is temporally closest to the query time. The proposed mobility rules generation algorithm is presented in Algorithm 3.

4. Related Work & Discussion

Our work uses a modified version of the Apriori technique to discover all frequent mobility patterns. This technique is suitable for our problem since we can make it efficient by pruning most of the candidates generated at each iteration step of the algorithm based on the topology constraint as follows: all candidates with length-$(k+1)$ $C_{k+1} = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_{k+1}, t_{k+1}) \rangle$, which are generated from the length-k frequent mobility pattern $P_k = \langle (c_1, t_1), (c_2, t_2), \ldots, (c_k, t_k) \rangle$, must satisfy the coverage region directed graph G, i.e. cell c_{k+1} must be neighbour to cell c_k.

Moreover, mining sequences with time constraints allows a more flexible handling of the transactions. Our paper uses temporal attributes to generate transaction database. Each mobile node has a log file of mobility history. It is created whenever a mobile node performs a handover to a new cell. Bergh et al. [6] proposed to generate the transaction database using a log of a node mobility history as follows: a new transaction is started every T seconds to indicate the beginning of a new mobility session. Due to the various reasons such as loss of

Algorithm 3 Mobility Rules Generation Algorithm

Input: a set of frequent mobility patterns, L
minimum confidence threshold, $conf_{min}$
Output: a set of frequent mobility rules, $Rules$

```
$Rules \leftarrow \varnothing$
**for all** frequent mobility $k$-pattern $P_k = \langle(c_1, t_1), (c_2, t_2), \ldots, (c_k, t_k)\rangle \in L, k \geqslant 2$ **do**
    $P_l \leftarrow P_k$ //$P_l$ is a $l$-pattern
    **repeat**
        //$A \leftarrow$ dropping the last point of $P_l$ i.e. $A$ is a $(l-1)$-subpattern of $P_l$
        $A \leftarrow \langle(c_1, t_1), (c_2, t_2), \ldots, (c_{l-1}, t_{l-1})\rangle$
        $conf \leftarrow support(P_k)/support(A)$
        **if** $conf \geqslant conf_{min}$ **then**
            //$R \leftarrow (A \rightarrow P_k - A)$
            $R \leftarrow \{\langle(c_1, t_1), (c_2, t_2), \ldots, (c_{l-1}, t_{l-1})\rangle \rightarrow \langle(c_l, t_l), \ldots, (c_k, t_k)\rangle\}$
            $R.w = year(t_k)$ //all rules generated in the same year have the same weight
            $Rules = Rules \cup R$
        **else**
            break
        **end if**
        $l = l - 1$
    **until** $l > 1$
**end for**
RETURN $Rules$
```

connection or delaying of update in application environment, the transaction database obtained by the above procedure may be incorrect. In order to tackle this problem, we apply the time constraints between locations of a node to make valid transactions. Time constraints restrict the time gap between two locations in a transaction. The occurrence time point of consecutive movements, which are denoted by t_j and t_{j+1}, satisfies:

$$t_{j+1} - t_j \leqslant gap_{max}$$

Our approach to generating frequent mobility rules is based on the work by Yavas et al. [5]. However, rather than neglecting the temporal factor of rules, the rules which are extracted from the mobility patterns are assigned a weight value with time. Since in our viewpoint, the rules from the recent patterns should be more important than the rules extracted from older mobility patterns. Our work also reveals that if the timestamp of the last position of the head of the rule (t_j) is far away from the query time t, the prediction result may not be accurate. Thus, T_{diff} is included to incorporate temporal similarity to the query time. The algorithm extends the problem of matching the current trajectory of a node to the rule head and finding the best matching using the time constraints. The best matched rule is temporally closest to the query time.

5. Conclusions

This paper presented a mathematical approach of mobility in the wireless network. We model mobility with spatiotemporal state and then define mobility patterns and mobility rules based on this model. Algorithms for discovering mobility patterns and weighted mobility rules have been developed. In this paper, we use the modified version of the Apriori technique to discover significant patterns in the transaction database, which is generated from the log file of node mobility history. Then, mined significant patterns are used to extract all the mobility rules. Our extension approach utilizes temporal attribute in order to assign weight value for each mobility rule. The discovered weighted rules are used in predicting the location of mobile node in the wireless network. The time constraints allows a more flexible and accurate handling of the mobility prediction. Developing mobility prediction algorithm based on this model and analyzing as well as evaluating its performance will be our future work.

References

[1] Jean-Marc Francois, *Performing and Making Use of Mobility Prediction*, Doctor thesis, University of Lyte, (2007).

[2] Shilpy Gupta, Amit Sharma and Mohit Chowdhary, *An Improved Framework for Enhancing QoS in MIPv6*, International Journal on Computer Science and Engineering , ISSN : 0975-3397, Vol. 3 No. 2, (2011).

[3] Shanthy Lucy Menezes, *Optimization of handovers in present and future mobile communication networks*, Doctor of philosophy in computer science, University of Texas at Dallas, (2010).

[4] Youn-Hee Han and Seung-Hee Hwang, *Movement detection analysis in Mobile IPv6*, IEEE Communications letters, Vol. 10, No. 1, (2006).

[5] G.Yavas, D. Katsaros, O. Ulusoy and Y. Manolopoulos, *A data mining approach for location prediction in mobile environments*, Journal of Data and Knowledge Engineering, Vol. 54 (2005), 121-146.

[6] Andre E. Bergh and Neco Ventura, *Prediction Assisted Fast Handovers for Mobile IPv6*, IEEE MILCOM 2006 Unclassified Technical Sessions, Washington D.C., (2006).

[7] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, M. Kyriakakos, A. Kalousis, *Predicting the location of mobile users: a machine learning approach*, ACM 978-1-60558-644-1/09/07, London, (2009).

[8] D. Taniar and J. Goh, *On mining movement pattern from mobile users*, International journal of distributed sensor networks, ISSN: 1550-1329, 3: 69-86, (2007).

[9] Thuy Van T. Duong, Dinh Que Tran and Cong Hung Tran , *Spatiotemporal Data Mining for Mobility Prediction in Wireless Network*, National Conference of Fundamental and Applied IT Research (FAIR), Dongnai, Vietnam (2011).

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

[11] Vincent S. T. , and Kawuu W. L., *Efficient mining and prediction of user behaviour patterns in mobile web systems*, Elservier, Information and software technology, 357-369, (2006).

[12] Xiaobai Yao, *Research Issues in Spatio-temporal Data Mining*, Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Science, Virginia (2003).

[13] S. Shekhar, R. R. Vatsavai, M. Celik, *Spatial and Spatiotemporal Data Mining: Recent Advances*, (2008). Available at: http://www-users.cs.umn.edu/ shekhar/research/sdm-chapter-ngdm-v2-2008.pdf