PREDICTING THE LOCATION AND TIME OF MOBILE PHONE USERS BY USING SEQUENTIAL PATTERN MINING TECHNIQUES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERT ÖZER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2014

Approval of the thesis:

PREDICTING THE LOCATION AND TIME OF MOBILE PHONE USERS BY USING SEQUENTIAL PATTERN MINING TECHNIQUES

submitted by **MERT ÖZER** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _______________

Assoc. Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering Department, METU** _______________

Prof. Dr. İ. Hakkı Toroslu
Co-supervisor, **Computer Engineering Department, METU** _______________

Examining Committee Members:

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _______________

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU _______________

Assoc. Prof. Dr. Halit Oğuztüzün
Computer Engineering Department, METU _______________

Assoc. Prof. Dr. Osman Abul
Computer Engineering Department, TOBB ETU _______________

Dr. Cevat Şener
Computer Engineering Department, METU _______________

Date: _______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MERT ÖZER

Signature :

ABSTRACT

PREDICTING THE LOCATION AND TIME OF MOBILE PHONE USERS BY USING SEQUENTIAL PATTERN MINING TECHNIQUES

Özer, Mert
M.S., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Pınar Karagöz
Co-Supervisor : Prof. Dr. İ. Hakkı Toroslu

July 2014, 63 pages

Predicting the location of people from their mobile phone logs has become an active research area. Due to two main reasons this problem is very challenging: the log data is very large and there is a variety of granularity levels both for specifying the spatial and the temporal attributes, especially with low granularity level it becomes much more complicated to define common user behaviour patterns. For the location prediction problem domain, we focused on 3 sub-problems and proposed 3 different methods for these problems. The idea in all of the three methods follows these two steps; cluster the spatial data into the regions and group temporal data into the time intervals to get higher granularity level, and apply sequential pattern mining techniques to extract frequent movement patterns to predict accordingly. We have validated our results with real data obtained from one of the largest mobile phone operators in Turkey. Our results are very encouraging, and we have obtained very high accuracy results in predicting the location of mobile phone users.

Keywords: Location Prediction, Mobile Phone Users, Sequential Pattern Mining, AprioriAll Algorithm

ÖZ

MOBİL TELEFON KULLANICILARININ SIRALI ÖRÜNTÜ MADENCİLİĞİ TEKNİKLERİ İLE KONUM VE ZAMAN TAHMİNİ

Özer, Mert
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Pınar Karagöz
Ortak Tez Yöneticisi : Prof. Dr. İ. Hakkı Toroslu

Temmuz 2014 , 63 sayfa

Telefon kullanım kayıtlarından insanların konumlarının tahmini aktif bir araştırma alanı haline gelmiştir. Kullanım kayıtlarının büyüklüğü ve mekansal ve zamansal bilgilerin oldukça farklı tanecik seviyelerinde incelenebilir olması bu problemin zorlaşmasının iki ana sebebini oluşturur; özellikle küçük tanecik seviyelerinde kullanıcıların ortak davranış örüntülerini çıkarmak çok daha zorlaşır. Konum tahmini problemi alanı için 3 tane alt problem tanımladık ve bu problemler için 3 farklı metod önerdik. Bütün metodlardaki temel düşünce şu iki adımı takip eder; konum bilgisini daha büyük alanlara grupla ve zaman bilgisini daha büyük zaman aralıklarına grupla ve daha sonra sıralı örüntü madenciliği yöntemleri uygulayarak sonuçlara göre konum tahmininde bulun. Sonuçlarımızı Türkiye'nin en büyük mobil operatörlerinden birinden alınan gerçek veriler ile doğruladık. Sonuçlarımız oldukça cesaret verici ve cep telefonu kullanıcılarının konumlarının tahminlerinde çok yüksek doğruluk değerleri elde ettik.

Anahtar Kelimeler: Yer Tahmini, Cep Telefonu Kullanıcıları, Sıralı Örüntü Madenciliği, AprioriAll Algoritması

To the beauty of ruins

ACKNOWLEDGMENTS

I would like to thank my supervisor Associate Professor Pınar Karagöz and co-supervisor Associate Professor İ. Hakkı Toroslu for their constant support, guidance and friendship and I also would like to thank them for listening my personal problems as well. It was a great honour and pleasure to work with them for the last two years and our cooperation influenced my academical view and my self confidence highly.

I want to thank my colleague İlkcan Keleş for his incredible effort and understanding during our co-working experience. Without him, this work would be finished much harder than it has been.

I want to thank my parents Bahriye and Tayfur and my sister Göknil for their constant support and understanding. Without knowing that they are always there for me, it would be very hard to overcome the problems that I faced during my graduate studies.

There are lots of friends that do not participate with me in this work but support me generously during my undergraduate and graduate studies. This work has been done by the help of them as well. It is great to have them in my life, Hanen Çiftdoğan, Utku Şirin, Fikret Kurt, Pınar Şen, Bahadır Eken, Gülşah Çolak, Deren Atlı, Hasan Güntürkün, Hüsnü Zeybek, Barış Uğurlu and Behlül Uçar. This work is also supported by Ministry of Science, Industry and Technology of Turkey with project number 01256.STZ.2012-1.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CDR	Call Detail Record

CHAPTER 1

INTRODUCTION

In today's world, mobile phones are commonly used devices. Basic usage information including base station, call records, short message records, GPS records are logged and used by mobile phone operator companies for various purposes. One of these purposes is location prediction which helps companies to model their users' daily life behaviour. By modelling behaviour of their users, companies build more reasonable advertisement strategies. These results may also be used by city administrators to determine mass people movement patterns (in terms of location and time) around the city.

User location prediction can be studied in terms of different levels of granularities. Determining the exact coordinate and the time of the next location of a person is almost impossible. Mobile phones are usually attached to the nearest base stations. Therefore, each base station coordinate can be considered as the center of a region, and location prediction can be made at this granularity level. In densely populated city centers, these regions will be very small and in rural areas they will be very large. Also due to large number of base stations in densely populated areas, the movements of people will correspond to jumping over many areas because the number of records during the movement potentially will not be large enough to have records for each region that have been passed. Therefore, it is worth to cluster base stations using their coordinates to define fewer number of regions. Moreover patterns involving high number of very small regions are not suitable for interpreting mass people movements in an urban area.

In this work we have empirically set the number of regions to 100 after trying some larger and smaller values. With this many regions we had small number of jump overs and still the area sizes of the regions became small enough to capture the details of the people movements. Also notice that by clustering we have obtained regions with sizes closer to each other. Moreover different number of regions are analyzed as well. In this work Call Detail Record (CDR) data obtained from one of the largest mobile phone operators in Turkey has been used. A quick analysis of our data shows that more than 80% of users' next location is their current location. Only 20% of the data contains different locations between two consecutive records of each user. Therefore, although we present the results for next location prediction here, it makes more sense to predict this change rather than predicting the next location, which will be the same one for 80% of data. This idea comprises our second problem definition and second proposed method; next location change prediction using spatial data.

These two approach are based on constructing the regions by clustering the base stations and then applying sequence pattern mining techniques. In order to realize this, we follow four phases, which are preprocessing the data, clustering base stations, extracting sequence patterns mining methods and predicting the change of location for mobile phone user.

User location change prediction can also be studied with different conducive attributes to increase accuracy rates such as temporal attributes. As a third problem definition, we introduced the next location change prediction using spatio-temporal data. In this domain, we both used historical temporal information and predicted the time of the next location change. Experiments shows that spatio-temporal sequence mining gives more valuable prediction accuracies rather than simply using spatial attribute. Moreover alignments on spatial and temporal attribute of the sequences augment the probability of pattern matching. All three solutions embraces the basics of Apriori-All algorithm. The experiments show that the methods we proposed generates both concrete and complete location predictions.

CHAPTER 2

RELATED WORK

In this chapter, we give information about the previous works that deal with the problem of location prediction. We also summarize various aspects of each technique. In recent years, a variety of location prediction schemes and scientific findings about human mobility have been presented in [5], [3], [21], [17], [7], [9], [6], [15], [22], [20], [19], [8], [4].

Some findings about the human mobility habits and its predictability are presented in [6] and [15]. In [6], Montjoye et al. proposes 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 propose that four randomly chosen points are enough to characterize 95% of the users while two of them characterize more than 50%.

In [15], Song et al. analyze the limits of predictability in human mobility. They used the data collected from mobile phone carriers for 3-month-long of 50,000 individuals. They propose three entropy measures which is believed to be the most fundamental quantity 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 predicted user's future movements. They find that 93% potential predictability in user mobility at best and it is not under 80% for any user.

There are other methods to use for location prediction problem rather than sequential pattern mining such as markov models and expectation maximization algorithms. In [17], 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 [7], Gao et al. use both spatial and temporal data to predict users' location. They propose 10 models which can be categorized as spatial-based, temporal-based and spatio-temporal. 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 2 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 [9], Gidofalvi et al., proposes 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 when this change occurs. 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 predict the departure time correctly with the 45 minute error and the next region correctly 67% of the cases.

Similar to our work, in [19], [8] and [4], they propose sequential pattern mining techniques for the location prediction problem. In [19], 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 predicting. The rules are 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 claim that they get higher accuracies than the methods of Mobility Prediction based on Transition Matrix and Ignorant Prediction.

In [8], 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 which is named as ROI. The difference with the conventional sequence pattern mining technique is the use of trajectories (T-patterns) rather than itemsets. Their method for mining T-patterns extract both computationally feasible and useful patterns.

In [4], Cao et al. introduces a method for discovering of periodic patterns in spatiotemporal 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 try to further increase the prediction accuracies by the help of social networks. In [5], Cho et al. proposes that general human mobility do 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 source 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 develop an expectation maximization based prediction model and they reach 40% accuracy while predicting user's location at any time.

In [3], Boldrini et al. propose a model that integrates 3 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 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 [21], Zhang et al. further improves the user mobility models of [3] and [5] 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 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 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 [22] and [20] they make use of the distinctive features of spatial attribute in the data. In [22], 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 (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 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 [20], Ying et al. proposes 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 excellent performance.

CHAPTER 3

BACKGROUND

In this chapter, basics of conventional algorithms used in this work are introduced. In the first section, definition of clustering and one clustering method namely k-means are presented. In the second section, the definition of sequential pattern mining and the AprioriAll algorithm are presented.

3.1 Clustering

Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering [16]. Clustering methods can be categorized according to their two attributes; nested or not and exclusive or overlapping or fuzzy. We preferred a non-nested and exclusive clustering method for clustering the base stations since we needed each base station to be the member of only one cluster and did not need any hierarchical connection between clusters.

3.1.1 K-Means Algorithm

K-Means Algorithm is a non-nested and exclusive clustering method which embraces the idea of grouping similar objects into same clusters and non-similar objects into different clusters. It was the first time when its name was used in [12] while the more efficient version of it is introduced in [11]. Formally, given a data set, D, of n objects, and k, the number of clusters to form, k-means algorithm organizes the objects into k partitions ($k <= n$), where each partition represents a cluster[10]. The number of partitions k is expected to be defined by the user. Partitioning is done according to the centroids of the clusters. Each data object is assigned to the nearest cluster while the concept of nearness can be defined using several distance metrics such as Euclidian, Manhattan, Chebychev distances etc. Distance metric is chosen according to the problem definition.

Algorithm can be divided into two steps; data assignment and relocation of centroids. After every data object assignment to the partitions are completed, new centroids are found by computing the means of the elements of that particular partition. This process is iterated recursively until the members of partitions do not change or some user defined condition is satisfied.

Algorithm 1 K-Means Algorithm

Input: Dataset D, number of clusters k

Output: Set of cluster centroids C, cluster membership vector m

```
function KMEANS($D$, $k$, $C$, $m$)
    Randomly choose $k$ data points from $D$
    Use these $k$ points as initial set of cluster representatives $C$
    repeat
        Reassign points in $D$ to closest cluster mean
        Update $m$ such that $m_i$ is cluster id of $i^{th}$ point in $D$
        Update $C$ such that $c_j$ is the mean of points in $j^{th}$ cluster
    until convergence of objective function
end function
```

Usually, algorithm's objective function is to minimize the total squared Euclidean distance between each point and its closest centroid. It can be formulated as follows where x_i represents the i^{th} data object and c_j represents the centroid of the j^{th} cluster;

$$\sum_{i=1}^{k} \operatorname*{argmin}_{j} ||x_i - c_j||_2^2 \tag{3.1}$$

Drawbacks of the algorithm can be summarized in 4 aspects. First, because of the greedy nature of the algorithm, user defined initial centroids matter. In other words, there is no one correct clustering output for the algorithm, it is affected by the initial centroids. Second, choosing k can be difficult since the default algorithm do not generate or suggest any optimal number for how many clusters there should be. Third, standard algorithm is sensitive to outliers and last, it does not guarantee against empty clusters.

3.2 Sequential Pattern Mining

The concept of sequential pattern mining is first introduced by Agrawal and Sirikant in [2] as follows; Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified minimum support threshold, sequential pattern mining is to find all frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than minimum support.

In the spatio-temporal context, sequential pattern mining can be expressed as follows similar to Agrawal and Sirikant's definition. Given a set of sequences, where each sequence consists of a list of elements and each element consists of a spatial and temporal attributes, and given a user-specified minimum support threshold, spatio-temporal sequential pattern mining is to find all frequent time ordered movement pattern subsequences. In general, sequence of k elements is denoted in a form such as $s =< s_1, s_2, s_3, ..., s_k >$. A sequence s_1 is subsequence of s_2 if and only if all elements of s_1 is contained in s_2 in the same order. The concept of minimum support is the same as in the conventional itemset problems. Support of a sequence is the ratio of the number of the occurence of the sequence in the whole database to the total number of sequences with the same length in the whole database. A sequence satisfying the minimum support constraint is called a frequent sequence [18] or a large/maximal sequence. A sequence containing k elements is represented by k-sequence.

3.2.1 AprioriAll Algorithm

AprioriAll is a sequential pattern mining algorithm first introduced in [2] after introducing Apriori algorithm in [1] which constitutes a base for the AprioriAll. It is designed for extracting maximal sequences from a database. It consists of five phases, namely sort phase, fitemset (frequent itemset) phase, transformation phase, sequence phase and maximal phase. The main phase is the sequence phase while first three phase can be considered as a preprocessing phases and the last phase as a postprocessing phase.

In the first phase, database D is modified by taking sequence id and transaction time into consideration. Transaction time is used for creating time ordered sequences. Sequence id is used for making elements with same id appear in the same sequence. This phase is needed for the sake of convenience of the following phases.

In the second phase, the set of all large 1-sequences are extracted. In this phase, all fitemsets can be obtained by using conventional Apriori algorithm with the relevant modifications in counting and support. These fitemsets are mapped to ordinal integers so that comparing two fitemsets takes constant time.

In the third phase, each database entry or in other words transaction is modified such that the elements that are not the member of any fitemset are eliminated in that transaction. If there exist no element in transaction after elimination, it is not retained in the transformed database D_T anymore. However it is still used for counting the total number of sequences.

In the fourth phase, new candidate sequences are generated. The candidates are generated by using the previously generated fitemsets or maximal sequences. To generate k-sequence candidates algorithm uses (k-1)-sequences. It basically joins the (k-1)-sequences to find candidate k-sequences. After each candidate generation, algorithm counts the occurrences of the candidates in the database. This information is used for eliminating the sequences that fall below the predefined minimum support value.

In the fifth phase, frequent sequences that are not maximal are eliminated and a set containing maximal sequences are generated. Algorithmic definition of the Apriori-

All algorithm can be seen in Algorithm 2 taken from [18].

Algorithm 2 AprioriAll Algorithm

Input: D_t: transformed database of transaction sequences

$minsup$: minimum support parameter

Output: $frequentPatterns$: the set of large sequences

```
$F_1$ = frequent 1-sequences;//Result of fitemset phase
for $k = 2$; $F_{k-1} \neq \emptyset$; k++ do
    $C_k$ = apriori-gen($F_{k-1}$);//New candidate sequences
    for all transaction sequence $t \in D_t$ do
        $C_t$ = subseq($C_k, t$);//Candidate sequences contained in t
        for all candidate $c \in C_t$ do $c.count$++
        end for
    end for
    $F_k$ = $\{c \in C_k | c.count \geq minsup\}$;
end for
$frequentPatterns$ = maximal sequences in $\cup_k F_k$
```

CHAPTER 4

DATA AND PROBLEM DEFINITION

In this section, we present details of data used in this work and we give the definitions of three problems related with location prediction problem.

4.1 Call Detail Record Data

In this work we utilized the CDR data of one of the largest mobile phone operators of Turkey. The data corresponds to an area of roughly 25000 square km with a population around 5 million. Almost 70% of this population is concentrated in a large urban area of approximately 1/3 of the whole region. The rest of the region contains some mid-sized and small towns and large rural area with a very little population. The CDR data contains roughly 1 million user's log records for a period of 1 month. For each user there are 30 records per day on average. The whole area contains more than 13000 base stations.

Each record in data represents one of the followings; voice caller, voice callee, sms sender, sms receiver, gprs connection. Besides these cases, no record exists in the CDR data. These records consists of 11 attributes namely, base station id #1, phone number #1, city plate of the phone number #1, base station id #2, other phone number, city plate of the other phone number, call time, cdr type, url, duration, call date. Definition of these attributes and example record attributes are presented in the following subsection.

4.1.1 Attributes

- 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
- city plate of the phone number #1: an integer that represents the city 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
- other phone number: 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
- city plate of the other phone number: an integer that represents the city callee or sms receiver is in. e.g. 06
- call time: The time that action started in a "hhmmss" format. e.g. 170251
- cdr tpye: It can be one of the following;
  - mmo: voice caller
  - mmt: voice callee
  - msmo: SMS sender
  - msmt: SMS receiver
  - gprs: GPRS connection
- url: It is used only for GPRS data. It represents the url that user tries to get.
- duration: It 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

4.2 Problem Definition

In this section, we introduce the 3 narrower problem definitions for broader location prediction problem namely, next location and time prediction using spatio-temporal data, next location change prediction using spatial data and next location change and time prediction using spatio-temporal data. For all three problems there are common unnecessary attributes in data such as, city plate, cdr type, url, duration. These attributes are eliminated since they are not used in further computations. Call time can also be eliminated according to the problem type. We use the term action for any type of phone activity such as voice call, sms, gprs.

4.2.1 Next Location and Time Prediction Using Spatio-Temporal Data

The aim for this problem is to predict the location and the time of the next action in the next time interval of the user roughly. Rather than predicting exact coordinate or base station, it is aimed to find the next region of the user. For this reason, base stations are grouped into the regions and prediction is done accordingly. Moreover, rather than using exact time information, predicting the time interval of the next action is aimed.

Daily user sequences are temporal ordered location and time information pair sequences of the user. For one time interval, there exists one location and time information pair. The location with the most occurrence in the time interval represents the location attribute for that particular time interval. These records are stored in the user sequence database D'.

The problem can be formalized as follows; given a user sequence database D' (obtained from CDR database D) containing daily user sequences, the problem is to find the region and time interval of the next action in the next time interval by using the historical movement sequences.

4.2.2 Next Location Change Prediction Using Spatial Data

The aim for this problem is to predict the next location of the user when he/she changes his/her location. This problem is defined since people usually do not change their location between two actions and this causes misleading high accuracies for the solution of the first problem. Rather than trying to find the location of the next action, we focused on the prediction of the location when the user changes his/her.

For this problem, daily user sequences do not contain the time information. It is used only for temporal ordering while converting CDR database D to user sequence database D'.

The problem can be formalized as follows; given a user sequence database D' (obtained from CDR) database D containing daily user sequences, the problem is to find the next location of the user when he/she changes his/her location by using historical movement sequences.

4.2.3 Next Location Change and Time Prediction Using Spatio-Temporal Data

The aim for this problem is to predict the next location and time of the user when he/she changes his/her location. The difference is that, it is tried to predict the temporal information of the next action when the user changes its location, compared to the second problem definition. Rather than predicting exact time of the action, it is aimed to find the time interval that action takes place in.

Daily user sequences contain both spatial and temporal information of the actions. Different than the one in the first problem definition they contain successive repetitive time intervals.

The problem can be formalized as follows; given a user sequence database D' (obtained from CDR database D) containing daily user sequences, the problem is to find the next location of the user and the time of the action when he/she changes his/her location by using historical movement sequences.

CHAPTER 5

PROPOSED METHODS

In this section we introduce our three solution for the three problems defined in the section 4.2.

5.1 Next Location and Time Prediction Using Spatio-Temporal Data

The method is designed for the problem of predicting the location and time of the next action in the next time interval. Both spatial and temporal attributes of the data is used while building the model. The method consists of 4 steps namely, preprocessing, extracting the regions, extracting frequent patterns and prediction. Details of these steps are given in the following subsections.

5.1.1 Preprocessing

Due to the high volume of the data and high number of attributes, which of them are not relevant for our analysis such as city code, phone number etc., it is necessary to apply some basic preprocessing tasks on the data. First, we filter the unnecessary attributes. Date and time information are merged into a single column and, it is used for sorting records in temporal order. Second information is not used. We further combine call data records of a user on the same day into a single record. By this way, each record, which is structured as a sequence of <base station id, time of the day> pairs, represents a user's daily movement. Time of the day attribute is formatted as 'hhmm'. An example preprocessing step can be seen in 5.1 and 5.2. *B* stands for base station id while *R* stands for region id.

Table 5.1: Before preprocessing

B17083	phone#1	06	B17083	phone#2	06	20120907	010251	mmo	47
B17083	phone#1	06	B28744	phone#3	06	20120907	071008	mmo	3
B10592	phone#1	06	B20062	phone#4	06	20120907	092231	mmo	11
B10592	phone#1	06	B37382	phone#4	06	20120907	111540	mmo	8
B10592	phone#1	06	B10593	phone#5	06	20120907	144332	mmo	14
B10592	phone#1	06	B12912	phone#6	06	20120907	170304	mmo	12

Table 5.2: After preprocessing

B17083,0102	B17083,0710	B10592,0922	B10592,1115	B10592,1443	B10592,1703

5.1.2 Extracting the Regions

In populated parts of the cities, such as downtowns, the base stations are placed very close to each other. Under high number of base stations, it is not practical to consider each station as the center of a movement region to interpret the semantics of the movements. Therefore, in this work, we define regions by grouping the base stations. In spatial clustering, K-Means and K-Medoids are commonly used partitional algorihms[14]. To this aim, we cluster base stations according to their location information (x and y coordinate attributes) using k-means algorithm. The aim of k-means clustering algorithm is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest cluster mean. There are 13281 base station ids in the original data and after exploring several other k values, we group them into 100 clusters which we name as regions. Then, base station ids in the preprocessed data are replaced with the corresponding region ids. At the end of this process, the largest cluster contains 656 base stations and the smallest cluster contains only 6 base stations. Visualization of the regions can be seen in Figure 5.1, Figure 5.2 and Figure 5.3 in three different zoom levels.

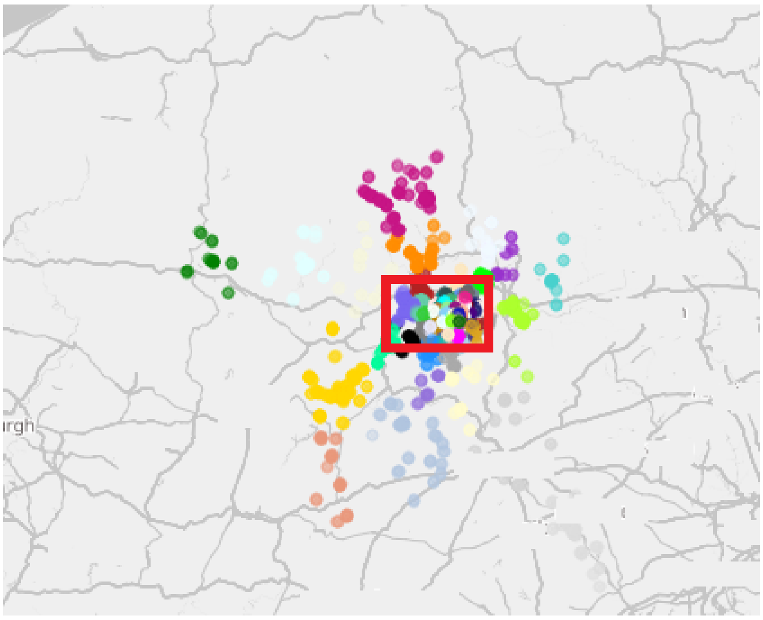

Figure 5.1: Regions in Zoom Level 1

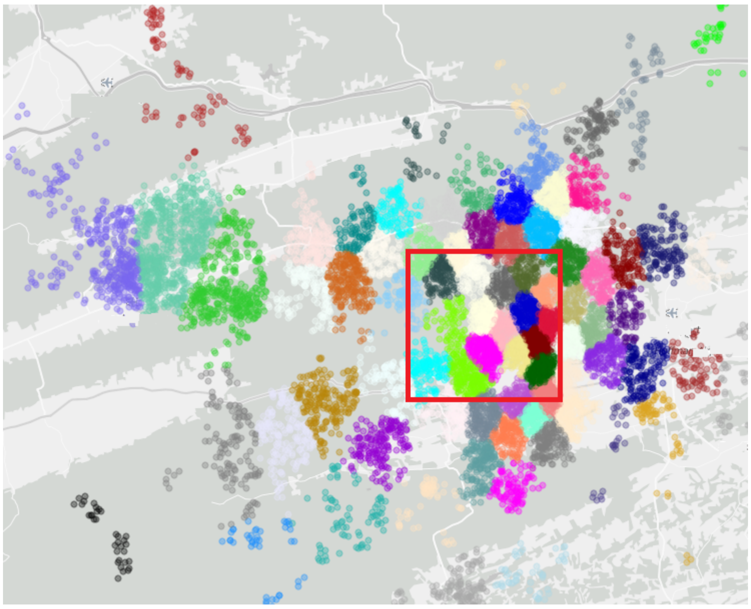

Figure 5.2: Regions in Zoom Level 2

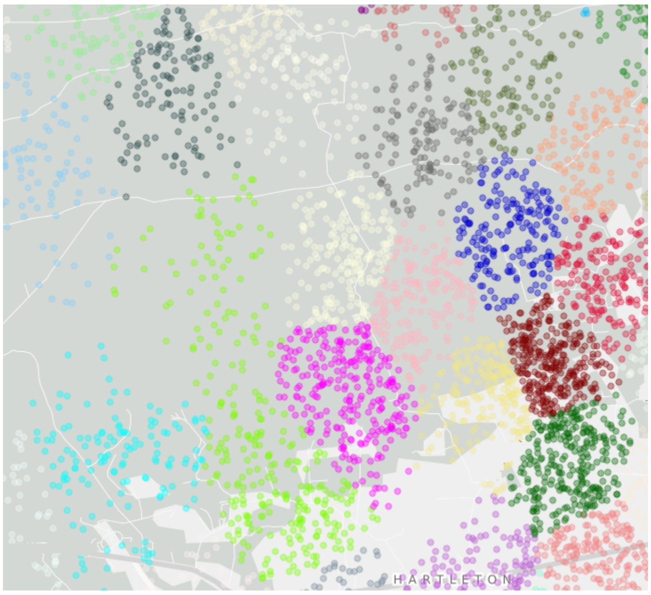

Figure 5.3: Regions in Zoom Level 3

5.1.3 Extracting Frequent Patterns

In this approach we use both spatial and temporal information of each call record. Spatial attribute is the region id and temporal attribute is the time of the day information of the call. We do not use day information because each user's daily sequence corresponds to a single day and we do not take the day of the week or the month into consideration.

Our operation of extracting frequent patterns work with four arguments, namely preprocessed CDR data, pattern length, minimum support and time interval length. Pattern length describes the length of the desired frequent patterns. Minimum support describes the candidate frequent patterns' required proportion in data. Time interval length is used for discretizing time of the day. It indicates the span of discretized time interval. Rather than using exact time information of the action, we prefer to use discretized format for time of the day to be able to augment frequent patterns in data. Applying discretization allows us to eliminate small time differences. Each day is divided into predefined number of equal length time intervals. Action's time of the day information is replaced with the starting time of its corresponding interval. After this change, user's daily sequence may have more than one base station id and time of the day pair having the same time interval value. In order to reduce them to a single pair, for each time interval, the most frequently observed station id is selected as the representative of that interval.

Our extraction method can be defined as a modified version of AprioriAll algorithm. As given in the literature, AprioriAll algoritm consists of two phases, namely; candidate generation and elimination. Difference is in the candidate generation. Normally k-length candidates are generated from (k-1)-length patterns. Since this operation is costly in time for big data and we are only interested in patterns of a given length, we have changed the candidate generation phase. Our algorithm generates all candidates while traversing the data. Two index pointers are used such that one of them points the start of candidate pattern while the other one points the end of it. Count of each candidate pattern observed is recorded for elimination phase. As in the conventional AprioriAll algorithm's candidate elimination phase, candidate patterns whose support value falls below minimum support are eliminated, while the others constitute the frequent pattern set. Table 5.3 demonstrates 3 sample frequent patterns for pattern length 4. In the table, the pairs separated by paranthesis represent region id and discretized time of the day. Region id and discretized time of the day are separated by comma.

Table 5.3: Sample Frequent Patterns

Frequent Pattern (Sequence)	Support
<(R91,1000), (R95,1215), (R45,1615), (R48,1800)>	4.0212e-06
<(R91,1000), (R95,1215), (R45,1615), (R70,1900)>	3.6897e-06
<(R91,1000), (R95,1215), (R45,1615), (R55,1915)>	2.5369e-06

5.1.4 Prediction

In the prediction phase, initially, traversal data of the user for whom prediction will be performed is preprocessed and formatted same as frequent patterns. Assume that the traversal pattern is of length (k-1) and we want to predict the next step, which is the kth element, for this user's traversal. Then, this (k-1) length pattern is used as the test sequence for prediction and we search this test sequence in the frequent pattern set that is created in the extraction phase. If patterns starting with the test sequence have been found, the last element of the matching pattern with the maximum support is generated as the prediction. This process is given in Algorithm 3

Algorithm 3 Prediction Algorithm

```
Input: $testsequence$
Output: $prediction$
$maximumSupport \leftarrow 0$
for all $pattern \in frequentPatterns$ do
    if $testsequence == pattern[1 : k - 1]$ then
        if $maximumSupport < pattern.support$ then
            $prediction \leftarrow pattern[k]$
            $maximumSupport \leftarrow pattern.support$
        end if
    end if
end for
return $prediction$
```

Although we take this approach as our base method, we added two tolerance parameters to the prediction algorithm to improve our results. These are tolerance in time and the multi prediction limit allowed for one instance. Under tolerance in time, patterns are not fixed to some time interval value anymore. They are moved forward or backward in time with a tolerance value. If one pattern is not in frequent pattern set, then tolerance mechanism runs and tries to find tolerated prediction value. Assume that we have a traversal instance

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

and we want to predict next location time pair for that instance but our frequent pattern set does not have a pattern starting with

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

but has

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

As it can be easily seen, *<(R91,1000), (R95,1245), (R45,1630)>* is in the range of 15 minute tolerance of *<(R91,1015), (R95,1230), (R45,1630)>*. If tolerance value for time is greater than 15 minutes, then our method gives the result *(R52,1700)* as the prediction.

The second tolerance parameter, namely the multi prediction limit allowed for one instance, is introduced to utilize the cases in which there are more than one frequent pattern starting with traversal instance. As it can be seen in Algorithm 3, the method returns the last element of the frequent pattern starting with traversal instance with maximum support and does not take other possible matchings into consideration. However by adding multi prediction limit parameter, more than one prediction value are generated. This parameter puts a limit to the proportion of the total support of the patterns in the prediction set, to the total support of all patterns that start with given test sequence. All frequent patterns starting with traversal instance are sorted in decreasing support value order and prediction set is populated by adding kth elements of frequent patterns until the multi prediction limit is satisfied.

For example, *<(R91,1000), (R95,1215), (R45,1615)>* is the traversal instance and there are frequent patterns with length 4 as given in Table 5.3. For this test sequence, in the single prediction method, among the matching patterns, it chooses only *(R48,1800)*, which has the maximum support. If the multi prediction limit is 0.5, it only gives one prediction value which is *(R48,1800).* However if the limit is 0.8, then it gives two predictions which are *(R48, 1800)* and *(R70, 1900).*

5.2 Next Location Change Prediction Using Spatial Data

The method is designed for the problem of predicting the location of the user when he/she changes his/her location. Only spatial attribute of the data is used while extracting frequent patterns without any successively repetitive region ids. The method consists of 4 steps namely, extracting the regions, preprocessing, extracting frequent patterns and prediction. Details of these steps are given following subsections.

5.2.1 Extracting the Regions

Clustering the base stations are done in the same way with the method described in section 5.1.2. We need to cluster base stations before preprocessing because we need to use the region ids in the user's daily sequence which is created in the preprocessing step. The base stations are grouped into 100 regions for this method as well. Then, base station ids in the CDR data are replaced with the corresponding region ids.

5.2.2 Preprocessing

As in the first method, we filter the unnecessary attributes such as city code, phone number etc. Date and time information are also merged into a single column and, it is used for sorting records in temporal order. We again combine call data records of a user on the same day into a single record but this time successive region ids are deleted. By this way, each record, which is structured as a sequence of region ids, represents a user's daily location change pattern. An example preprocessing step can be seen in 5.4 and 5.5

Table 5.4: Before preprocessing

R91	phone#1	06	R91	phone#2	06	20120907	010251	mmo	47
R91	phone#1	06	R21	phone#3	06	20120907	071008	mmo	3
R55	phone#1	06	R27	phone#4	06	20120907	092231	mmo	11
R55	phone#1	06	R27	phone#4	06	20120907	111540	mmo	8
R55	phone#1	06	R91	phone#5	06	20120907	144332	mmo	14
R55	phone#1	06	R3	phone#6	06	20120907	170304	mmo	12

Table 5.5: After preprocessing

R91	R55

5.2.3 Extracting Frequent Patterns

Except for the use of time information, basic intuition behind the extraction method is nearly the same as that of the method described in section 5.1.3. In this approach, the patterns are generated in order to keep only the change of region ids in a single day. To this aim, 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. In addition, time interval elements are also deleted.

5.2.4 Prediction

The basic idea of the prediction that deals with spatio-temporal data is also used in this approach. However, one of our tolerance parameters is different in this prediction method. Since we do not have any time information, time tolerance is not applicable to this approach and it is replaced with the tolerance in pattern length.

Tolerance in pattern length can be applied in two ways. In the first way, tolerance in pattern length gives us opportunity to predict one traversal instance's next region id by examining the shorter frequent patterns. This can be possible when the shorter frequent pattern is a subset of the exact traversal instance (order of region ids are important). Assume that, we have the test sequence;

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

however, there is no exactly matching pattern. Instead, we have the following frequent pattern,

<R77, R95, R16, R22, R41>

Since the set *<R77, R95, R16, R22, R41>* is a subset of *<R77, R91, R95, R16, R22, R41>*, in which the second element of the larger pattern is missing, the last element of frequent pattern that starts with *<R77, R95, R16, R22, R41>* can be given as the prediction result for test sequence *<R77, R91, R95, R16, R22, R41>*.

Second way of tolerating the pattern length gives us opportunity to predict the next region by examining the longer frequent patterns. This is possible when the longer frequent pattern contains the exact traversal instance (order of region ids are important) but also contains some additional region ids. Assume that, we have the following test sequence;

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

however, there is no exactly matching pattern. Instead, we have the frequent pattern that starts with the following,

<R77, R91, R95, R18, R16, R22, R41>

Since the set *<R77, R91, R95, R18, R16, R22, R41>* contains *<R77, R91, R95, R16, R22, R41>* in the same order, which also has the fourth element (R18) as the difference from the traversal instance, last element of frequent pattern that starts with *<R77, R91, R95, R18, R16, R22, R41>* is given as the prediction result for traversal instance <*R77, R91, R95, R16, R22, R41*>.

5.3 Next Location Change and Time Prediction Using Spatio-Temporal Data

The method is designed for the problem of predicting the location and time of the user when it changes its location. Both spatial and temporal attributes of the data is used while extracting the frequent patterns. The method consists of 4 steps namely, extracting the regions, preprocessing, extracting frequent patterns and prediction. Details of these steps are given following subsections.

5.3.1 Extracting the Regions

Clustering the base stations are done in the same way with the two methods described in 5.1.2 and 5.2.2. We need to cluster base stations before preprocessing because we need to use the region ids in the user's daily sequence which is created in the preprocessing step. The base stations are grouped into 100, 200, 400, 800, 1600, 3200 and 6400 regions for this method to analyze the effect of the different region numbers. Then, base station ids in the CDR data are replaced with the corresponding region ids.

5.3.2 Preprocessing

As in the first and second method, after preprocessing, each record, which is structured as a sequence of region ids, represents a user's daily location change pattern. Difference with the second method is the usage of temporal information. For this method user's daily sequence contains not only spatial attribute but also temporal attribute. Repetitive time is allowed for this method while it is not for the first method. An example preprocessing step can be seen in Table 5.6 and Table 5.7

Table 5.6: Before preprocessing

R91	phone#1	06	R91	phone#2	06	20120907	010251	mmo	47
R91	phone#1	06	R21	phone#3	06	20120907	071008	mmo	3
R55	phone#1	06	R27	phone#4	06	20120907	072231	mmo	11
R55	phone#1	06	R27	phone#4	06	20120907	111540	mmo	8
R55	phone#1	06	R91	phone#5	06	20120907	144332	mmo	14
R55	phone#1	06	R3	phone#6	06	20120907	170304	mmo	12

Table 5.7: After preprocessing

R91,0102	R55,0722

5.3.3 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. To this aim, 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.

5.3.4 Prediction

In this method, we use both tolerance parameters, time tolerance and tolerance in pattern length for prediction. Apart from that the prediction algorithm works the same with the first method. For the traversal pattern with the length (k-1), we again predict its next location by searching the frequent patterns starting with that traversal pattern.

CHAPTER 6

EVALUATION AND EXPERIMENTAL RESULTS

In this section, first we introduce our evaluation method and evaluation metrics, and then we give the experimental results for three methods explained in the previous section 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.

6.1 Evaluation

In order to asses the quality of the predictions made by the methods proposed in the previous section we have used k-fold cross validation technique with k=5, on a real CDR data set that has been introduced earlier. Training phase of the evaluation process is nothing but applying the frequent pattern extraction steps of the proposed methods on the training data, in order to generate frequent patterns.

The testing phase has two steps: 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.

For the method proposed for the next location and time prediction using spatio-temporal data problem, we do not prefer to present detailed results. The reason for this preference is the nature of human mobile telephone usage routines. They usually do not change their location between two mobile telephone activities. Because of that prediction accuracy results are misleading. Further analysis will be discussed in the following sections.

For the method proposed for the next location change prediction using spatial data problem, we analyze the effects of minimum support, multi prediction limit, pattern length and length tolerance parameters.

For the method proposed for the next location change and time prediction using spatio-temporal data problem, we analyze the effects of minimum support, multi prediction limit, time interval length, time tolerance, pattern length and the cluster count of base station ids.

6.1.1 Evaluation Metrics

This section describes the method and the metrics that we used in order to measure the success of the proposed prediction methods. We used three different metrics, namely p-accuracy, g-accuracy and prediction count.

Accuracy measures how much of our predictions match with exact next region id of the test pattern. It simply can be defined as the ratio of true predictions to the all predictions. In our case, we have two types of accuracy. The first one, which is the *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. The second one, which is the *p-accuracy* (predictions' accuracy), is the ratio of the number of true predictions to the number of all predictions we are able to make. 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 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. For each of the methods *g-accuracy* and *p-accuracy* metrics are used for representing and interpreting the results.

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.

In addition to true prediction; true positive, false positive and false negative values are calculated for the first two methods that are related with the next location and time prediction using spatio-temporal data and next location change prediction using spatial data problems as given in Algorithm 4

Algorithm 4 Evaluation Algorithm

```
for all prediction in predictionSet do
    if $prediction = actualNextRegionId$ then
        incrementTruePositive(prediction)
        return
    else
        incrementFalsePositive(prediction)
    end if
end for
incrementFalseNegative(actualNextRegionId)
```

For multi prediction, since we increment false positives for each item in our prediction set, and increment false negatives only when none of our predictions do not hold, our results are biased through the recall, rather than precision.

Precision can be defined as the ratio of the number of true positives to the sum of the number of true positives and false positives.

Recall can be defined as ratio of the number of true positives to the sum of the number of true positives and false negatives. Further definitions of these metrics can be found at [13]

6.2 Experimental Results

In this section, the results of the experiments of three proposed methods under different parameters are given.

6.2.1 Results for Next Location and Time Prediction using Spatio-Temporal Data

In this subsection the effect of pattern length and minimum support on g-accuracy are experimentally analyzed.

Pattern Length

In this set of experiments, we analyze the effect of length of the frequent patterns on the g-accuracy of prediction. For this set of experiments, time tolerance is 75 minutes for 15 minute time interval length, minimum support is 10^{-6}, cluster count is 100, and multi prediction support limit is 1.0, which means use all frequent patterns matching with test set patterns.

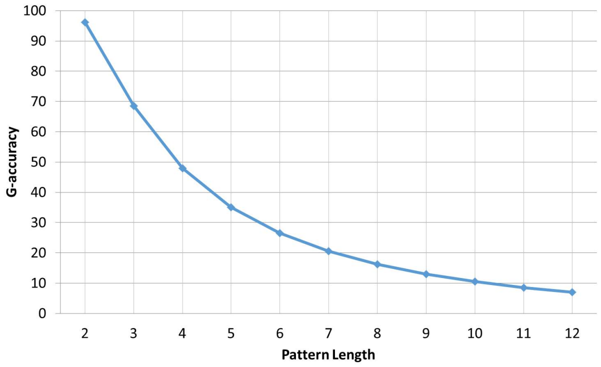

Figure 6.1: Pattern Length vs g-Accuracy

As it can be seen from Figure 6.1, 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 6.1.

Table 6.1: Number of Frequent Patterns for Different Pattern Lengths

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

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 a potential next region we have observed that these numbers are quite high as presented in Table 6.2.

Table 6.2: Number of Average Total Predictions Per Instance for Different Pattern Lengths

Pattern Length	Average Total Prediction Count
2	59.7937065534
3	11.8247757538
4	6.91793091885

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.

Minimum Support

In this set of experiments, we analyze the effect of change in minimum support threshold on the prediction g-accuracy. For this set of experiments, time tolerance is 75 minutes for 15 minute time interval length, pattern length is 6, cluster count is 100, and multi prediction limit is 1.0 which means use all frequent patterns matching with test set patterns.

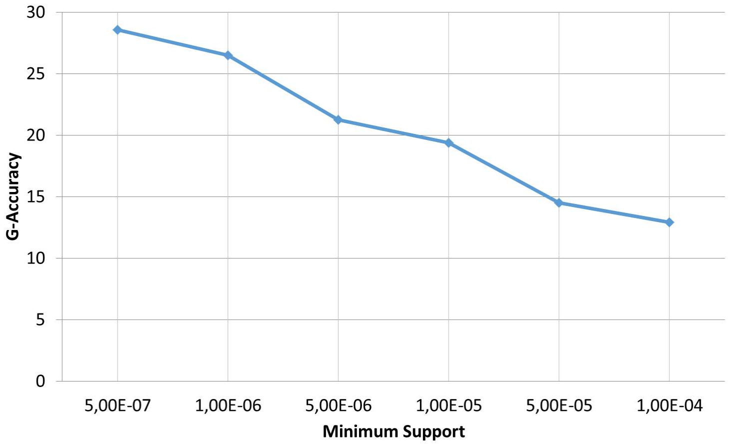

Figure 6.2: Minimum Support vs g-Accuracy

As it can be seen from Figure 6.2, when minimum support threshold value increases, prediction g-accuracy drops. This is due to the fact 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 nth 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 try to evolve this method and do not present further results of this method in this work.

6.2.2 Results for Next Location Change Prediction using Spatial Data

In this subsection, the effect of the pattern length, minimum support, length tolerance and multi prediction limit on the success of the prediction are experimentally analyzed for our second method which aims to predict the change of the location of the users.

Pattern Length

In this part, we analyze the effect of the pattern length on prediction in terms of accuracy, precision and recall. In this set of experiments, multi prediction limit is 0.8, the length tolerance is 2, cluster count is 100, and the minimum support is $4 * 10^{-7}$.

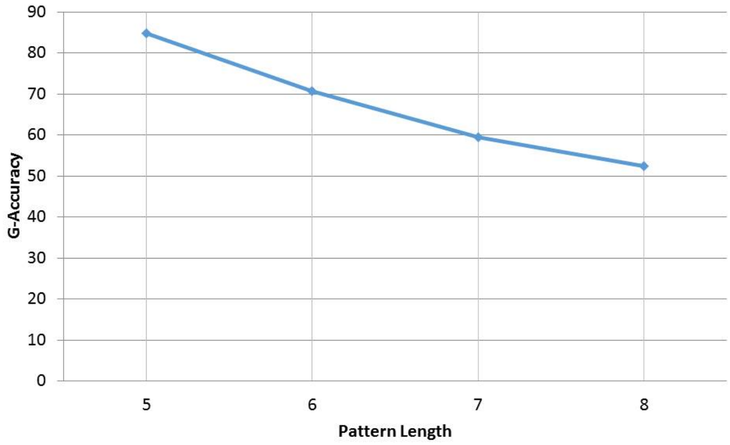

Figure 6.3: Pattern Length vs g-Accuracy

As it can be seen from Figure 6.3, 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 with multi prediction limit 0.8 generated 2.3 predictions on average for successful prediction, which is reasonable value for number of generated predictions.

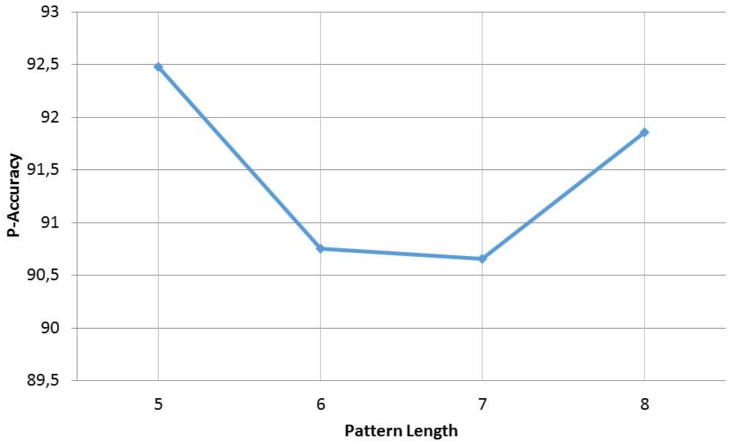

Figure 6.4: Pattern Length vs p-Accuracy

Figure 6.4 shows the relationship between pattern length and p-accuracy. Although it does not present a regular behaviour compared to that of Figure 6.3, it is an expected result. 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 expected not to almost linear decline when pattern length increases. Reason of the lower g-accuracies of higher pattern lengths in the Figure 6.3 is non-predicted instances in test data. However, in the Figure 6.4, we do not include non-predicted patterns in p-accuracy. Prediction count also affects it which can be seen in 6.5. When the quick reduction of prediction count finished at the pattern length 7, p-accuracy starts to increase. It is expected to have greater p-accuracy for the longer patterns with nearly same prediction count.

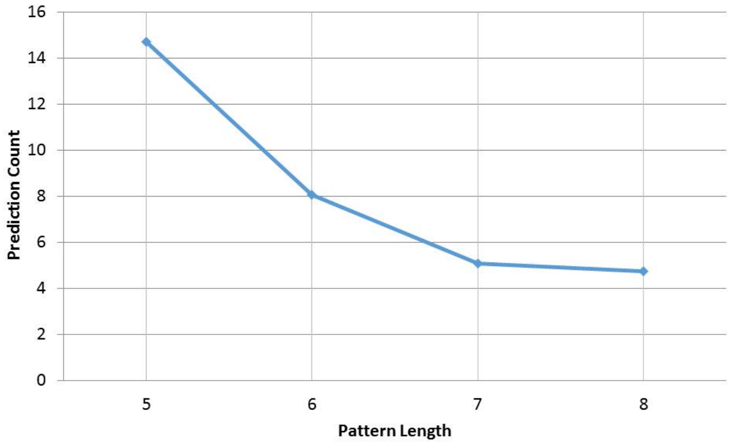

Figure 6.5: Pattern Length vs Prediction Count

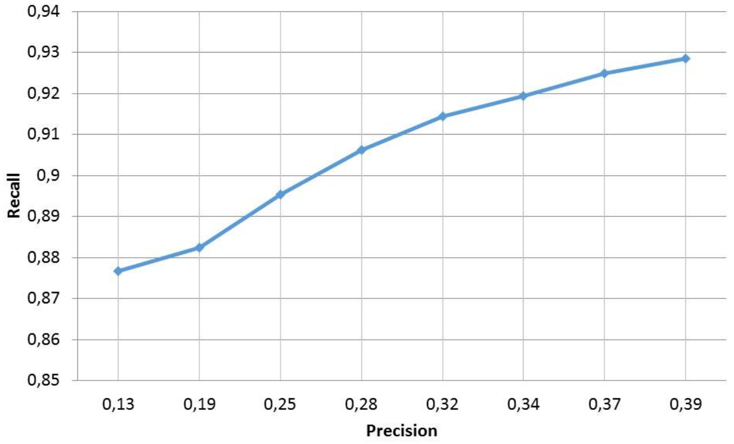

Figure 6.6: Precision vs Recall

As it can be seen from Figure 6.6, both precision and recall values increase as pattern lengths increase from five to twelve. They both increase because the number of true positives grow more than both false positives and false negatives. Reason of getting much larger values of recall than precision is the bias of false positives to false negatives.

Minimum Support

In this set of experiments, we analyze the effect of minimum support on the prediction g-accuracy. In the experiments, pattern length is 5, length tolerance is 2, multi prediction limit is set to 0.8 and cluster count is 100.

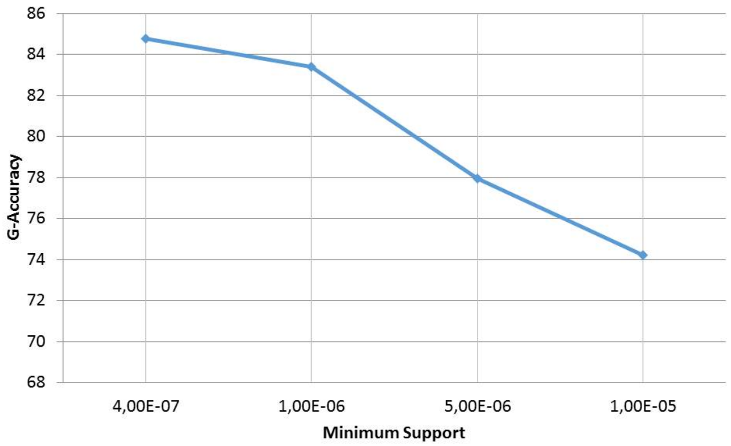

Figure 6.7: Minimum Support vs g-Accuracy

As it can be seen from Figure 6.7, when minimum support value increases, prediction g-accuracy drops as in our first method. Similarly, this is due to that fact that as the minimum support increases, the number of generated frequent patterns decreases.

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

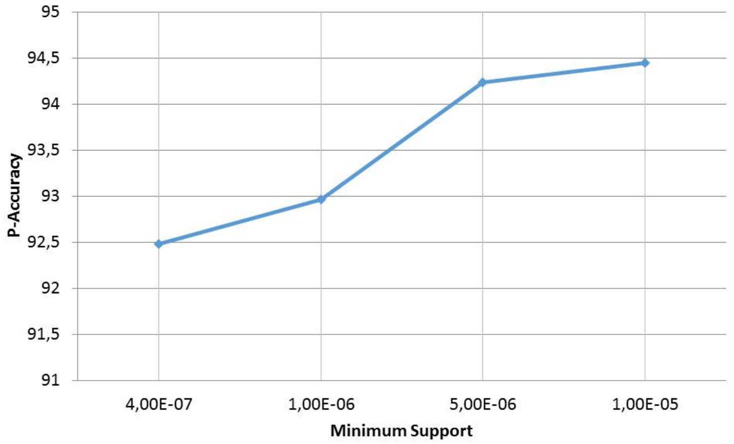

Figure 6.8: Minimum Support vs p-Accuracy

As can be seen in the Figure 6.8, when minimum support value increases, p-accuracy also increases. Since p-accuracy is the prediction accuracy in the predicted test sequences it increases when frequent patterns with higher support values are used. Frequent patterns with lower support values increase makes number of false predictions increase eventually decrease the p-accuracy.

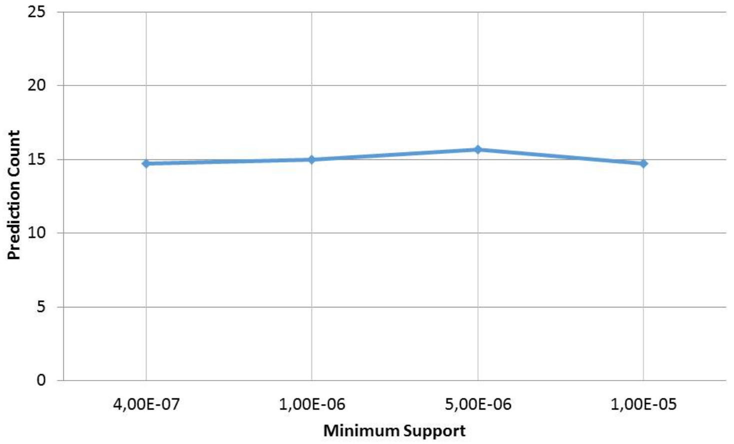

Figure 6.9: Minimum Support vs Prediction Count

As can be seen in the Figure 6.9, minimum support value does not affect the number of predictions made for one correctly predicted test sequence. However it is expected to decrease while minimum support value increases. The reason for stable prediction count is the multi prediction limit value. Only for this experiment multi prediction limit is set to 0.5 to compare the effects of minimum support and multi prediction limit on prediction count. This experiment show that multi prediction limit outweighs the effect of minimum support on prediction count. To see the effect of only minimum support on prediction it can be referred to Figure 6.18 where multi prediction limit is set to 0.8.

Length Tolerance

In this set of experiments, we analyze the effect of length tolerance on the g-accuracy, precision and recall performance of the prediction. In the experiments, multi prediction limit is 0.5 and pattern length is 7. As given in Table 6.3, g-accuracy values are lower than previous algorithm, since minimum support used in this set of experiments is 0.0001 for the sake of execution time.

Table 6.3: Length Tolerance vs g-Accuracy

Length Tolerance	g-Accuracy
0	0.199340297199
1	0.234839334017
2	0.289309194395

As it can be seen in the table, when the length tolerance increases, prediction g-accuracy also increases. G-accuracy values increase since tolerating length feature provides the opportunity to look up different frequent patterns with different lengths for non-predicted test set patterns.

As seen in Table 6.4, precision values decrease when length tolerance increases. It is due to the fact that extra frequent patterns are traversed that have different lengths for non-predicted test set patterns. This increases the number of false positives much more than that of true positives, since the prediction set for one test instance gets larger for higher length tolerance values.

Table 6.4: Length Tolerance vs Precision and Recall

Length Tolerance	Precision	Recall
0	0.873208275761	0.878258665055
1	0.793619440058	0.896896539759
2	0.537772714164	0.913341913657

Multi Prediction Limit

In this set of experiments, we analyze the effect of multi prediction limit on the accuracy. In the experiments, length tolerance is 2, pattern length is 5 and minimum support is 4e-7.

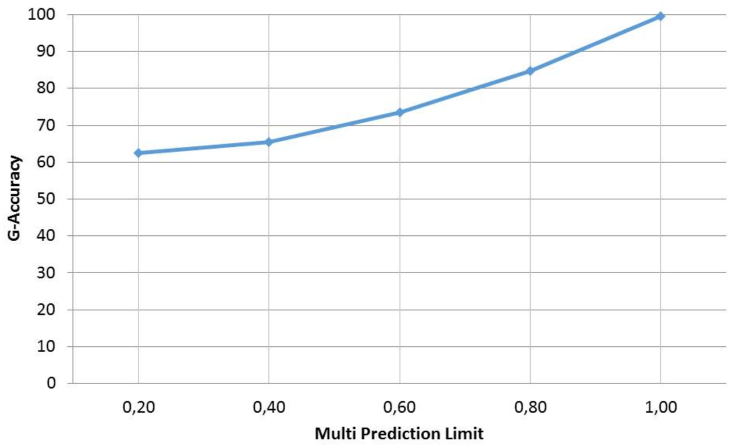

Figure 6.10: Multi Prediction Limit vs g-Accuracy

As it can be seen from Figure 6.10, when multi prediction limit increases, prediction g-accuracy also increases. It is due to the fact that it enlarges the prediction set for each test set pattern, although number of non-predicted test set patterns remains same. Since prediction set increases, the number of false predictions that were made with lower multi prediction limit decreases and the number of true predictions increases, when multi prediction limit increases.

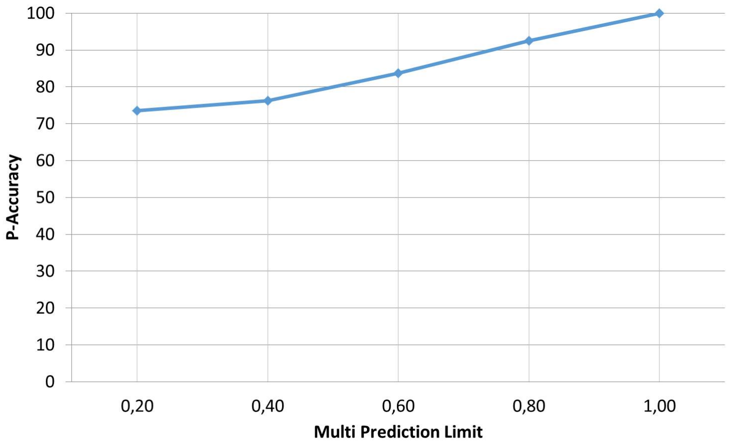

Figure 6.11: Multi Prediction Limit vs p-Accuracy

As it can be seen from Figure 6.11, when multi prediction limit increases, prediction p-accuracy also increases. The important thing in Figure 6.11 is the identicalness of the curve, without taken accuracy values into consideration. This visualize that although prediction accuracy values increase, the number of non-predicted test set patterns remain same.

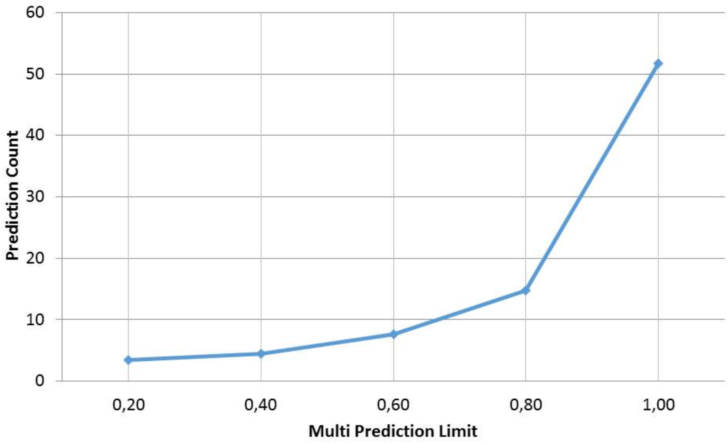

Figure 6.12: Multi Prediction Limit vs Prediction Count

In the Figure 6.12 it can be seen that when multi prediction limit increases, prediction count also increases. It is the expected behaviour by definition since multi prediction limit is introduced to limit the prediction count for the prediction of one test sequence.

6.2.3 Results for Next Location Change and Time Prediction using Spatio-Temporal Data

In this subsection the effect of minimum support, multi prediction limit, length tolerance, pattern length, cluster count, time interval length and time tolerance on g-accuracy, p-accuracy and prediction count are experimentally analyzed for our third method which aims to predict the change of the location and time of the users.

Pattern Length

For this set of experiments, length tolerance is 2, time interval length is 60, time tolerance is 120, multi prediction limit is 0.8, cluster count is 100, and minimum support is 4e-7.

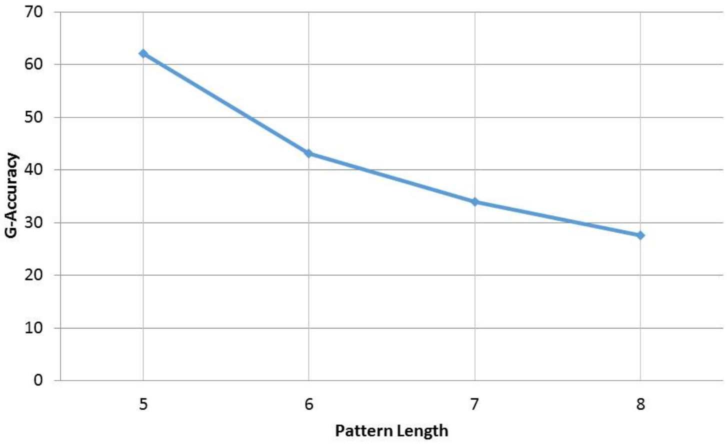

Figure 6.13: Pattern Length vs g-Accuracy

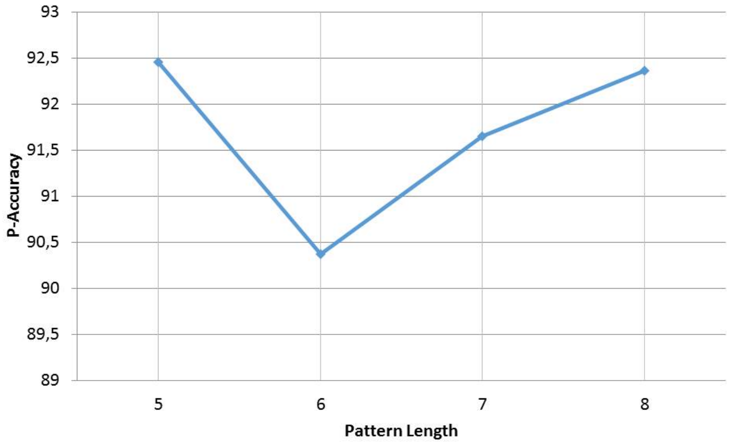

Figure 6.14: Pattern Length vs p-Accuracy

As it can be seen from Figure 6.13, when pattern length increases, g-accuracy decreases. Since it is much harder to find frequent patterns for longer patterns, g-accuracy eventually decreases. However as can ben seen in Figure 6.14 p-accuracy do not have a continuous increase or decrease when pattern length increases. The reason for this different behavior is presented in the previous section.

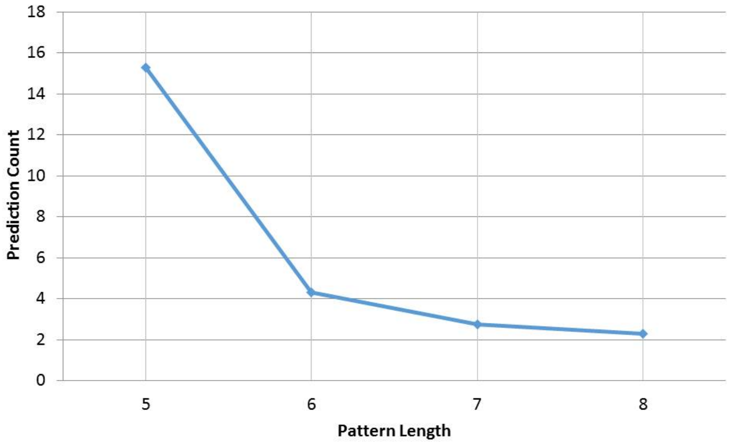

Figure 6.15: Pattern Length vs Prediction Count

Minimum Support

For this set of experiments, pattern length is 5, length tolerance is 2, time interval length is 60, time tolerance is 120, cluster count is 100, and multi prediction limit is 0.8.

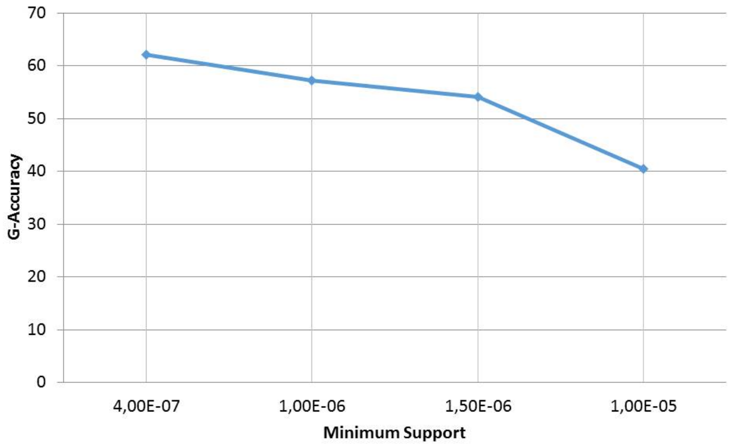

Figure 6.16: Minimum Support vs g-Accuracy

As it can be seen from Figure 6.16, when minimum support value increases, g-accuracy decreases. It is an expected behaviour to have smaller g-accuracy values for the greater minimum support values since the greater minimum support value means the less frequent pattern which eventually causes the predicted sequence's number superficially drop. Consequently the g-accuracy decreases.

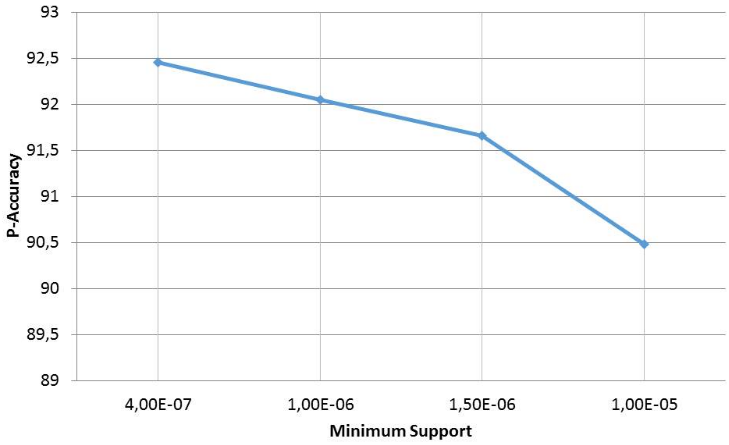

Figure 6.17: Minimum Support vs p-Accuracy

As it can be seen from Figure 6.17, when minimum support value increases, p-accuracy decreases. When compared to the effect on g-accuracy, reduction in p-accuracy is much less than it. The reason is related with the definition of p-accuracy. P-accuracy does not take unpredicted sequences into consideration. However still there is a small decline in the graph, since the extracted frequent patterns are much lower for the greater minimum support values.

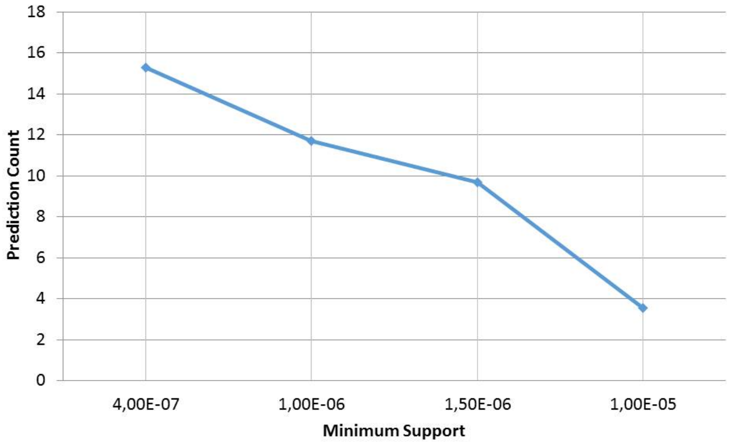

Figure 6.18: Minimum Support vs Prediction Count

As it can be seen from Figure 6.18, when minimum support value increases, prediction count decreases. The reason is the same with the previous two graphs, lower number of extracted frequent patterns.

Length Tolerance

For this set of experiments, pattern length is 5, time interval length is 60, time tolerance is 120, multi prediction limit is 0.8, cluster count is 100, and minimum support is 4e-7.

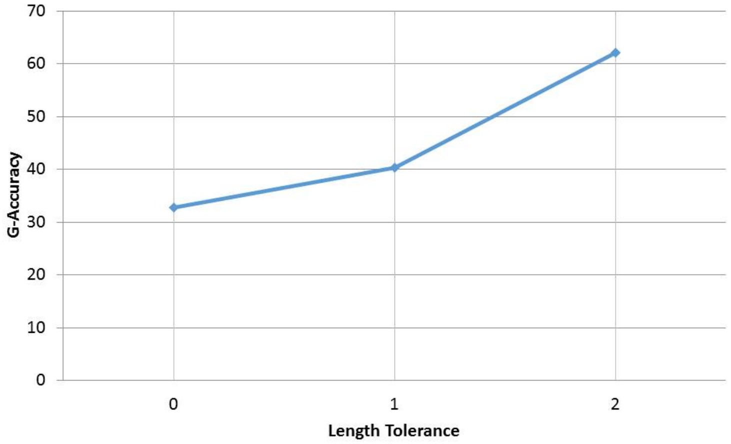

Figure 6.19: Length Tolerance vs g-Accuracy

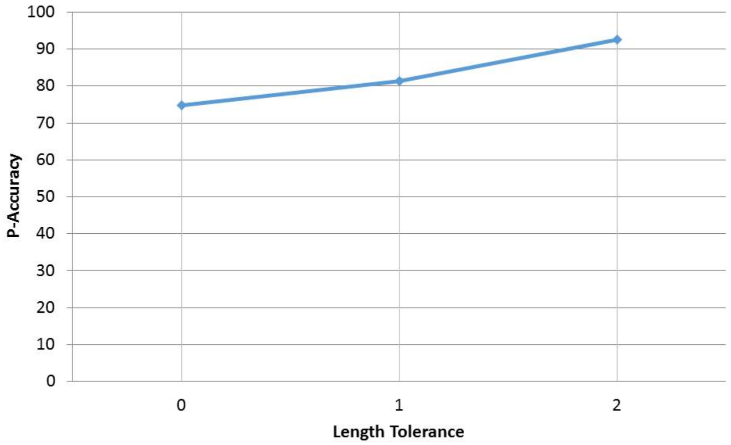

Figure 6.20: Length Tolerance vs p-Accuracy

As it can be seen from Figure 6.19 and 6.20, when length tolerance increases g-accuracy and p-accuracy increases. Increasing length tolerance makes some unpredicted test sequences predictable which increases the g-accuracy. True predicted with greater length tolerance sequences also increases the p-accuracy although its increase is much lower than the g-accuracy. Moreover, as it can be seen from 6.21 more length tolerance makes prediction sets larger.

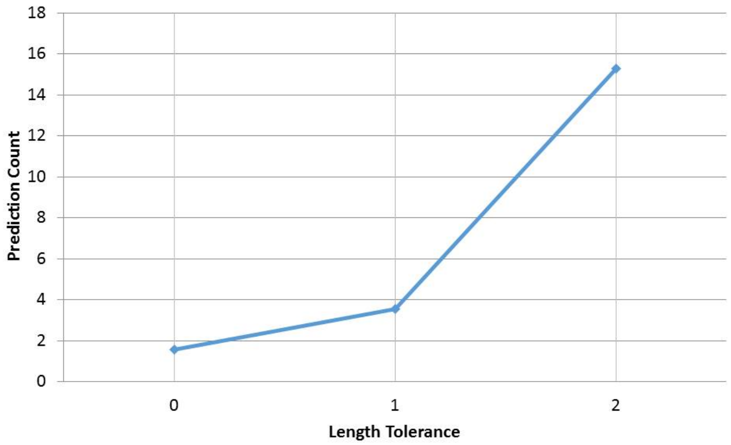

Figure 6.21: Length Tolerance vs Prediction Count

Multi Prediction Limit

For this set of experiments, pattern length is 5, length tolerance is 2, time interval length is 60, time tolerance is 120, cluster count is 100, and minimum support is 4e-7.

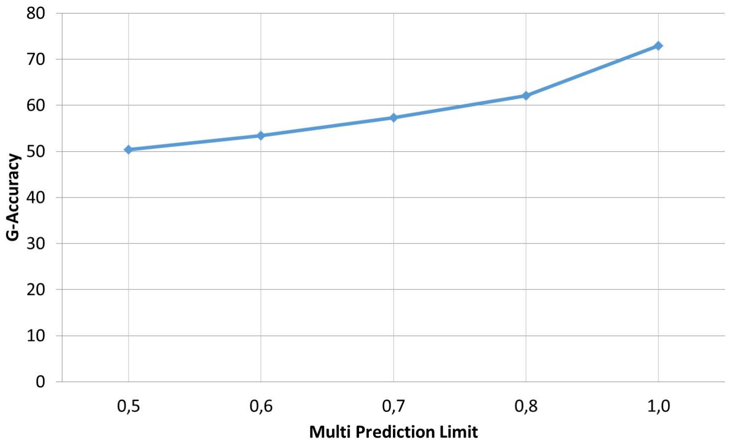

Figure 6.22: Multi Prediction Limit vs g-Accuracy

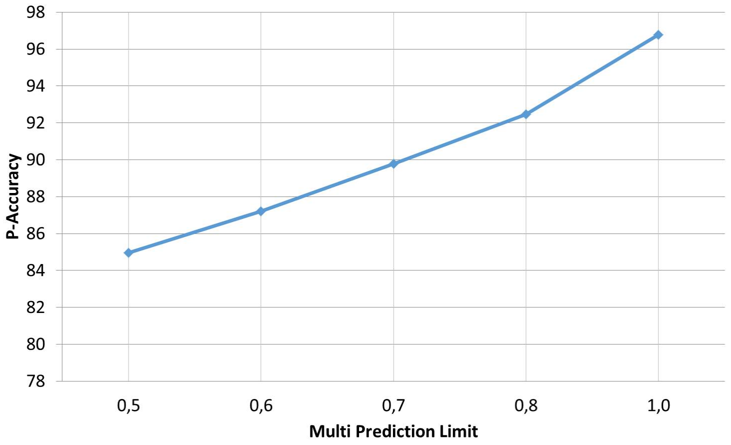

Figure 6.23: Multi Prediction Limit vs p-Accuracy

As it can be seen from the 6.22 and 6.23, when multi prediction limit increases g-accuracy and p-accuracy increase. The greater multi prediction limit means the larger prediction set for one test sequence. Therefore it increases the value of the both of the accuracy metrics. In addition to it, prediction count increases by definition which can be seen in 6.24.

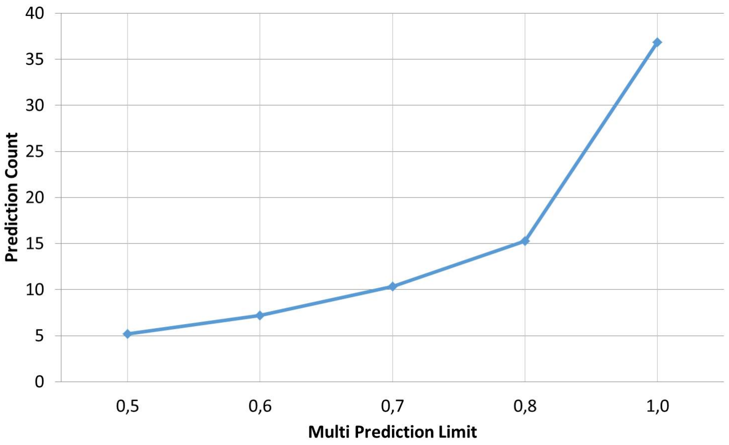

Figure 6.24: Multi Prediction Limit vs Prediction Count

Cluster Count

For this set of experiments, pattern length is 5, length tolerance is 2, time interval length is 60, time tolerance is 120, multi prediction limit is 0.8 and minimum support is 4e-7.

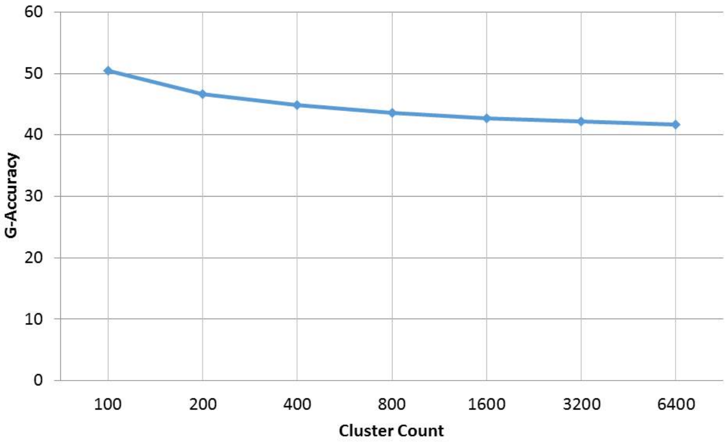

Figure 6.25: Cluster Count vs g-Accuracy

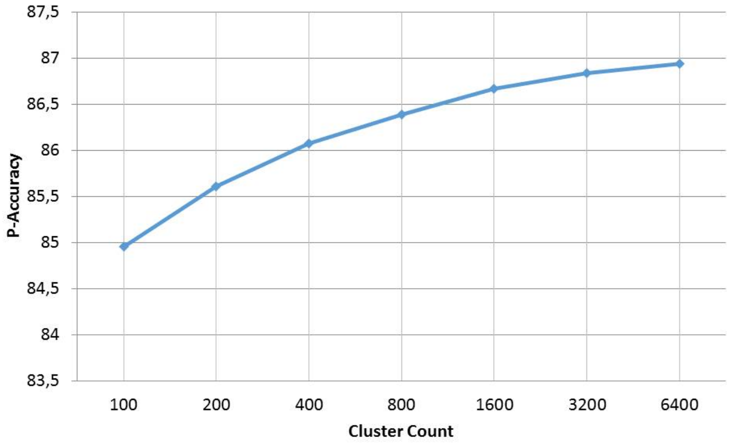

Figure 6.26: Cluster Count vs p-Accuracy

As it can be seen from Figure 6.25, when cluster count 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 as it can be seen Figure 6.26, p-accuracy increases when cluster count increases. It is because of 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, they usually give correct predictions when compared to the less cluster counts. It also eventually decrease the size of prediction set in other words prediction count which can be seen in Figure 6.27. It should also be noted that for this analysis we used multi prediction limit as 0.5 rather than 0.8.

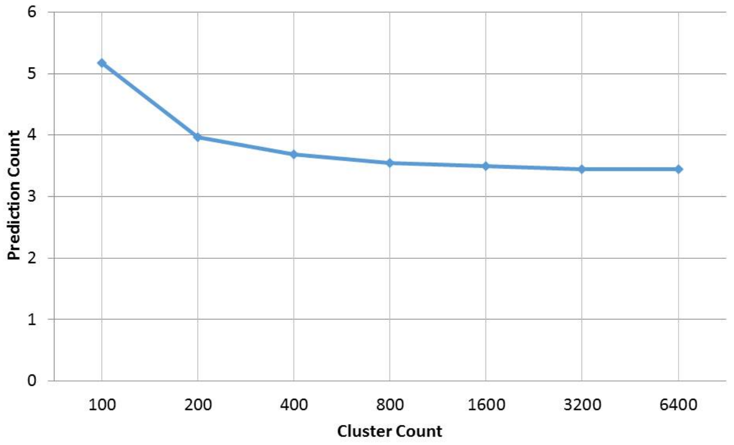

Figure 6.27: Cluster Count vs Prediction Count

Time Interval Length

For this set of experiments, pattern length is 5, length tolerance is 2, time tolerance is 0, multi prediction limit is 0.8, cluster count is 100, and minimum support is 4e-7.

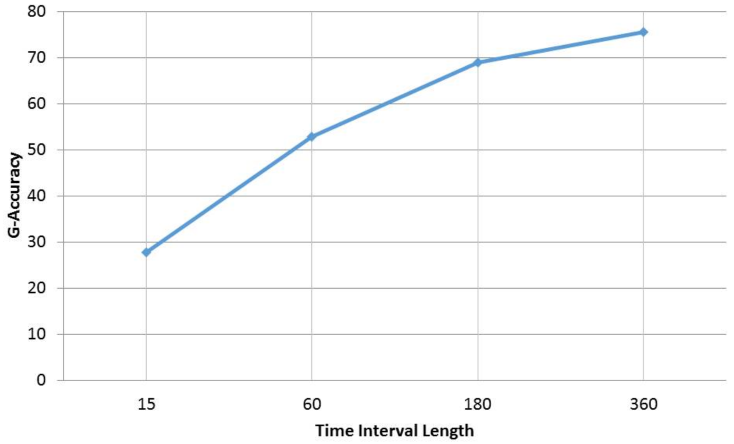

Figure 6.28: Time Interval Length vs g-Accuracy

As it can be seen from the Figure 6.28 and 6.29, when time interval length increases, g-accuracy and p-accuracy increase. Since the larger time interval means the more similar daily sequences and eventually higher number of frequent patterns, increase in the values accuracy metrics is an expected behavior. We can say that prediction count increases in general while the time interval length increases although for time interval length 360 it decreases, but it is a negligible. As can ben seen in 6.30 the reason for increase in the size of the prediction set is the same reason for the g-accuracy and p-accuracy increase; higher number of frequent patterns.

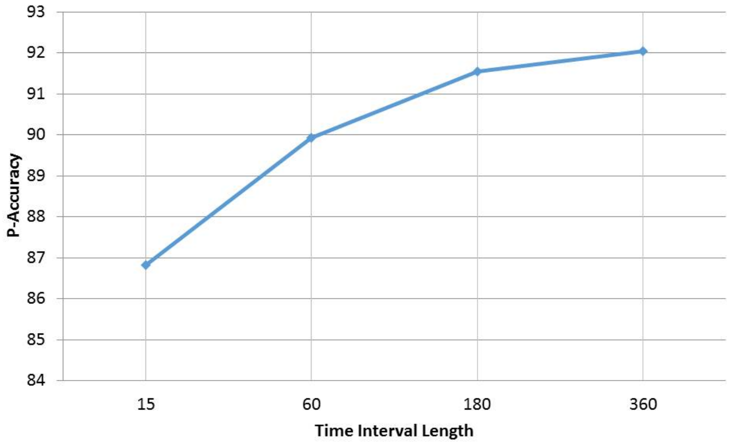

Figure 6.29: Time Interval Length vs p-Accuracy

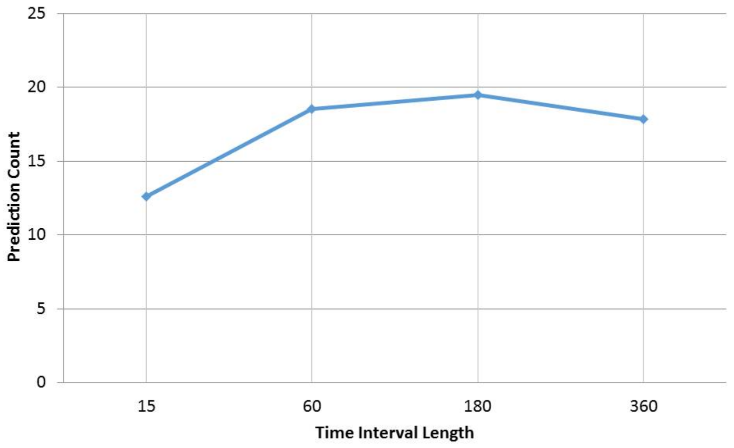

Figure 6.30: Time Interval Length vs Prediction Count

Time Tolerance

For this set of experiments, pattern length is 5, length tolerance is 2, time interval length is 60, multi prediction limit is 0.8, cluster count is 100, and minimum support is 4e-7.

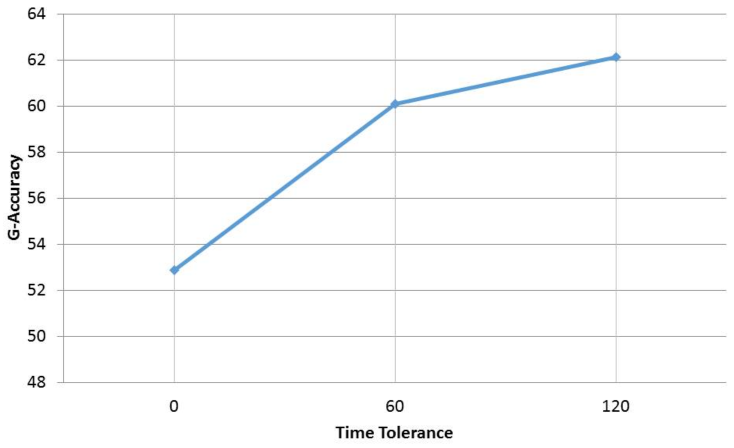

Figure 6.31: Time Tolerance vs g-Accuracy

As it can be seen from Figure 6.31 and 6.32, when time tolerance increases g-accuracy and p-accuracy increases. It is expected since the greater time tolerance gives prediction model ability to search for different time intervals when it can not create prediction for a fixed time interval sequences or can not predict true region id and time interval.

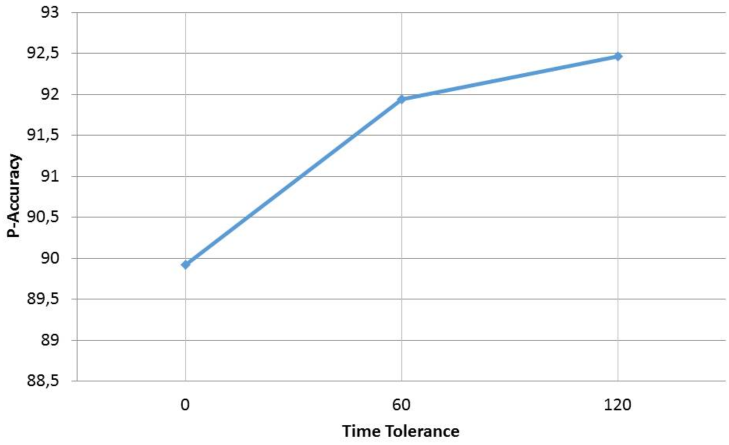

Figure 6.32: Time Tolerance vs p-Accuracy

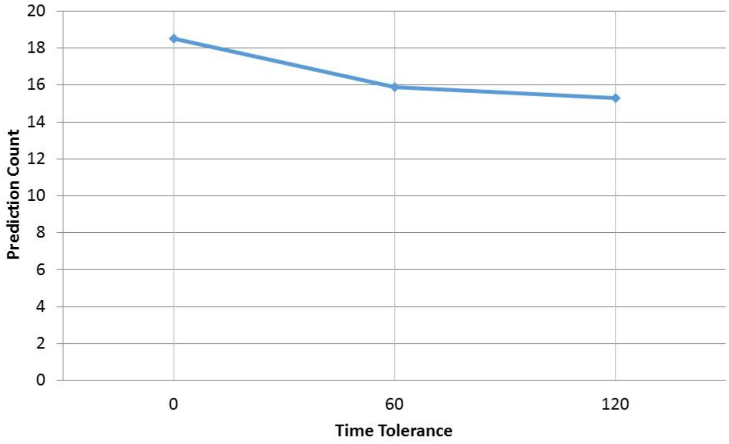

Figure 6.33: Time Tolerance vs Prediction Count

As it can be seen from Figure 6.33, when time tolerance increases prediction count decreases. It is because prediction count represents the size of the prediction set when it only gives correct prediction. Since the correct predictions increase while unpredicted sequences decrease, prediction count decreases.

CHAPTER 7

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 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 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 we focoused 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.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *ICDE*, pages 3–14. IEEE Computer Society, 1995.

[3] Chiara Boldrini and Andrea Passarella. Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Computer Communications*, 33(9):1056–1074, June 2010.

[4] Huiping Cao, Nikos Mamoulis, and David W. Cheung. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Trans. on Knowl. and Data Eng.*, 19(4):453–467, April 2007.

[5] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[6] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3(1376), March 2013.

[7] Huiji Gao, Jiliang Tang, and Huan Liu. Mobile location prediction in spatio-temporal context. In *the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing*. Nokia, June 2012.

[8] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 330–339, New York, NY, USA, 2007. ACM.

[9] Győző Gidófalvi and Fang Dong. When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Work-*

shop on Mobile Geographic Information Systems, MobiGIS '12, pages 57–64, New York, NY, USA, 2012. ACM.

[10] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[11] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[12] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[13] Mert Ozer, Ilkcan Keles, İsmail Hakki Toroslu, and Pinar Karagoz. Predicting the change of location of mobile phone users. In *Proceedings of the Second ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, MobiGIS '13, pages 43–50, New York, NY, USA, 2013. ACM.

[14] Shashi Shekhar, Michael R. Evans, James M. Kang, and Pradeep Mohan. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 1(3):193–214, 2011.

[15] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[16] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[17] Nguyen Thanh and Tu Minh Phuong. A gaussian mixture model for mobile location prediction. *The 9th International Conference on Advanced Communication Technology*, 2(9):914 – 919, February 2007.

[18] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, December 2007.

[19] Gökhan Yavas, Dimitrios Katsaros, Özgür Ulusoy, and Yannis Manolopoulos. A data mining approach for location prediction in mobile environments. *Data Knowl. Eng.*, 54(2):121–146, August 2005.

[20] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 34–43, New York, NY, USA, 2011. ACM.

[21] Daqiang Zhang, Athanasios V. Vasilakos, and Haoyi Xiong. Predicting location using mobile phone calls. *SIGCOMM Comput. Commun. Rev.*, 42(4):295–296, August 2012.

[22] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA, 2009. ACM.