OXFORD UNIVERSITY PRESS | The Computer Journal

# Predicting the Location and Time of Mobile Phone Users by Using Sequential Pattern Mining Techniques



| | |
|---|---|
| Journal: | *The Computer Journal* |
| Manuscript ID: | COMPJ-2015-03-0106.R1 |
| Manuscript Type: | Original Article |
| Date Submitted by the Author: | 16-May-2015 |
| Complete List of Authors: | Ozer, Mert; Arizona State University, School of Computing, Informatics, Decision Systems Engineering<br>Keles, Ilkcan; Aalborg University,, Computer Science<br>Toroslu, Hakki; Middle East Technical University, Computer Engineering<br>KARAGOZ, PINAR; Middle East Technical University, Computer Engineering<br>Davulcu, Hasan; Arizona State University, School of Computing, Informatics, Decision Systems Engineering |
| Key Words: | Human Mobility Patterns, Mobile Phone User, Sequence Mining, Location and Time Prediction |
| | |



SCHOLARONE™ Manuscripts

For Review Only

# Predicting the Location and Time of Mobile Phone Users by Using Sequential Pattern Mining Techniques

Mert Ozer[1], Ilkcan Keles[2], Hakki Toroslu[3], Pinar Karagoz[3] and Hasan Davulcu[1]

[1]*School of Computing, Informatics, Decision Systems Engineering, Arizona State University, USA*
[2] *Department of Computer Science, Aalborg University, Denmark*
[3]*Computer Engineering Department, METU, Turkey*
*Email: mozer@asu.edu, ilkcan@cs.aau.dk, toroslu@ceng.metu.edu.tr, karagoz@ceng.metu.edu.tr, hdavulcu@asu.edu*

**In recent years, using cell phone log data to model human mobility patterns became an active research area. This problem is a challenging data mining problem due to huge size and the non-uniformity of the log data, which introduces several granularity levels for the specification of temporal and spatial dimensions. This paper focuses on the prediction of the location of the next activity of the mobile phone users. There are several versions of this problem. In this work, we have concentrated on the following three problems: Predicting the location and the time of the next user activity, predicting the location of the next activity of the user when the location of the user changes, and predicting both the location and the time of the activity of the user when the user's location changes. We have developed sequential pattern mining based techniques for these three problems and validated the success of these methods with real data obtained from one of the largest mobile phone operators in Turkey. Our results are very encouraging, since we were able to obtain quite high accuracy results under a small prediction sets.**

*Keywords: Human Mobility Patterns; Mobile Phone User; Sequence Mining; Location and Time Prediction*

*Received ; revised*

For Review Only

## 1. INTRODUCTION

Since the introduction of the first mobile phones, especially after 1990s, mobile phones quickly became indispensable devices for ordinary people. Nowadays almost 95% of the people in the world use mobile phones. Mobile phone usages of people generate huge amount of data for mobile phone operators. This data is mainly used for generating customer invoice.

However, in addition to the information used for generating invoice such as caller and callee information, the time and the duration of the call, this data also contains location information of both the caller and the callee. This location information is not precise since the mobile phone operators only keep/know the base station id of both users, not exact locations. Although exact locations of mobile phone users can be determined, it is typically not obtained by mobile phone operators, since it is not feasible. Some operators use coarse location data to improve their service quality, and some of them exploit this data to create new forms of businesses such as generating appropriate advertisement messages to users selected according to their predicted movements [1, 2, 3, 4].

One of the most well-known problems related to the user location information is the prediction of the next location of the mobile phone user. Users' navigation behavior patterns are important knowledge for mobile phone operators, so that they can calculate potential next location of individual users in order to be able to optimize their advertisement strategies. There are also other potential usages of user behavior patterns in terms of mass people movement modeling, such as city planning and traffic optimization.

In this paper, we focus on predicting individual mobile phone user's next location using her previous log data. User log data, also named as Call Detail Record (CDR), contains the base station identifiers (and their locations) for the caller and callee and the

time of the activity (such as voice call, sending SMS or use of internet). This historical data can be processed using sequential pattern mining and time series analysis techniques in order to predict the time and the location of the next event for users. The main challenges of this problem are due to the huge size and the non-uniformity of the data. User events do not come with uniform distribution in time or spatial dimensions. Sometimes events are very rare and sometimes are very often. Similar non-uniform pattern can also be observed in terms of location distribution of the data. However, a simple analysis also shows that 80% of the users' location of next activity is the same as their current location (in terms of the base station identifiers their mobile phones are connected to). Only 20% of the two consecutive events are at different locations.

Although the prediction of the next location of the mobile phone users seems like a well-defined problem, since it contains different parameters, several different variations of it can be defined. In this paper, we have investigated three versions of the next location prediction problem, which are listed below:

- Determining the location and the time of the next user activity, regardless of whether the location of the user changes or not,
- Predicting the location of the next activity of the user when the location of the user changes,
- Predicting both the location and the time of the activity of the user when the user's location changes.

In this study, we have utilized CDR data obtained from one of the largest mobile phone operators in Turkey. Typically each mobile phone activity is associated with the closest base station. Therefore, each base station can be assumed to be defining a region covering the activities in that region. In CDR data the exact time of each activity is recorded. However, in the time prediction of user activity, exact time is not very informative. Therefore, we have divided a day into time intervals in our process.

Also, we have clustered base stations according to their locations into regions and aimed to predict the region of the next activity of the user in terms of these regions. In the first problem, we have tried to predict both the region and the time interval of the next activity.

For the second problem, we have focused on only predicting the location of the next activity of the user in case the user's location changes. Since for 80% of the activities the location of the the activity is same as the location of the previous activity, the location change problem is important.

Finally, in the last problem we have tried to predict both the time and the location of the user's next activity when the location of the user changes. This is a kind of the extended version of the second problem. Basically, we aimed to show that using time information, in terms of time intervals, in addition to the location change information, increases the accuracy of the predicting the changed location and the time of the change.

We have made an extensive set of experiments to measure the applicability and the accuracies of these approaches using real data of more than 1 million mobile phone users for a period of 1 month for a region of roughly 25000 km$^2$. Usually there is a typical tradeoff in this kind of prediction problems such that in order to increase the accuracy of a prediction it might be necessary to make a large number of alternative suggestions. When the suggestion or prediction set gets smaller, usually the accuracy of the prediction quickly drops. Our results are very encouraging, since for each problem, high accuracy values are obtained by making only a very small number of alternative suggestions. Our solutions for the first and second problem with limited experimental analysis were also included in our previous works [11] and [5]. A shorter version of our solution for the third problem was also studied in [6].

The rest of the paper is organized as follows. Section 2 introduces previous work on location prediction. Section 3 presents the details of the data and the problem definition. Section 4 introduces the proposed solutions for the problem defined. Section 5 contains the experimental results of our proposed methods. Section 6 concludes our work and points out possible further studies.

## 2. RELATED WORK

In recent years, variety of location prediction schemes on human mobility have been studied in various dimensions [7], [8], [9], [10],[11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [2], [21], [22].

Interesting findings about the human mobility habits and its predictability are reported in [14] and [15]. In [14], Montjoye et al. propose a method using both Voronoi diagrams involving base stations and spatial and temporal properties of users' movement data to find the minimum number of points enough to uniquely identify individuals. They show that, almost for all users, distinct sequences with four spatio-temporal points exist in CDR data. Therefore, such short sequences are sufficient to uniquely identify 95% of the users, while sequences of length two characterize more than 50%.

In [15], Song et al. analyze the limits of predictability in human mobility. They used the data collected from 50,000 mobile phone users for 3 months. They propose three entropy measures which are believed to be the most fundamental quantities to analyze the limits of predictability, the random entropy, the temporal-uncorrelated entropy and the actual entropy. They also use a probability measure for correctly predicting user's future movements. They find that there is a 93% potential predictability in user mobility at best and 80% at worst for any user [15].

For Review Only

In [23], Zheng et al. investigate human mobility from mobile data both in individual and group behavior aspects. They use probabilistic, unsupervised approach for uncovering the behavior similarity among users and for clustering individual behaviors.

There are other methods to use for location prediction problem rather than sequential pattern mining such as Markov models and expectation maximization algorithms. In [10], Thanh et al. make use of Gaussian distribution and expectation maximization algorithm to learn the model parameters. Then, mobility patterns, where each is characterized by a combination of common trajectory and a cell residence time model, are used for making predictions. They use Gaussian mixture models to find similarities in cell-residence times of mobile users. They outperform the methods that ignore temporal characteristics of user movements. However, they are in need of studying their method in real data.

In [12], Gao et al. use both spatial and temporal data to predict users' location. They propose ten different models which can be categorized as spatial-based, temporal-based and spatio-temporal-based. They make use of Bayes' rules for their prediction models which use historical data while predicting the next location. They also make use of Markov models to build two of their models. For the best model named as HPY Prior Hour-Day Model, they managed to predict user locations with an accuracy rate of 50%. They do not use any social network information together with spatio-temporal patterns.

In [13], Gidofalvi et al., propose a method which use both spatial and temporal GPS data for building Markov model which is used for next location and time prediction of user. In other words, they both predict the change of location and the time of this change. They use an Inhomogeneous Continuous-Time Markov (ICTM) model since the prediction depends on the previous locations and time. They use both spatial and temporal information for building the model. Their ICTM model predicts the departure time correctly with the 45 minute error and the next region correctly 67% of the cases.

Similar to our work, in [18], [19], [11] and [20], the authors propose sequential pattern mining techniques for the location prediction problem. In [18], Yavas et al. propose an AprioriAll-based algorithm which is similar to our three methods. They extract frequent user trajectories which they name user mobility patterns (UMP) from a user move database and predict the user's next movement accordingly. However they do not use any spatial or temporal information while extracting UMPs or generating predictions. The rules consist of only cell ids rather than any spatial attribute. They introduce alignment parameters on the length of the sequences and maximum number of predictions as ours. They show that they get higher accuracies for mobility prediction than previously proposed methods using transition matrices.

In [19], Giannotti et al. propose methods to solve different trajectory pattern mining problems. They define spatio-temporal sequences as the pairs of spatial attribute and the time that user has spent in there. They also try to detect the popular regions. The difference of this technique from the conventional sequence pattern mining technique is the use of trajectories (T-patterns) rather than itemsets.

In [20], Cao et al. introduces a method for discovery of periodic patterns in spatio-temporal sequences. They also make use of an AprioriAll-based algorithm for extraction of periodic patterns. The distinctive feature of these periodic patterns is that they are not frequent in the whole time span but in some time interval, so they change their support definition accordingly.

There are various works that aim to further increase the prediction accuracies by the help of social networks. In [7], Cho et al. propose that general human mobility does not have a high degree of freedom and variation as it is believed. They work on three features of human mobility; geographic movement, temporal dynamics and the social network. Social network is used since human mobility is partly driven by our social relationships, e.g. we move to visit our friends. They use three main data sources, where two of them are popular online location based social networks, Gowalla and Brightkite and the other is a trace of 2 million mobile phone user's phone activity in Europe. They find that social relationships can explain about 10% of human movement in cell phone data and 30% of movement in location based social networks. However periodic movement behaviour explains about 50% to 70% of it. They reach 40% accuracy while predicting user's location at any time.

In [8], Boldrini et al. propose a model that integrates three main properties believed to be fundemental for human mobility. First, user mobility largely depends on their social relationships. Second, users are disposed to spend their most of time in a few number of locations. Third, users mostly move shorter distances rather than the longer ones. The main novelty of their model named Home-cell Community-based Mobility Model (HCMM) is to integrate these three features. They incrementally improved HCMM starting with a pure social-based model and mathematically justifying the need for extending the features. Finally, they claim that HCMM is able to regenerate the main properties of human movement patterns.

In [9], Zhang et al. further improve the user mobility models of [8] and [7] by amplifying the effect of social network information in location prediction. They also claim that call patterns are strongly related with co-locate patterns and mainly affect user's short-time mobility. They further propose a method named NextMe which takes social interplay into consideration as well. However this time, when the social interplay will affect social mobility is identified and used accordingly. They validate their scores with the MIT

For Review Only

Reality Mining dataset. They reach up to 60% accuracy levels for the prediction with their NextMe method.

Rather than using social relationships or networks of the user, in [16] and [17] distinctive features of spatial attribute in the data are made use of. In [16], Zheng et al. aim to extract interesting locations such as culturally significant places, shopping malls, city centers etc., and travel sequences from multiple users' GPS logs. They used tree-based hierarchical graph (TBHG) to model user's historical movement patterns then introduce a HITS (Hypertext Induced Topic Search)-based inference model, which represents one of the users' travel to a location as a vertex. The weight of the vertex is defined by user's experience. Location's interest is also defined by user's experience as well as the number of user's visit. They claim that such a model can be used for location recommendation like a mobile tourist guidance. They evaluated their method with the GPS data of the 107 users of a 1 year period.

In [17], Ying et al. propose an algorithm which uses semantic labels for locations rather than just using spatial attributes. They explore semantic trajectories of the users and predict the next location of the user accordingly. Rather than using sequential pattern mining techniques, they use clustering methods for next location prediction. They group users hierarchically according to their semantic trajectories by using Maximal Semantic Trajectory Pattern Similarity (MSTP-Similarity) which they define. It was the first work which combines the semantic tags for location and spatial attributes for next location prediction problem and their proposed location prediction model has a high performance.

## 3. DATA AND PROBLEM DEFINITION

### 3.1. Call Detail Record (CDR) Data

The data used in this study is provided by one of the largest mobile phone operators in Turkey. The CDR data contains more than 1 million user's mobile phone records corresponding to a period of 1 month. The area corresponding to the calls is around 25000 km$^2$ and the population of the area is almost 5 million. Two thirds of the population lives in a large urban area, corresponding to less than 30 percent of the whole area, and the rest of the population is scattered in small towns and villages. Due to this population distribution, most of the 13000 base stations are located in densely populated areas of the region. In rural areas the distances among base stations reach tens of kilometers, while in the downtown area sometimes these distances are as small as hundred meters.

Each record in data represents one of the following mobile user activities; voice caller, voice callee, SMS sender, SMS receiver, GPRS connection. Besides these cases, no record exists in the CDR data. These records consist of 11 attributes. For both the caller (i.e., #1) and callee (i.e., #2), base station id, phone number, province code of the phone number are included. In addition, call time, CDR type, URL, duration, call date also exist in these records. Definition of these attributes and example record attributes are presented in Table 1.

### 3.2. Problem Definition

Due to the content of our data set, the location of user activity corresponds to the location of the base station s/he is connected to. In some dense areas, the base station locations are very close to each other, and sometimes users are not connected to the nearest base stations due to load balancing. Therefore, we have grouped base stations into larger regions and aimed to obtain possible region of the user.

It is possible to construct next location and time prediction model for each user separately from her/his CDR records. Typical weekday and holiday patterns of most users can be constructed using statistical methods if sufficient amount of CDR data (at least a couple of months) is available. Travels, insufficient action records, and heterogeneity of user actions are main drawbacks for constructing models for each user separately. However, in our model we wanted to determine frequent common user patterns from daily user patterns, regardless of which users daily activities support them. Thus, for example, a frequent common pattern may be supported by some of the weekday activities of a number of users. Therefore, this frequent pattern may just correspond to a few hours of a day. This way, when a new user sequence is given, it may be compared against existing sequences to determine if it matches with one frequent pattern, and that frequent pattern can be used to predict potential next location and the time for that user. In this approach, the number of frequent patterns becomes much smaller compared to user based model generation. Also considering users switching mobile phone operators more frequently nowadays, the cold start problem for many users are overcome with this approach.

One month CDR data of almost one million users are used in this study. At the preprocessing phase of raw CDR data, each activity record is converted to a triple as $(User\_Id, Action\_Location, Action\_Time)$. The first field, namely user id, is used only in order to generate a sequence of location and time tuples for each user. We have used these daily sequences as the main input of our sequential pattern mining methods, which is defined as follows:

**Definition (DUAS: Daily User Activity Sequence):** $< (L_1, T_1), ...(L_n, T_n) >$ is a daily user activity sequence obtained from the one day activities of one of the users, in which each $(L_i, T_i)$ pair represents the location and the time of an activity of the selected user.

All the problems that are discussed in this work are based on finding frequent patterns obtained basically

For Review Only

TABLE 1: List of attributes for CDR data

| Attributes | Description |
|---|---|
| base station id#1 | unique integer representing the base station which caller, SMS sender or GPRS user connected to. e.g. 17083 |
| phone number#1 | unique string representing the caller, SMS sender or GPRS user. Due to the privacy reasons, it is not a regular phone number. e.g. 7bcfc0259b9c8a4af95177a7e79bcd28 |
| province code of phone number #1 | an integer that represents the province user started a call or a GPRS connection, or sent an SMS. e.g. 06 |
| base station id #2 | unique integer representing the base station which callee or SMS receiver is connected to. It is null if the type of the record is GPRS connection. e.g. 17083 |
| phone number #2 | unique string that represents the callee or SMS receiver. Due to the privacy reasons, it is not a regular. phone number. It is null if the type of the record is GPRS connection. e.g. 28119ffa652d31607a3bb573bd3d594b |
| province code of the phone number #2 | an integer that represents the province callee or SMS receiver is in. e.g. 06 |
| call time | The time that action started in a "hhmmss" format. e.g. 170251 |
| CDR type | It can be one of the following: voice caller, voice callee SMS sender, SMS receive, GPRS connection |
| URL | It is used only for GPRS data. It represents the URL that user tries to get. |
| duration | t is an integer that represents the duration of the call. It is null for SMS. e.g. 47 |
| call date | it is the date that action performed in a "yyyyMMdd" format. e.g. 20120907 |

For Review Only

from DUAS. A DUAS can contribute (i.e., increase) to the frequency of a pattern only once, even if that pattern occurs more than once in that DUAS, which may occur only for problems that do not include the time information. Matching between a pattern and a DUAS is done in terms of substring matching with some tolerance, whenever it is defined. This is inheritanly enforced by the time dimension whenever it is used by converting exact times into time intervals.

Since the exact activity time for most activities do not have any significance, we have used simple abstraction approach and divided each day into a predefined number of time intervals. If no action has been recorded in a given time interval, then it is dropped from the DUAS. If more than one action is recorded in a given time interval, then the most frequent location is selected as the location information. As a result, DUAS is converted into Daily User Location-Time Sequence, which is defined below:

**Definition (DUS-LT: Daily User Sequence with Location and Time):** $< (L_1, T_1), ...(L_n, T_n) >$ is a daily user location-time sequence obtained from DUAS, in which each $T_i$ corresponds to the beginning of predefined time intervals and each $L_i$ corresponds to the location of the most of the activities occurred for a selected user in that time interval.

The first problem is defined on DUS-LT to determine the location and the time of the next activity of the given user.

**Definition (FDUS-LT: Frequent DUS-LT):** $\{D_1, D_2, ...D_m\}$ is a set of DUS-LT such that each $D_i$ is a frequent DUS-LT generated under some tolerance on time intervals for pre-selected locations.

**Problem 1:** predicting the Location and the Time for the Next Activity in the following time interval (LTNA): In this problem, for a given user sequence, which is a DUS-LT, such as $u =< (L_{u_1}, T_{u_1}),...(L_{u_n}, T_{u_n}) >$ a frequent pattern is searched from FDUS-LT and if a matching (under some tolerance) has been found, the next location-time pair is predicted for that user sequence from the matching frequent pattern.

For the next problem, time information is not used, and DUS-LT are converted into a sequence of locations, which is defined as below:

**Definition (DUS-NL: Daily User Sequence with Non-repeating Location):** $< L_1, L_2, ...L_n >$ is a daily user location sequence obtained from DUS-LT by dropping time attribute and replacing successively repeated locations with a single one. Thus, two successive locations $L_i, L_{i+1} L_i \neq L_{i+1}$.

**Definition (FDUS-NL: Frequent DUS-NL):** $\{D_1, D_2, ...D_m\}$ is a set of DUS-NL such that each $D_i$ is a frequent DUS-NL for pre-selected locations.

The problem is defined below:

**Problem 2:** predicting Location for the first Successive Activity, which has a different location from the current location (LSA): In this problem, for a given user sequence, which is a DUS-NL, such as $u =< L_{u_1}, ...L_{u_n} >$, a frequent pattern is searched from FDUS-NL, and if a matching (under some tolerance) has been found, the next location is predicted for that user sequence from the matching frequent pattern. In the third problem, the location and the time of the first successive activity is predicted which has a different location than from the current location. Therefore DUS-LTs are converted into sequences without successively repeating locations, which is defined below:

**Definition (DUS-NLT: Daily User Sequence with Non-repeating Location and Time):** $< (L_1, T_1), ... >$ is a daily user non-repeating location and time sequence obtained from DUS-LT, in which each $T_i$ corresponds to the beginning of predefined time intervals and $L_i$s correspond to the location of the most of the activities occurred for a selected user in that time interval, and successively repeated locations are replaced by a single one. Thus, two successive locations $L_i, L_{i+1}$ where $L_i \neq L_{i+1}$.

**Definition (FDUS-NLT: Frequent DUS-NLT):** $\{D_1, D_2, ..., D_m\}$ is a set of DUS-NLT such that each $D_i$ is a frequent DUS-NLT generated under some tolerance on time intervals for pre-selected locations. The problem definition is given below:

**Problem 3:** predicting Location and Time for the first Successive Activity, which has a different location from the current location (LTSA): In this problem, for a given user sequence, which is DUS-NLT, such as $u =< (L_{u_1}, T_{u_1}), ...(L_{u_n}, T_{u_n}) >$ a frequent pattern is searched from FDUS-NLT, and if a matching (under some tolerance) has been found, the next location-time pair is predicted for that user sequence from the matching frequent pattern.

## 4. PROPOSED METHODS

In all three problems, since it is almost impossible to predict the exact location of the next activity, we tried to determine the region of the activity. The regions are created by combining the areas corresponding to base stations. Using the location coordinates of base stations, they have been clustered by using k-means method. More than 13000 base stations are clustered in varying number of regions. Then, the CDR data has been processed to replace base station identifiers with their corresponding region identifiers. Due to uneven distribution of the base stations, when we constructed 100 regions, we have obtained clusters containing between 6 and 656 base stations. Figures 1, 2 and 3 show the regions with different zoom levels.

As in almost all big data problems, CDR data also contains a lot of irrelevant information and therefore it has to be preprocessed before it can be utilized. In this preprocessing phase, irrelevant fields such as province code, anonymized callee number etc. are removed and date and time fields are joined together and then all the records are sorted according to caller id in temporal order. Afterwards, for each user, each day's activity is converted into DUAS format, as described in Section 3.2. Table 3 shows a small sample of this process applied to the data shown in Table 2.

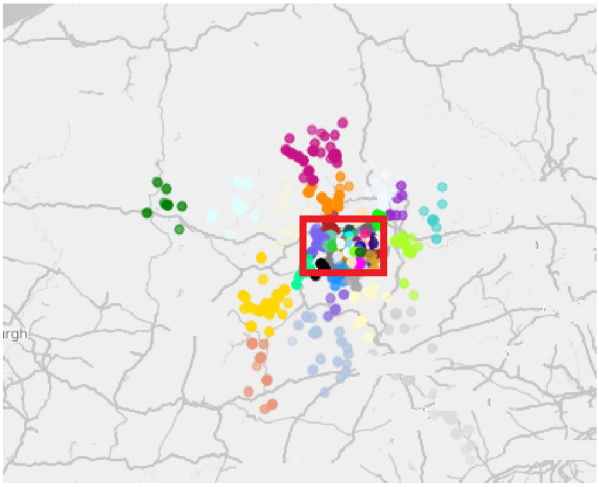


FIGURE 1: Regions in Zoom Level 1

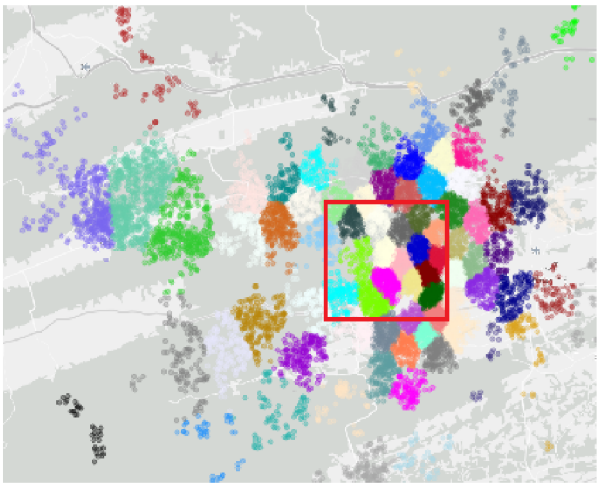


FIGURE 2: Regions in Zoom Level 2

### 4.1. Predicting the Location and the Time for the Next Activity in the following time interval (LTNA)

*4.1.1. Extracting Frequent Patterns*

The main aim of this problem is to predict the common frequent daily navigation patterns of mobile phone users. Therefore, the input of the problem is just a huge set of daily user sequences of region-time interval pairs. The actual day information or even the user identifier is

TABLE 2: Sample CDR data before preprocessing

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| R91 | phone#1 | 06 | R91 | phone#2 | 06 | 2012/09/07 | 01:02:51 | mmo | 47 |
| R91 | phone#1 | 06 | R21 | phone#3 | 06 | 2012/09/07 | 07:10:08 | mmo | 3 |
| R55 | phone#1 | 06 | R27 | phone#4 | 06 | 2012/09/07 | 09:22:31 | mmo | 11 |
| R55 | phone#1 | 06 | R27 | phone#4 | 06 | 2012/09/07 | 11:15:40 | mmo | 8 |
| R55 | phone#1 | 06 | R91 | phone#5 | 06 | 2012/09/07 | 14:43:32 | mmo | 14 |
| R55 | phone#1 | 06 | R3 | phone#6 | 06 | 2012/09/07 | 17:03:04 | mmo | 12 |

TABLE 3: Sample CDR data of Table 2 after preprocessing

| | | | | | |
|---|---|---|---|---|---|
| R91,01:02 | R91,07:10 | R55,09:22 | R55,11:15 | R55,14:43 | R55,17:03 |

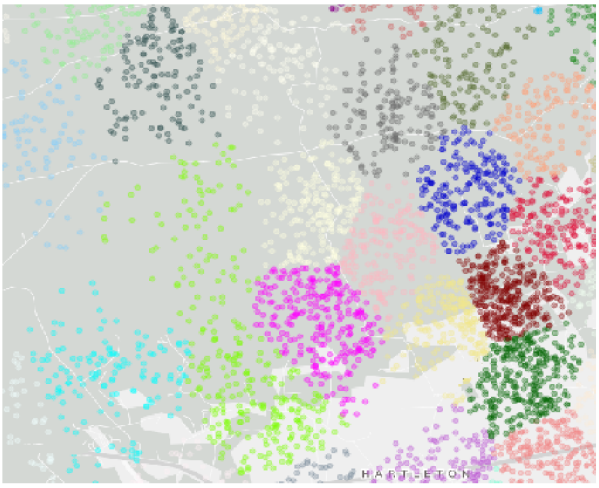


FIGURE 3: Regions in Zoom Level 3

not important. From this set of sequences, we would like to determine frequent patterns. These frequent patterns are determined according to the following parameters:

- pattern length, which describes the length of the desired frequent pattern,
- minimum support, which describes the minimum ratio of the pattern to occur in order to be identified as frequent,
- time interval length, which is used to discretize the time of the day, and defines the length of each interval.

The final parameter has somehow similar effect as defining regions on spatial dimension. Since it is not feasible to process the data with exact times, we have used this parameter to define varying length time intervals to discretize the time dimension. As a result, each day is divided into a predefined number of equivalent length time intervals, and, each exact time data is replaced with the beginning time of the interval it falls into. This may generate sequences of region-time interval pairs with potentially more than one region for the same time interval value. Such pairs are reduced into a single region-time interval pair by choosing the most frequent region for that interval in the sequence.

Our frequent pattern extraction algorithm is a variant of classical AprioriAll algorithm. The standard AprioriAll algorithm consists of two phases, namely; candidate generation and elimination phases. In the standard algorithm, k-length candidates are generated from (k-1)-length patterns. Since this bottom-up process is very costly and since we are only interested in patterns of a given length, this phase has been modified, and our algorithm generates all candidates with predefined fixed length by making a single pass on the data. During this pass the counts of all fixed length patterns are determined, and those that do not satisfy the required minimum support are eliminated. Table 4 depicts a sample with 3 frequent patterns for pattern length 4. In the table, the pairs, constituting region id and the start of the time interval, separated by comma represent region id and discretized time of the day.

#### *4.1.2. Prediction*

The prediction phase works as follows: After a navigation pattern of a current user's region-time interval pair length (k-1) sequence has been obtained, we try to predict the $k^{th}$ region-time interval pair of the user by comparing it with the existing frequent patterns. In order to do this, simply, previously constructed frequent patterns are searched to find if there is a matching between the current user's length (k-1) sequence and the prefixes of length (k-1) of the existing frequent sequences. If such a matching is found, the $k^{th}$ item of the existing sequence can be predicted as the next region-item pair of the current user. Since there can be more than one such predictions, a set of predictions can be generated, which are sorted according to the decreasing support values.

Due to the difficulty of finding exact matches between the current user navigation sequence and existing frequent sequences, we have added a tolerance parameter in time dimension in order to be able to make more flexible predictions. This tolerance parameter allows time intervals to match if they overlap.

TABLE 4: Sample Frequent Patterns

| Frequent Pattern (Sequence) | Support |
|---|---|
| $< (R91, 1000), (R95, 1215), (R45, 1615), (R48, 1800) >$ | $4.02 \times 10^{-6}$ |
| $< (R91, 1000), (R95, 1215), (R45, 1615), (R70, 1900) >$ | $3.68 \times 10^{-6}$ |
| $< (R91, 1000), (R95, 1215), (R45, 1615), (R55, 1915) >$ | $2.53 \times 10^{-6}$ |

Assume that we have a user, with the following navigation sequence:

$$< (R91, 1015), (R95, 1230), (R45, 1630) >$$

However, there is no frequent pattern starting exactly with the same sequence, but we have the following frequent pattern:

$< (R91, 1000), (R95, 1245), (R45, 1630), (R52, 1700) >$

This frequent pattern and the above user navigation pattern have only 15 minutes time difference. We can assume that, the current user's navigation pattern is very similar to this existing frequent pattern, and therefore we can predict the next region-time interval pair of this user as the last pair in the frequent pattern as:

*(R52,1700)*

In order to be able to produce these kind of results, our method uses *time-tolerance* parameter. In this example, it should be set to 15 minutes or larger in order to be able to accept these matchings.

In general, more than one frequent pattern's prefix may match with the current user's navigation pattern. When such case occurs, as a simple solution, the k$^{th}$ pair of the frequent pattern with the highest support value may be returned as a prediction. We may prefer to have more than one prediction in order to increase the accuracy of the prediction. However, it is not feasible to produce a large set of prediction just to increase the accuracy. This trade-off has been handled by our system with the introduction of the multi-prediction limit parameter. This parameter works as follows: All frequent patterns starting with given user's traversal sequence are sorted in decreasing order of the support values and then the sum of the support values are normalized to 1. After that, the prediction set is generated by adding k$^{th}$ elements of frequent patterns one by one in support-sorted order, until sum of the normalized support values reach to the multi prediction limit for the selected sequences. The details are given in the following section.

For example, for the user sequence $< (R91, 1000), (R95, 1215), (R45, 1615) >$ there are three frequent patterns with length 4 as given in Table 4. For this sequence, if the frequent pattern with the highest support value is used to make the prediction, *(R48,1800)*, will be predicted. If the multi prediction limit is set to 0.5, again only the same prediction will be made. However, if the prediction limit is increased to 0.8, then, the first two frequent sequences are going to be used, and two predictions, which are *(R48, 1800)* and *(R70, 1900)*, going to be produced.

### 4.2. Predicting Location for the first Successive Activity, which has a different location from the current location (LSA)

In this method, each record, which is structured as a sequence of region ids, represents a user's daily location change pattern. An example sequence, which is obtained from the sample data given in Table 2 is going to be $< R91, R55 >$.

#### *4.2.1. Extracting Frequent Patterns*

Since in this problem we are interested in the change of the regions of the mobile phone users, the frequent pattern generation phase is slightly different from the first problem, in which all frequent patterns as pairs of regions and time intervals are determined. In this problem, time information is not used and only temporal relations of regions are considered in order to determine frequent user sequences corresponding to region changes. To achieve this, firstly as a preprocessing, for all user sequences, successively repeated regions are eliminated from each daily sequence. After that, standard frequent pattern mining algorithm has been applied on these sequences, and, as a result, frequent patterns corresponding to users's region changes are obtained.

#### *4.2.2. Prediction*

The prediction method used in this problem is very similar to the first problem. Since time information is not used, there is no need for the time-tolerance parameter. Instead, a new tolerance parameter has been introduced in order to be able to match patterns with different lengths, as a simple alignment operation between sequences. We have not used standard alignment algorithms since in our problem the sequences are very short, and therefore, the amount of tolerance needed is very small. As an example, consider that we have a user sequence as follows:

$$< R77, R91, R95, R16, R22, R41 >$$

Although there is no exact matching frequent sequence, let us assume that we have a frequent sequence starting with:

$$< R77, R95, R16, R22, R41 >$$

or

$$< R77, R95, R35, R16, R22, R41 >$$

In this case, we may tolerate one additional region *(R91)* in the user sequence or one additional region *(R35)* in the frequent sequence in the matching process and predict the next region of the frequent pattern as

For Review Only

a potential next region of the user.

Since potentially the lengths of frequent patterns and user sequences are quite small, our tests have shown that except for length tolerances of 1 or 2 the quality of predictions using general alignment methods sharply drops.

### 4.3. Predicting Location and Time for the first Successive Activity, which has a different location from the current location (LTSA)

For this method, user's daily sequence contains not only spatial attribute but also temporal attribute.An example sequence which is obtained from the sample data given in Table 2 will be $< (R91, 01:02), (R55, 09:22) >$.

#### *4.3.1. Extracting Frequent Patterns*

Basic intuition behind the extraction method is nearly the same as that of the first proposed method. In this approach, the patterns are generated in order to keep only the change of region ids in a single day. The difference with the second method is the use of temporal information. This time user's daily sequences have pairs of region id and time information as in the first method. Thus, pairs having the same region id as in the previous pair are eliminated. This guarantees that there will be no successive repetition of region ids in one frequent pattern, and predictions never have the same region id with the last region id of traversal instance.

#### *4.3.2. Prediction*

In this method, we use both tolerance parameters, time tolerance and tolerance in pattern length for prediction. Apart from this difference, the prediction algorithm works similar to the first method.

## 5. EVALUATION AND EXPERIMENTAL RESULTS

### 5.1. Problem Parameters

In the evaluation process we have measured the qualities of the proposed solutions for the three problems by using the following problem parameters:

- *Pattern Length (l):* Defines the length of the patterns that are constructed at the frequent pattern construction phase. During the prediction phase whenever a user pattern of length *(l-1)* has been reached, it is compared against the frequent patterns in order to be able to find matching patterns and then use the last items of those patterns to predict the $l^{th}$ item of the user sequence.
- *Length Tolerance (lt):* Defines the amount of alignment tolerance for matching two patterns. If *lt* is 1 then two sequences with length $n$ and *(n+1)* matches with each other if $n$ of their items are same.
- *Minimum Support Threshold (s):* Defines the minimum number of occurrences of a sequence of a given length in daily user sequences in order to mark that sequence as frequent sequence, which is specified in terms of the percentage of the size of the daily user sequences.
- *Multi Prediction Limit (p):* Defines, in terms of percentages, how to construct prediction set from all frequent patterns that match with the given user sequence using the supports of these frequent patterns, whose values are normalized as the summation of them is 100. This is done as follows: First, all frequent patterns matching with the given users sequence are sorted in decreasing order of the support values, and the sum of their support values are normalized to 100, and the supports of the frequent patterns are also normalized accordingly. After that, the prediction set is populated by choosing the last items of the first $k$ frequent patterns until the normalized summation of the support values of the chosen frequent patterns reach to the specified multi prediction limit $p$.
- *Region/Cluster count (r):* Defines the number of regions/clusters which are generated using the coordinates of the base stations via clustering.
- *Time Interval Length (t):* Defines the length of time intervals in terms of minutes, which is used to divide one day (24 hours) into same size time intervals.
- *Time Tolerance (tt):* Defines the amount of the tolerance time in terms of minutes, that two time parameters can match. For example if both *tt* and $t$ is 15, then an activity that occurred at 13:10 (which is converted to 13:00, after mapping it to start time of the time interval it is in) can match with an activity which occurred in a time interval (12:45-13:15)

Varying values of the above parameters are used in the evaluation of the three problems introduced in the previous sections in order to determine the qualities of the solutions:

- *LTNA:* Only pattern length and minimum support threshold are used.
- *LSA:* Pattern length, minimum support, length tolerance and multi prediction limit parameters are used.
- *LTSA:* All of the above parameters, namely pattern length, length tolerance, minimum support threshold, multi prediction limit, region/cluster count, time interval length and time tolerance parameters are used.

### 5.2. Evaluation Process

In order to assess the quality of the predictions made by the methods proposed in the previous section, we have used 5-fold cross validation on a real CDR data

For Review Only

set that has been introduced earlier. Training phase of the evaluation process consists of applying the frequent pattern extraction steps of the proposed methods on the training data, in order to generate frequent patterns.

The testing phase works as follows: In step one, the test data is processed as in the training phase to extract all sequential patterns, except this time with no minimum support, in order to generate all traversal patterns. For each one of the traversal patterns, prediction algorithm introduced in the previous section has been applied to predict the last elements of these patterns. The result of the prediction is compared against the actual last element of the traversal pattern. These results are used in the calculations of the evaluation metrics which is introduced below.

### 5.3. Evaluation Metrics

In order to measure the qualities of the proposed methods, we have introduced three new metrics, namely p-accuracy, g-accuracy and prediction count.

*g-accuracy* (general accuracy) is the ratio of number of true predictions to the number of all patterns with the same length in the test set.

$$g - accuracy = \frac{|Correctly\ Predicted\ Instances|}{|Test\ Set|}$$

*p-accuracy* (predictions' accuracy) is the ratio of the number of true predictions to the number of all predictions we are able to make.

$$p - accuracy = \frac{|Correctly\ Predicted\ Instances|}{|Predicted\ Instances|}$$

The reason for using two different accuracy calculation is due to the fact that the proposed algorithm may not be able to generate prediction for each one of the test instances, if there is no matching frequent pattern found for the queried instance. In the first form of accuracy calculation, the accuracy result superficially drops for such cases.

In addition to the accuracy, the quality of the results obtained also depends on the size of the prediction set.

*Prediction Count* metric is required because of the multi prediction limit parameter. It quantifies the size of the prediction set when correct prediction result is in the prediction set.

### 5.4. Experimental Results

In our experiments, we fix all the related problem parameters except the one which we measure the effect on the performance.

TABLE 5: Number of Frequent Patterns for Different Pattern Lengths for LTNA

| Pattern Length | Number of Frequent Patterns |
|---|---|
| 2 | 1777423 |
| 3 | 1706778 |
| 4 | 1186798 |
| 5 | 796505 |
| 6 | 539586 |
| 7 | 381818 |
| 8 | 281931 |
| 9 | 214897 |
| 10 | 168218 |
| 11 | 134827 |
| 12 | 110334 |

#### 5.4.1. *Results for Problem 1 (LTNA):*

In the first set of experiments, we analyze the effect of length of the frequent patterns and support threshold using the following parameter values:

- pattern length is 6
- time tolerance is 75 minutes,
- time interval is 15 minutes,
- minimum support is $10^{-6}$,
- cluster count is 100,
- multi prediction support limit is 1.0, (which means allowing to use all frequent patterns matching with test set patterns)

*The Effect of Pattern Length*

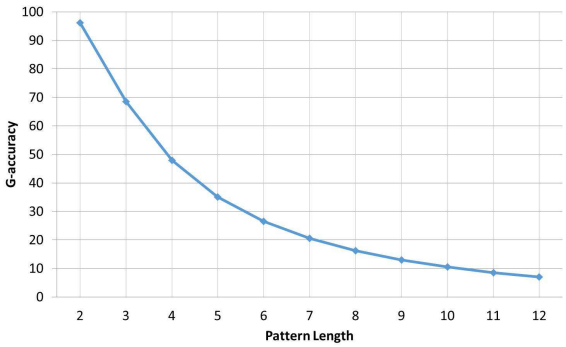



FIGURE 4: The effect of Pattern length on g-accuracy for LTNA

As it can be seen in Figure 4, when the pattern length increases, prediction g-accuracy decreases. This is due to the fact that the number of longer frequent patterns is much fewer than the number of shorter frequent patterns. The number of frequent patterns for various pattern lengths are given in Table 5.

An important observation in this result is that using multi prediction, a very high g-accuracy has been obtained for patterns with length smaller than 5. However, when we have analyzed the number of predictions made with multi prediction method as



For Review

TABLE 6: Prediction Counts for Different Pattern Lengths for LTNA

| Pattern Length | Prediction Count |
|---|---|
| 2 | 59.79 |
| 3 | 11.82 |
| 4 | 6.92 |

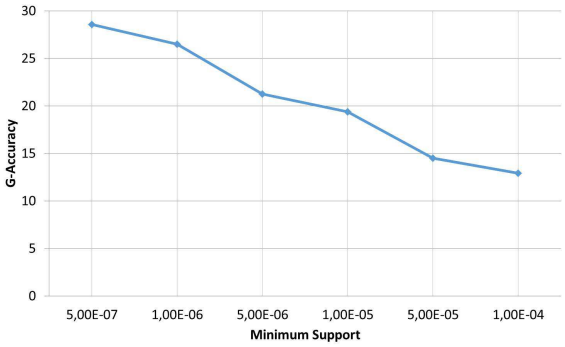



FIGURE 5: The effect of Minimum Support on g-Accuracy for LTNA

a potential next region we have observed that these numbers are quite high as presented in Table 6.

When the total number of regions, which is 100 in our case, are considered, the number of predictions obtained from multi prediction method is not practical and useful for real cases. For example, for length 2, the size of the prediction is almost 60 on average. This explains the superficially high g-accuracy values for patterns shorter than five.

*The Effect of Support Threshold*

Figure 5 shows that, when minimum support threshold value increases, prediction g-accuracy drops. The reason for this result is that as minimum support threshold increases the number of generated frequent patterns decreases.

The most remarkable result that we found in this analysis is the ratio of the number of the patterns (any length n) that have the same region id for $n^{th}$ and $(n-1)^{th}$ time interval to the number of all patterns. It holds for almost 80% of patterns having lengths greater than 4. This causes prediction for test set pattern to be the last element of the matching key in frequent pattern, in other words causes to predict one person's next location as the current location for 80% of the test data. Since our first motivation was change of location problem, we did not evolve this method and do not elaborate on further results of this method.

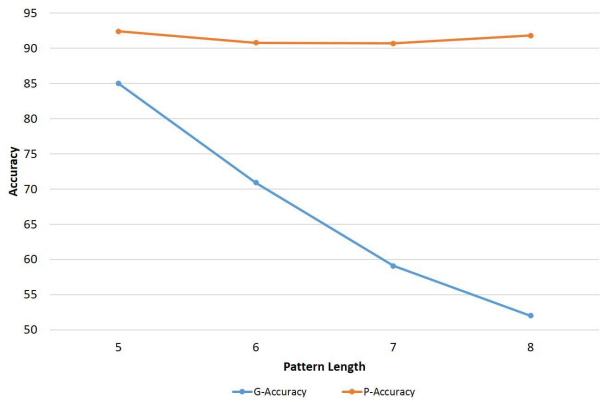
*5.4.2. Results for Problem 2 (LSA):*

In the experiments for this problem, we analyze the effect of the pattern length, support threshold, length tolerance, and the multi prediction limit in terms of accuracy and prediction count using the following values of parameters:

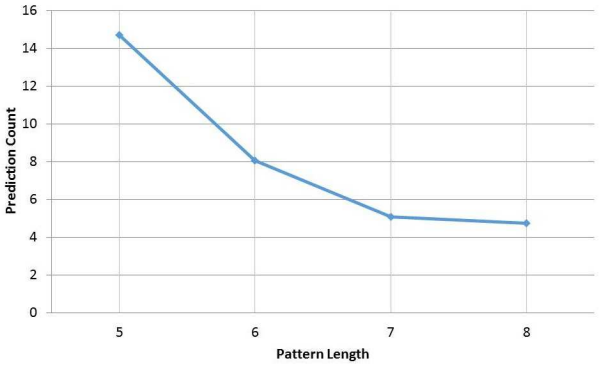
- pattern length is 5
- multi prediction limit is 0.8,
- the length tolerance is 2
- cluster count is 100,
- the minimum support is $4 \times 10^{-7}$.

For the length tolerance experiment, three parameters are set to different values, namely pattern length is set to 7, multi prediction limit is set to 0.5, and the minimum support value is set to 0.0001.

*The Effect of Pattern Length*




(a) Pattern Length vs Accuracy




(b) Pattern Length vs Prediction Count

FIGURE 6: The effect of pattern lenght on g-Accuracy, p-Accuracy and Prediction Count for LSA

As it can be seen in Figure 6a, when the pattern length increases, prediction g-accuracy drops. It is because of the decreasing number of frequent patterns as the pattern length increases. We did not include patterns shorter than 5 since for patterns with length 4, multi prediction method generates 7 alternatives on average. For pattern length 5, our method under multi prediction limit 0.8 generated 2.3 predictions on average for successful prediction, which is reasonable value for the number of generated predictions. Figure

TABLE 7: Length Tolerance vs g-Accuracy for LSA

| Length Tolerance | g-Accuracy |
|---|---|
| 0 | 0.20 |
| 1 | 0.23 |
| 2 | 0.29 |

6a also shows the relationship between pattern length and p-accuracy. Since p-accuracy is the ratio of true predictions to the number of predictions made (instead of the total number of test patterns), it is not expected to have a similar behavior when pattern length increases. The reason for the lower g-accuracies of higher pattern lengths in the Figure 6a is the non-predicted instances in test data. However, we do not include non-predicted patterns in p-accuracy.

Prediction count has been positively affected with the increase in pattern lengths, as can be seen in Figure 6b. After quick drop of prediction count at pattern length 7, p-accuracy starts to increase. It is expected to have greater p-accuracy for the longer patterns with nearly the same prediction count.

*The Effect of Support Threshold*

Figure 7a shows that when minimum support value increases, prediction g-accuracy drops as in our first problem. Similarly, this is due to the fact that as the minimum support increases, the number of generated frequent patterns decreases.

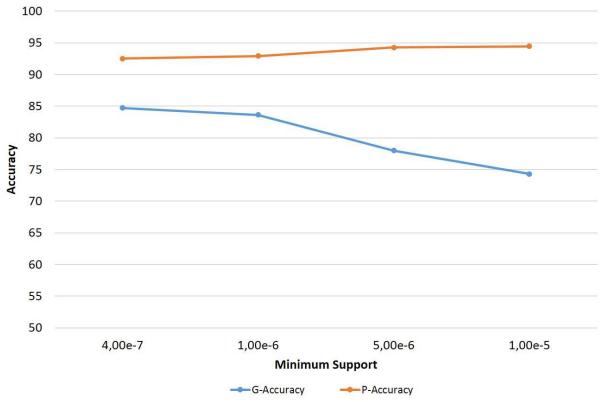
When compared to the first problem, it can be seen that g-accuracy values are much higher in the second problem. There are two reasons for it; length tolerance and eliminating successively repetitive region ids. Length tolerance gives the ability to search test set pattern throughout different lengths of frequent patterns. Eliminating repetitive region ids gives less variety in frequent patterns. These factors reduce the number of non-predicted patterns as expected (from 2,214,700 to 1,237,313), and increment both true and false predictions biased to true predictions.

Similar to the previous experiments, as can be seen in Figure 7a, p-accuracy does not seem to be effected with the increase in the support value, and there is a very small increase.

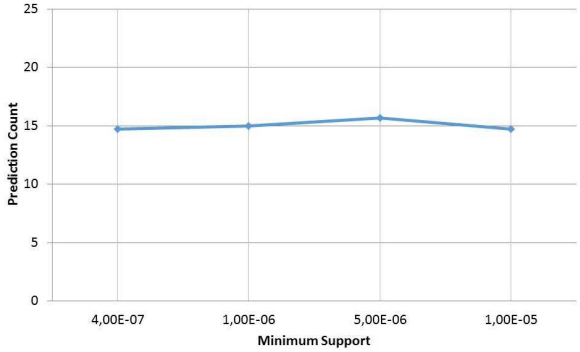
For the selected parameter set, prediction count is stable. This experiment shows that multi prediction limit outweighs the effect of minimum support on prediction count.

*The Effect of Length Tolerance*

As given in Table 7, g-accuracy values are lower than the first problem, since minimum support used in this set of experiments is 0.0001. As it can be seen in the table, when the length tolerance increases, prediction g-accuracy also increases.




(a) Minimum Support vs Accuracy




(b) Minimum Support vs Prediction Count

FIGURE 7: The effect of Support Threshold on g-Accuracy, p-Accuracy and Prediction Count for LSA

*The Effect of Multi Prediction Limit*

Figure 8a depicts that when multi prediction limit increases, both prediction g-accuracy and p-accuracy also increase, as expected. Nevertheless, Figure 8b shows that, with the increase in multi prediction limit prediction count also increases.

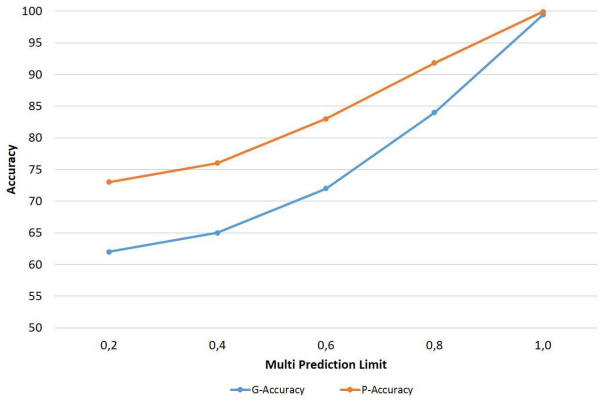
*5.4.3. Results Problem 3 (LTSA):*

For this problem, we analyze the effect of all 7 paramaters on the accuracy and prediction count using the following values of parameters:

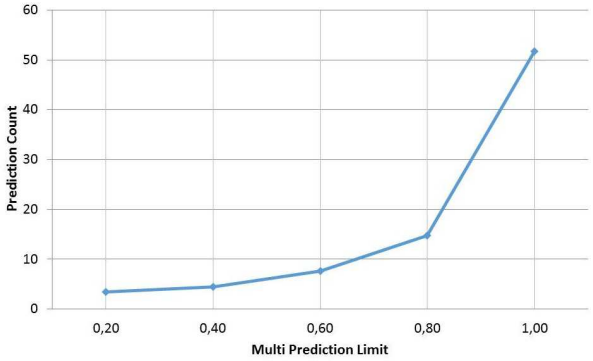
- pattern length is 5
- length tolerance is 2,
- time interval length is 60,
- time tolerance is 120,
- multi prediction limit is 0.8,
- cluster count is 100,
- minimum support is $4 \times 10^{-7}$.

In the time interval experiment, time tolerance is set to 0.

For Review Only

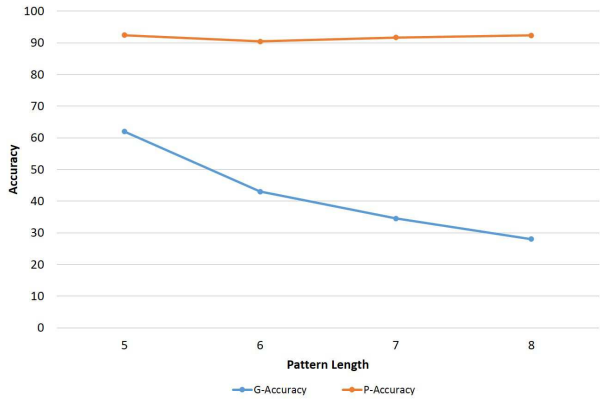



(a) Multi Prediction Limit vs Accuracy

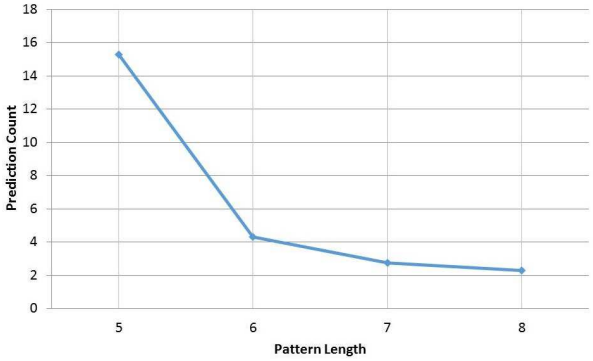



(b) Multi Prediction Limit vs Prediction Count

FIGURE 8: The effect of multi prediction limit on g-Accuracy, p-Accuracy, Prediction Count for LSA




(a) Pattern Length vs Accuracy




(b) Pattern Length vs Prediction Count

FIGURE 9: The Effect of pattern length on g-Accuracy, p-Accuracy and Prediction Count for LTSA

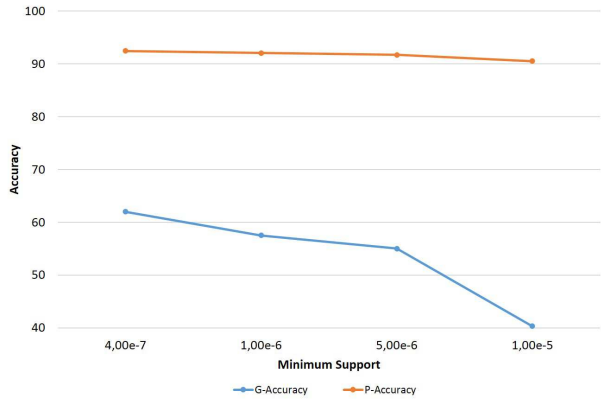
*The Effect of Pattern Length*
Figure 9a shows that when pattern length increases similar to two previous problems, g-accuracy decreases, and p-accuracy is almost stable. On the other hand, the drop in prediction count is remarkable. It drops to 4, for pattern length 6, and then quickly drops almost to 2 for the pattern length 7.

*The Effect of Support Threshold*
We have observed similar behaviours for the minimum support paramater, and as it can be seen in Figure 10a, when minimum support value increases, g-accuracy decreases. On the other hand, similar to previous problems, p-accuracy is almost stable. Furthermore, Figure 10b also shows that when minimum support value increases, prediction count also decreases, as expected.

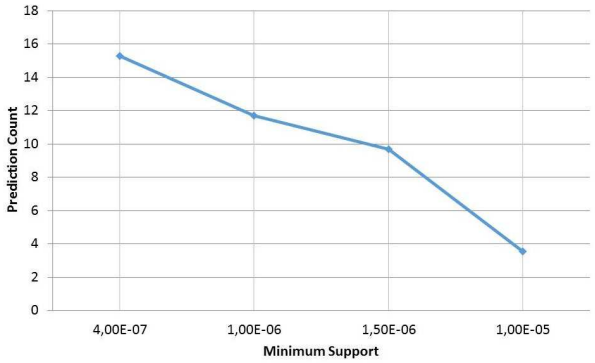
*The Effect of Length Tolerance*
As it can be seen in Figure 11a, when length tolerance increases, both g-accuracy and p-accuracy increase. Increasing length tolerance makes some unpredicted test sequences predictable which increases the g-accuracy. Unfortunately, as it can be seen in Figure 11b, setting higher length tolerance leads to larger prediction sets.

*The Effect of Multi Prediction Limit*
The increase in multi prediction limit increases both accuracy results, as seen in Figure 12a, as in previous problem, while increasing the prediction count very quickly also, shown in Figure 12b.

*The Effect of Number of Regions*
In this problem, we have also analysed the effect of the region sizes. As it can be seen in Figure 13a, when cluster count, i.e, number of base station regions, increases g-accuracy decreases slightly. It is because of the unpredicted test sequences rather than false predictions since increasing cluster count makes frequent patterns harder to extract. However, Figure 13a also shows that, p-accuracy increases slightly. This is due to the fact that when cluster count increases movement patterns of users can be defined more precisely which makes frequent patterns harder to find but more accurate ones. Therefore, usually correct predictions are generated when compared to fewer numbers of clusters. It also eventually decreases the

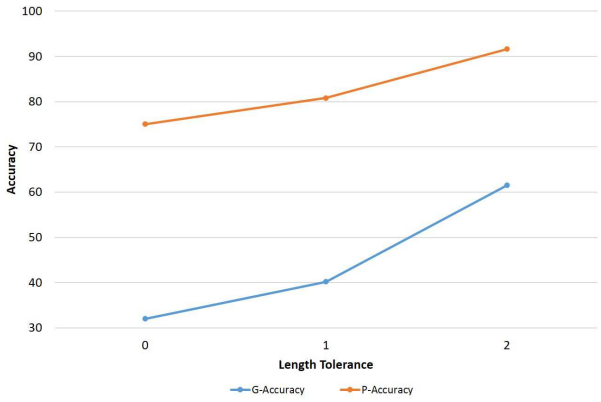



(a) Minimum Support vs Accuracy

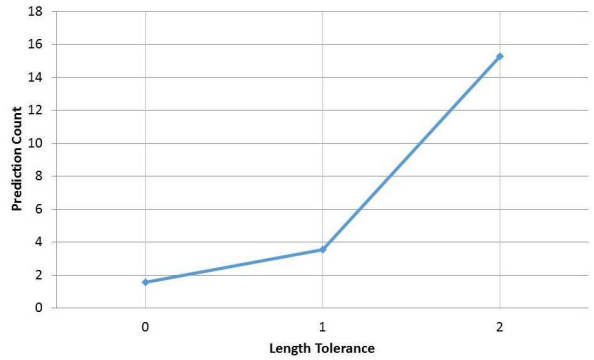



(b) Minimum Support vs Prediction Count

FIGURE 10: The effect of support threshold on g-Accuracy, p-Accuracy and Prediction Count for LTSA




(a) Length Tolerance vs Accuracy




(b) Length Tolerance vs Prediction Count

FIGURE 11: The effect of length tolerance on g-Accuracy, p-Accuracy and Prediction Count for LTSA

prediction count, which can be seen in Figure 13b.

*The Effect of Time Interval Length*

When time interval length increases, both g-accuracy and p-accuracy increase, as shown in Figure 14a. Although there is a sharp increase in g-accuracy, the increase in p-accuracy is limited. Since the larger time interval means the more similar daily sequences and eventually higher number of frequent patterns, increase in the values of accuracy metrics is expected. We can say that prediction count increases in general, as it can be seen in Figure 14b, the time interval length increases. However, for time interval length 360, there is small drop.

*The Effect of Time Tolerance*

Similarly, when time tolerance increases both g-accuracy and p-accuracy increases slightly, as it can be seen in Figure 15a. Moreover, also Figure 15b shows that, when time tolerance increases prediction count slightly decreases. However, for this set of parameters the prediction count is very large.

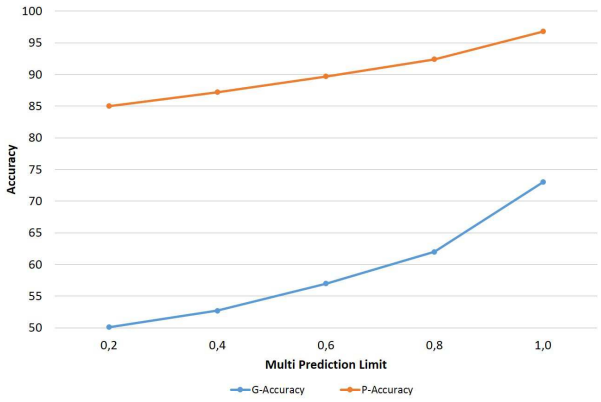
## 6. DISCUSSION AND CONCLUSION

In this work, we applied sequence pattern mining techniques for location prediction problem domain. We used one of the largest mobile phone operator companies' CDR data. We focused on three different subproblems in the location prediction problem space namely, next location and time prediction using spatio-temporal data, next location change prediction using spatial data, next location change and time prediction using spatio-temporal data. The main novelties are time prediction and spatio-temporal alignments for the prediction task. In the experiments, we have evaluated our model's prediction quality with respect to g-accuracy, p-accuracy and prediction count and further analyzed the effects of change of minimum support, multi prediction limit, length tolerance, pattern length, cluster count, time interval length and time tolerance on prediction accuracies and count. Here are the some basic findings and most valuable prediction results for these three methods;

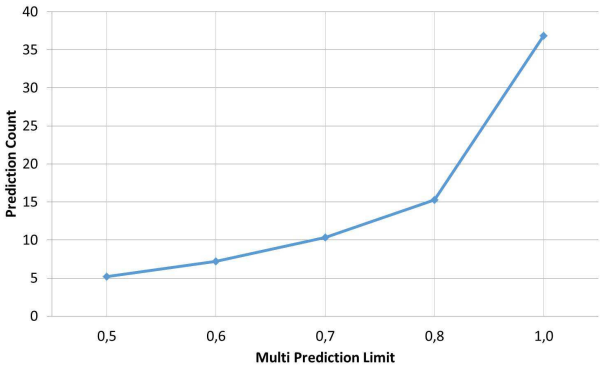
- For the spatio-temporal next location prediction, it does not make sense to present the results below or around 80% accuracy since 80% of the user's next



For Review Only

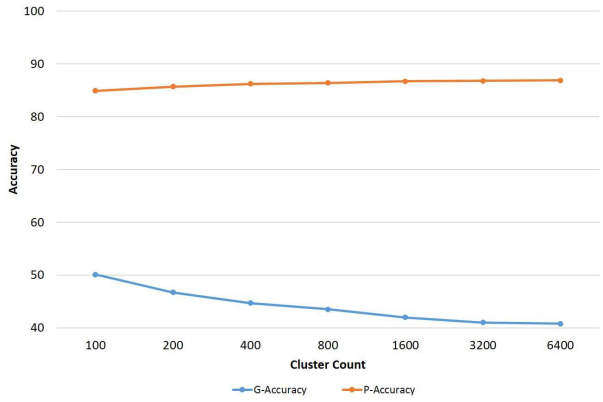



(a) Multi Prediction Limit vs Accuracy

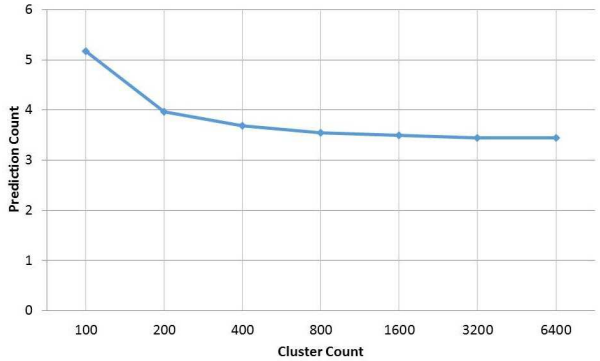



(b) Multi Prediction Limit vs Prediction Count

FIGURE 12: The effect of multi prediction limit on g-Accuracy, p-Accuracy and Prediction Count for LTSA




(a) Cluster Count vs Accuracy




(b) Cluster Count vs Prediction Count

FIGURE 13: The effect of number of regions on g-Accuracy, p-Accuracy and Prediction Count for LTSA

location is their current location.

- For the spatial next location change prediction, g-accuracies differ between 48% and 84% for the prediction counts 2.4 and 14 for 100 regions while p-accuracies differ between 74% and 99% for the same prediction counts. These values show that our proposed model for this problem can generate successful accuracy values with acceptable prediction counts.
- For the spatio-temporal next location change and time prediction, while it predicts nearly half of the test sequences, p-accuracies reach up to 93% for 14 prediction count for possible 9600 ([24 x 1 hour time interval] x 400 clusters) spatio-temporal prediction combination. Moreover it generates 87% p-accuracy for 3.44 prediction count for possible 153600 ([24 x 1 hour time interval] x 6400 clusters) prediction combination.

As a future work, we plan to enlarge our problem space with the followings; next location change prediction using spatio-temporal data, next action time prediction using temporal data, location and time prediction of the next action using spatio-temporal data.

## 7. ACKNOWLEDGEMENT

This research was supported by Ministry of Science, Industry and Technology of Turkey with project number 01256.STZ.2012-1 and title "Predicting Mobile Phone Users' Movement Profiles".

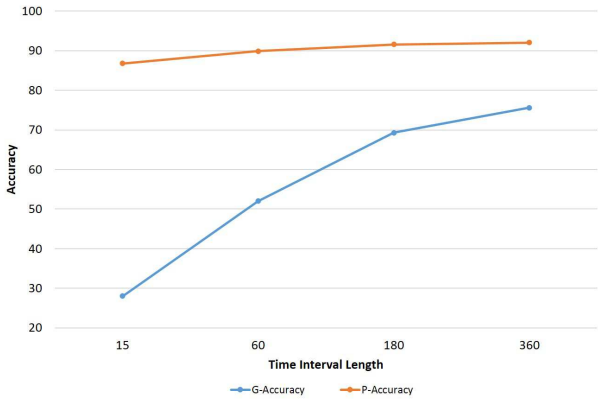
## REFERENCES

[1] Tseng, V. S. and Lin, K. W. (2006) Efficient mining and prediction of user behavior patterns in mobile web systems. *Information and Software Technology*, **48**, 357 – 369.

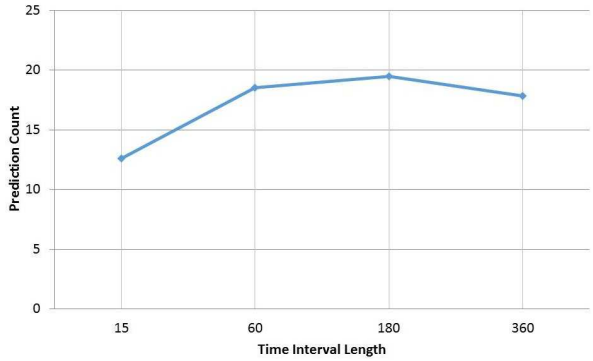
[2] Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. (2008) Understanding individual human mobility patterns. *Nature*, **453**, 779–782.

[3] Phithakkitnukoon, S., Horanont, T., Di Lorenzo, G., Shibasaki, R., and Ratti, C. (2010) Activity-aware map: Identifying human daily activity pattern using mobile phone data. In Salah, A., Gevers, T., Sebe, N., and Vinciarelli, A. (eds.), *Human Behavior Understanding*, Lecture Notes in Computer Science, **6219**, pp. 14–25. Springer Berlin Heidelberg.

[4] Zhu, Y., Zhang, Y., Shang, W., Zhou, J., and Ying, C. (2009) Trajectory enabled service support platform for mobile users' behavior pattern mining.

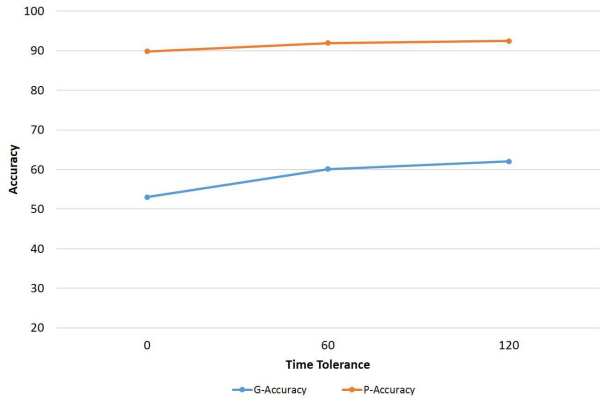



(a) Time Interval Length vs Accuracy

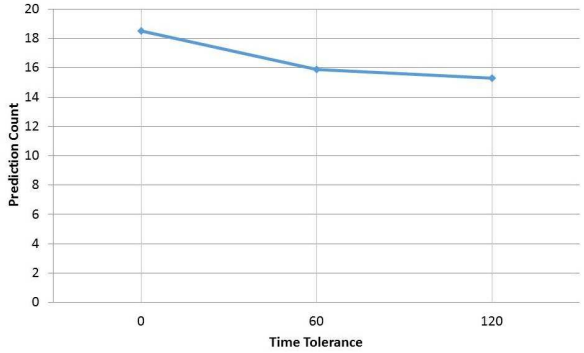



(b) Time Interval Length vs Prediction Count

FIGURE 14: The Effect of time interval length on g-Accuracy, p-Accuracy and Prediction Count for LTSA




(a) Time Tolerance vs Accuracy




(b) Time Tolerance vs Prediction Count

FIGURE 15: The effect of time tolerance on g-Accuracy, p-Accuracy and prediction count for LTSA

*Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, July, pp. 1–10.

[5] Ozer, M., Keles, I., Toroslu, I. H., and Karagoz, P. (2013) Predicting the change of location of mobile phone users. *Proceedings of the Second ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, New York, NY, USA MobiGIS '13, pp. 43–50. ACM.

[6] Ozer, M., Keles, I., Toroslu, I. H., Karagoz, P., and Ergut, S. (2014) Predicting the next location change and time of change for mobile phone users. *Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, New York, NY, USA MobiGIS '14, pp. 51–59. ACM.

[7] Cho, E., Myers, S. A., and Leskovec, J. (2011) Friendship and mobility: user movement in location-based social networks. *KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1082–1090. ACM.

[8] Boldrini, C. and Passarella, A. (2010) Hcmm: Modelling spatial and temporal properties of human mobility driven by users social relationships. *Computer Communications*, **33**, 1056–1074.

[9] Zhang, D., Vasilakos, A. V., and Xiong, H. (2012) Predicting location using mobile phone calls. *SIGCOMM Comput. Commun. Rev.*, **42**, 295–296.

[10] Thanh, N. and Phuong, T. M. (2007) A gaussian mixture model for mobile location prediction. *The 9th International Conference on Advanced Communication Technology*, **2**, 914 – 919.

[11] Keles, I., Ozer, M., Toroslu, I. H., and Karagoz, P. (2014) Location prediction of mobile phone users using apriori-based sequence mining with multiple support thresholds. *Proceedings of the 3rd Workshop on New Frontiers in Mining Complex Patterns* NFMCP 2014, pp. 2–13.

[12] Gao, H., Tang, J., and Liu, H. (2012) Mobile location prediction in spatio-temporal context. *the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing*, June. Nokia.

[13] Gidófalvi, G. and Dong, F. (2012) When and where next: individual mobility prediction. *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, New York, NY, USA MobiGIS '12, pp. 57–64. ACM.

[14] de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013) Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, **3**.

[15] Song, C., Qu, Z., Blumm, N., and Barabsi, A.-L. (2010) Limits of predictability in human mobility. *Science*,

**327**, 1018–1021.

[16] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009) Mining interesting locations and travel sequences from gps trajectories. *Proceedings of the 18th International Conference on World Wide Web*, New York, NY, USA WWW '09, pp. 791–800. ACM.

[17] Ying, J. J.-C., Lee, W.-C., Weng, T.-C., and Tseng, V. S. (2011) Semantic trajectory mining for location prediction. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA GIS '11, pp. 34–43. ACM.

[18] Yavas, G., Katsaros, D., Ulusoy, O., and Manolopoulos, Y. (2005) A data mining approach for location prediction in mobile environments. *Data Knowl. Eng.*, **54**, 121–146.

[19] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. (2007) Trajectory pattern mining. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA KDD '07, pp. 330–339. ACM.

[20] Cao, H., Mamoulis, N., and Cheung, D. W. (2007) Discovery of periodic patterns in spatiotemporal sequences. *IEEE Trans. on Knowl. and Data Eng.*, **19**, 453–467.

[21] Candia, J., Gonzlez, M., Wang, P., Schoenharl, T., Madey, G., and Barabasi, A.-L. (2008) Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical*, **41**, 1–11.

[22] Ryder, J., Longstaff, B., Reddy, S., and Estrin, D. (2009) Ambulation: A tool for monitoring mobility patterns over time using mobile phones. *Computational Science and Engineering, 2009. CSE '09. International Conference on*, Aug, pp. 927–931.

[23] Zheng, J. and Ni, L. M. (2012) An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 153–162. ACM.

For Review Only