Master's thesis

Two years

The Department of Information Technology and Media (ITM)
Computer Engineering MA, Final project, 30 credit points

Mobile phone context prediction using Markov chains

Author: Emil Skariah
E-mail: emsk1003@student.miun.se
Examiner: Professor Tingting Zhang, Tingting.Zhang@miun.se
Tutor: Victor Kardeby, Victor.Kardeby@miun.se
Scope: 8965 words inclusive of appendices
Date: 2013-05-20
Abstract

The objective of this study has been to answer the question in relation to predicting the future local context of a mobile device. Context awareness is the idea behind this study which and the aim is to assist in the building of context-aware applications. There has been an increased demand for these applications based on the introduction of smart phones which contain built-in sensors and actuators. Discrete context data such as Cell ID, can assist in building a context-aware application which may be able to predict the future state of a mobile device. The use of a Markov chain is one possible means of assisting with the prediction of the next context state. This thesis presents a model of a context-aware application which can predict the future local context of a mobile device with the assistance of a Markov chain. This can be verified by comparing the model with the randomly generated values.

Keywords: Context awareness, Cell ID, WiFi, Markov chain, Prediction.
Acknowledgments

I highly appreciate the enormous help given to me by my supervisor Victor Kardeby. His encouragement and help has enabled me to successfully complete my thesis and write a report on it.

I also would like to give my sincere thanks and gratitude to Prof. Tingting Zhang for her support and assistance during my thesis work.
# Table of Contents

Abstract ........................................................................................................................................... ii
Acknowledgments .......................................................................................................................... iii
Terminology ...................................................................................................................................... vi

1 Introduction ................................................................................................................................... 1
  1.1 Background and problem motivation ...................................................................................... 1
  1.2 Overall aim .............................................................................................................................. 2
  1.3 Scope ...................................................................................................................................... 2
  1.4 Concrete and verifiable goals ................................................................................................. 2
  1.5 Outline ................................................................................................................................... 3

2 Theory .......................................................................................................................................... 4
  2.1 Context Information ................................................................................................................. 4
  2.2 Cell Ids and WiFi .................................................................................................................... 4
    2.2.1 A basic GSM network ........................................................................................................ 4
    2.2.2 WiFi .................................................................................................................................. 6
    2.2.3 Wireless Access Point ....................................................................................................... 6
  2.3 Markov chain .......................................................................................................................... 7
    Transition probability matrix ........................................................................................................ 8
  2.4 Android ................................................................................................................................... 8
    Android architecture .................................................................................................................... 8

3 Methodology .................................................................................................................................. 9
  3.1 Discrete context data ............................................................................................................... 9
    3.1.1 Identification of Cell ID .................................................................................................... 9
    3.1.2 Getting WiFi from available networks .............................................................................. 10
  3.2 Approach ................................................................................................................................ 10
  3.3 Alternative models .................................................................................................................. 10
    3.3.1 Markov chain Model ......................................................................................................... 11
    3.3.2 Discrete SEQ Model ........................................................................................................ 11
    3.3.3 Market Basket Analysis .................................................................................................... 12
    3.3.4 Prediction by Partial Matching ......................................................................................... 12
    3.3.5 Hidden Markov Model ..................................................................................................... 12
    3.3.6 Summary .......................................................................................................................... 12
  3.4 Verification .............................................................................................................................. 13
    3.4.1 Approach to the goal .......................................................................................................... 13
    3.4.2 Accuracy of the model ....................................................................................................... 13
    3.4.3 Response time .................................................................................................................... 14

4 Implementation ............................................................................................................................. 15
  4.1 Implementation overview ......................................................................................................... 15
  4.2 Proposed Implementation ......................................................................................................... 15
    4.2.1 External-Real World ......................................................................................................... 15
    4.2.2 Internal-Prediction World ................................................................................................. 17
    4.2.3 Application Phase .............................................................................................................. 17
Mobile phone context prediction using Markov chains
Emil Skariah
2013-05-20

4.3 Entity-Relationship Model................................................................. 17
4.3.1 Predict Context............................................................................... 18
4.3.2 GSM Cell Location........................................................................... 18
4.3.3 Prediction Engine............................................................................... 18
4.3.4 State Prediction.................................................................................. 18
4.3.5 Probability of Cell Ids................................................................. 18
4.3.6 Store Values.................................................................................... 18
4.4 Application Model.............................................................................. 19
4.5 Android Prototype............................................................................. 21

5 Results..................................................................................................... 22
5.1 Creating Transition Matrix Tables....................................................... 25
5.2 Transition Matrix Comparison............................................................. 28

6 Conclusions .......................................................................................... 33
6.1 Problem statements............................................................................. 33
6.2 Results.................................................................................................. 33
6.3 Future work.......................................................................................... 34

References.................................................................................................... 35

Appendix A: Documentation of own developed program code................. 37

Code for plotting the graph of results......................................................... 37
Terminology
This section will explain the different types of abbreviation that are used in this thesis report.

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BTS</td>
<td>Base Transceiver Station</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communication</td>
</tr>
<tr>
<td>HSDPA</td>
<td>High Speed Down-link Packet Access</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Developing Environment</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunication System</td>
</tr>
<tr>
<td>WiFi</td>
<td>Wireless Fidelity</td>
</tr>
</tbody>
</table>
1 Introduction

As technologies are changing, mobile phones are now capable of connecting to the Internet and their introduction is now worldwide. This has made the use of the Internet somewhat easier because of its main advantage of ‘connectivity on the go’. However sometimes the Internet becomes disconnected while traveling because of the low network coverage for that particular area. This temporary unavailability of service can create data starvation for those users who continuously use the Internet. This provides a mandate for producing a prototype that can enable continuous usage of evolving context information regardless of network connectivity.

As a solution to this data starvation, prediction can be used for perfecting the data in advance before the network disconnection occurs. This can be achieved by using the concept of the Markov chain and the finite state machine. In this project the intention is to predict the next context of a mobile phone with the assistance of both Markov chain and a finite state machine. The Markov chain model will predict the next context based on the previously stored context data. A finite state machine stores the status of the current state at a given time and has the ability to change the state to another state for any given change in the context data.

1.1 Background and problem motivation

Even though the availability of the Internet has been evolved vastly as compared to the previous years, mobile devices and their applications are still facing an Internet connectivity problem. This interrupted connectivity leads to a discontinuity in the usage of Internet applications in mobile devices. As a solution to this problem, if the mobile device can predict the interrupted Internet connectivity, then it is possible for it to pre-fetch the data before a network disconnection occurs. This will assist in creating an illusion for the user that there is still connectivity even though there is an Internet connectivity problem.

Prediction can be performed by making use of discrete context information of a mobile device. In order to formulate a continuous usage of emerging context information, context prediction can be used in the mobile applications, which are able to act on predicted context by pre-fetching the data before a network disconnection occurs. Any information that individuates the relationship between users, application and environment is defined as context information. This context information is used in building context-aware applications. Context-aware applications are only applications that can change their behavior based on the context information.

In this thesis, the problems associated with the internet connectivity have been addressed by implementing the concept of prediction in mobile devices.
1.2 Overall aim

The overall aim of this project is to predict the next future state of a mobile device in order to avoid data starvation. In order to predict the next future state of a mobile device, a data set must be created by recording the current state of the device and its state transitions. This data set can be created with the help of sensors that are built into the new generation of smart phones. This includes collecting context from the sensors and then predicting the future state of the mobile device.

One of the methods for solving the problem of prediction is to make use of the Markov chain concept. A Markov chain is a memory less system in which the future state does not depend on the past states; only the current state. By implementing the Markov chain concept in the prediction engine of the model, it becomes possible to predict the next state of a mobile device.

As a next step towards the context prediction, this context information can be shared by sending the model between the users. By using this model, the user A is able to determine the current location of user B instead of searching for the user location.

1.3 Scope

The project will focus on the context data which can be used to predict the future state of a mobile device. The context data will predict in the finite state machine and hence it is necessary to make use of discrete context data. A continuous set of data variables such as location coordinates which consists of both latitude and longitude does not have a finite state. It is not possible to make a model with an infinite number of positions. Hence latitude and longitude provides continuous data, however, it is not possible to change those data into finite states without including more work.

Cell Ids or available WiFi are the main context data that will be used within this project. These are discrete context data and are able to be predicted in the finite state machine by using Markov chain models. Since discrete context data are finite, it is possible to learn about such data and thus conduct prediction using sensors. Security is an important aspect in relation to securing all the data. However this aspect falls outside the scope of this project. It is necessary to solve the actual problem of making prediction before it becomes necessary to deal with security.

1.4 Concrete and verifiable goals

The basic goals of this project are (1) record the current state of a mobile device and its state transitions using sensors and (2) predict the future local context of a mobile device with an appropriate prediction model. All the goals are summarized in the following list;
• Create a data set from the input states such as Cell ID/WiFi by recording the current state of the device and its state transitions by using sensors that are built into the new generation of smart phones.
• Use model context as a finite state machine and use a prediction model in order to discover the next future state of a mobile device.
• Check the accuracy of the system by comparing the predicted contexts with previously predicted contexts.

1.5 Outline
Chapter 2 describes the theory, focusing on the discrete context data and context prediction in addition to the tools used in the prediction. Chapter 3 describes the methodology used for this thesis and chapter 4 focuses on the implementation part. While chapter 5 summarises the results and chapter 6 consists of conclusions.
2 Theory

The theory chapter represents the theoretical background of technologies that are used in this thesis. The theory starts with discrete context information. Following this are the components that are being used in relation to context data for the prediction of future states. Finally, the Markov chain process is dealt with and it concludes by explaining the Android platform.

2.1 Context Information

Context can be defined as any information or data in an environment. The representation of an object changes dynamically with data received from different sensors in the environment and also by interaction with other objects in the same environment[1]. In the author's opinion the definition by A. Dey and G. Abowd is the most appropriate. They state that context defines a subset of a physical or conceptual status, which is of interest to a certain entity. Their definition includes the complete range of context in such a universal way that it can be applied to very different applications. Dey and Abowd propose the following definition [2]:

“Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.”

In this thesis, Cell ID and WiFi are the two different context data that are used. Both contexts involve discrete data, which is necessary in the desired prediction model. Section 2.2 elaborates the selected context data in detail.

2.2 Cell IDs and WiFi

The Cell ID is a unique number that is used to identify the Base Transceiver Station (BTS), mainly the cell tower in which the mobile device is connected to the cellular networks such as GSM, GPRS, UMTS/HSDPA. A Cell ID is the basic method that to identify the location of a mobile device. [3]

2.2.1 A basic GSM network

The basic components and the functionality of a basic GSM cellular network[4] are explained in the following section;

- Mobile station (MS)

The mobile station consists of Mobile equipment (ME) and Subscriber Identity Module (SIM). The mobile equipment involves the mobile devices such as mo-
Mobile phone context prediction using Markov chains
Emil Skariah
2013-05-20

Mobile phones. The SIM consists of the entire entire user related information such as user identification, authentication secret key etc.

• **Base station subsystem (BSS)**

The base station subsystem consists of the Base transceiver station (BTS) and Base station controller (BSC). The BTS is the Mobile station's access point to the network, which is responsible for the radio communications between the Mobile station and the network. The Base station controller controls multiple BTSs by setting up the radio communications from the mobile device. It is also involved in the handover process from one BTS to another under the same BSC.

• **Network switching subsystem (NSS)**

The Network switching subsystem has a central component called the Mobile switching center (MSC), which is the heart of a GSM network. The functions of an MSC are in managing the location of mobiles, call routing, call switching, call setup and, additionally, controls the handover from the Base station controller. The MSC has a database which contains the following details;

– Home Location Register (HLR) maintains the information about the registered subscribers in the area of MSC.

– Visitor Location Register (VLR) maintains the information about the users other than the registered local subscribers in the MSC.

– Equipment Identify Register (EIR) maintains a database which contains the mobile terminals.

– Authentication Center (AUC) verifies the user identities.

The basic architecture of a GSM network is illustrated in figure 2.1 below.

• **Gateway mobile switching center (GMSC)**

The GMSC functions as an interface between the mobile switching center and an outside network such as a Public Switched Telephone Network (PSTN). This is used for placing a regular call from the regular mobile network to the outside landlines.
2.2.2 WiFi

Wireless Fidelity (WiFi) is a wireless networking technology used for a non physical wired communication between mobile devices in relation to transmitting data using a wireless signal. It uses the technology called Radio Frequency (RF) which operates within the electromagnetic spectrum of radio wave propagation. An electromagnetic field is created when a radio frequency current is supplied to an antenna, which will then propagate through space. [6]

2.2.3 Wireless Access Point

Every wireless network has an Access Point (AP) which is a device used to broadcast a wireless signal, hence the mobile devices can connect. A Router is a device that is commonly used as an Access point for connecting wireless devices to a wired network. Every router has a unique ID called its Media Access Control address (MAC) in order to identify each wireless Access point.

The structure of a wireless network and the access point are illustrated in the figure 2.2.
2.3 Markov chain

A Markov chain is a sequence of discrete stochastic processes, events or actions which are purely used for predicting various future states. It is a memory-less system where the future state depends on the current state and is independent of past states. A stochastic event is a series of random elements without an order or pattern [8]. A simple Markov chain, having 3 states, is illustrated in figure 2.3 below.
Transition probability matrix

In a Markov chain, the transition probability matrix is used to define the transitions of a Markov chain. Transition defines the state changes from one state to another at a given time.

2.4 Android

Android is a collection of open source software used in mobile devices. The Android SDK provides the tools and API necessary to begin developing applications on the Android platform using the Java programming language. Android provides users with the opportunity to build and publish their own applications by providing an open development environment. Android treats all applications (native and third-party) as equals. Therefore, having such an open development environment requires security measures to be taken in order to protect the integrity of the Android platform and the privacy of its users. The standard IDE for Android is Eclipse which gives a wide development environment. Android supports multitasking, so that the user can run multiple applications simultaneously, thus enabling the user to check his/her mail while loading an other web page[10].

Android architecture

The architecture of an Android platform is provided below.

Applications and Widgets: These are the developed applications for the end user to use and access it[11].

Application Framework: Application development consists of a framework which supports the testing of the application code in addition to compiling and debugging. The Android Application framework consists of certain managers and providers such as Activity Manager, Content Providers, Resource manager, Location Manager, Notification Manager etc[11].

Libraries: The Android developing platform consists of several libraries which support several functionalities. These are installed into the hardware during manufacturing by the mobile manufacturer. Surface Manager, Media Framework, SQLite, OpenGLES, FreeType, Accelerometer Pattern Recorder And Recognizer Webkit, SGL,SSL are the main libraries. These are actually rather low level programs and are thus invoked by higher level programs[11].

Linux Kernel: Android runs on a Linux kernel and thus uses Linux for service managements whose details are, in fact, hidden from the users[11].

Android Run-time: This involves the Core Java Libraries and Dalvik Virtual machine (Android's version of java)[11].
### 3 Methodology

This section of the report will focus on the chosen methodology that solves the goals for the implementation of this project. As this project is based on the prediction of the next context data of a mobile device, this chapter will start by investigating different discrete context data that could fit into the chosen prediction model. The next section of this chapter will discuss the chosen method and alternative methods that could make the context prediction possible in the mobile devices. The chapter will end by selecting an appropriate verification method for measuring the accuracy of the selected prediction model.

#### 3.1 Discrete context data

This section of the chapter will discuss how to solve the first goal of this project. The prediction of the future state of a mobile device depends on the context data used in the model. Since a Markov chain only predicts finite values, this project will focus on discrete context data for predicting the future states of a mobile device. Cell ID and WiFi are possible discrete context data which could be considered as input for prediction by a Markov chain. The Markov chain model can be represented by a finite state machine and this discrete context data has to be used in that finite state machine for the prediction. The discrete context data being used in this project are described below.

##### 3.1.1 Identification of Cell ID

Mobile phones are always connected to a cell in a network and each cell has a unique Cell ID. Based on knowledge of this Cell ID, it is possible to determine the location of a user. The process of capturing a Cell ID is as follows. At first the Cell Tower propagates the ID to the mobile device. The device then stores that data in the Cell ID Database, which is then retrieved by the application and presented to the User/Application. This process is explained in figure 3.1 below.

![Figure 3.1: Getting Cell ID from a cell](image-url)
3.1.2 Getting WiFi from available networks

In a WiFi zone there are a number of wireless networks or WiFi hotspots which are available and which can be connected to a user's WiFi enabled device. This can be used as another input for the Markov chain model since they are also discrete values with a finite state. Each WiFi port has a MAC address based on the router which is unique in nature. This MAC address can be used as the identifier for that network and can make use of the Markov chain model.

3.2 Approach

The basic method in relation to the prediction, the next state of a mobile device has the following parameters. The prediction process starts with input parameters which will only consider discrete values such as cell ID or WiFi. The input values are then processed in the prediction engine in order to predict the next context. The predicted values are then stored in a database associated with the model. Finally, the values are propagated to the application, which provides the context prediction. The overall flow of the methodology of this project is illustrated in figure 3.2 below.

![Figure 3.2: The basic model](image)

3.3 Alternative models

This section of the chapter will discuss how to solve the second goal of this project. This will be achieved by investigating alternative methods for predicting the next future states of a device and by discussing the advantages and disadvantages from the Markov chain model in relation to the prediction of future states. In addition, the model should meet certain properties which are provided in the following section.
3.3.1 Model properties

In relation to the prediction of future state of a mobile device, the prediction model should meet certain properties. The use of discrete values is an important property that the prediction model should meet. It will be easy to predict whether or not the model is using discrete values as the input. Discrete values are also called finite values such as Cell id or WiFi. It is not possible to make a model with an infinite set of values such as location coordinates involving latitude and longitude. Since the prediction states are changing from one state to another because of an event or condition, the prediction model should also be based on a finite state machine concept. Time independence is the another property that should be met by the prediction model. This project cannot be time dependent because the prediction is not based on time but rather on a sequence or actions which are not time dependent. The prediction model's learning should be unsupervised in nature, which is the another property. This is because the mobile device does not have access to a training set. Finally, the prediction model should also be in an unhidden state because the states are randomly generated and the predicted state should be visible to the user.

The properties of all alternative prediction models are detailed in the following sections of this chapter.

3.3.2 Markov chain Model

Markov chain model can be used for predicting the next state of a mobile device. In order to predict the next state from this model, discrete input values are required. The input states are determined from the Cell ID/WiFi which are discrete in nature. A data set is created by the Markov chain model from these available discrete input data. The input data will change according to the location of the mobile device. This will affect the state changes, which causes the Markov chain model to create transition probabilities that are represented in a transition matrix and thus the future state is predicted. The probability of obtaining a new state from the previous states can be observed from the data sets of states from the finite state machine and Markov chains.

3.3.3 Discrete SEQ Model

Discrete Sequence Model [12] is a supervised learning technique. It is a sequence prediction model which predicts the next state in a sequence. This model is time independent and hence the discrete input values must be stored sequentially at different locations. The possible outputs are the same input discrete values that are used to create the sequence. The sub sequence can be repeated at different points in a sequence. They are known in advance and the model predicts which state is next in the sequence.

Since this model is a supervised learning technique, the model must have a supervisor for learning each state for every occasion. Since this project is only fo-
cused on unsupervised learning models for learning the discrete input states, this model is not efficient for predicting the future states.

### 3.3.4 Market Basket Analysis

Market Basket Analysis \[12\][13] is an unsupervised learning technique in which a temporal prediction method is adopted. This model will learn all the input values and then create a data-set. From the data-set, the model will discover the relationship between the learned input states and create a cluster for similar values. From this cluster, the model will predict the next state of the device. Since it is a temporal prediction technique, it only determines the relationships between items in a large data-set. It does not build a model from historical training data and hence it cannot implement a flexible prediction model.

### 3.3.5 Prediction by Partial Matching

Prediction-by-Partial Matching \[14\], also called prediction by the Markov model of order N, is a method to predict the next symbol depending on the previous symbols. This model is based on context modeling and prediction. This is a basic data compression technique in which the next text depends on the previous texts. Since this model is basically used as a text compression mechanism, it is not wise to use this model for predicting the next state of a mobile device. The text prediction techniques does not use discrete values such as Cell ID and WiFi for the prediction and hence it cannot be used in this project. Prediction-by-Partial Match algorithm (PPM) is a variant of a Markov model that could successfully predict music tracks.

### 3.3.6 Hidden Markov Model

Hidden Markov Model \[15\] is a random process that cannot be observable, which means a hidden state. However, it can only be observed through another set of random processes and this observed random process will only produce the sequence of observed symbols. To understand the concept of Hidden Markov Model, consider the 'Coin Toss' example. The user-A in a room with a hidden barrier in which the user cannot see what is happening. On the other side of the barrier is another user-B, who is performing a coin tossing experiment. The user-B will not provide any information in relation to the exact nature of the experiment nor provide information regarding the result of each coin flip. Thus, after a sequence of coin tossing experiment, the user-A can only observe the results of the coin tosses.

Since the Hidden Markov model has this hidden state property and the proposed model cannot work with hidden states,, this model cannot be used in predicting the future states of a mobile device.

### 3.3.7 Summary

The overall comparison of alternative models have been conducted based on the following properties which are necessary to meet for the desired prediction result. Discrete values describe whether the model is based on a finite number of
values such as Cell ID or WiFi that can be used as input. The finite state machine compares whether the prediction models possess a model of computation consisting of a set of input states, transition function, and an output state. The time independent function will check whether or not the prediction is based on time. Unsupervised learning is another parameter that must be checked. It describes the machine learning task in relation to inferring a function from unsupervised training data. Hidden state describes whether the states in the stochastic process are hidden or not to the user.

The overall comparison of the prediction models are summarized and listed below in the figure 3.3.

![Figure 3.3: Model comparison](image)

After comparing all alternative prediction models, it was decided that the Markov chain model does satisfy all the requirements for predicting future states of a mobile device. Therefore, the Markov chain model has been selected in order to solve the goal of predicting the next future state of a mobile device.

3.4 **Verification**

This section of the thesis describes how to verify whether the implemented model meets all the requirements for the proposed solution. For that it is necessary to check the following two parameters which are described in the sub sections.

3.4.1 **Approach to the goal**

To achieve the goal of obtaining the current state of a mobile device and its state transitions, a data set will be created to record the current state using sensors that are built into the smart phones. To predict the local future context of a mobile device, a model of a finite state machine will be used by means of discrete context data such as Cell ID and WiFi. Finally, the prediction model can be implemented on an Android based mobile phone by developing an application.
3.4.2 **Accuracy of the model**

The accuracy of the prediction of the next Cell ID is an important parameter that must be checked to determine the accuracy of the model. The accuracy of the prediction can be calculated by comparing the values from each prediction table. The values from each prediction table can be plotted in a graph against the result from the new prediction in the same area. For the evaluation of the result, some selected fixed Cell ID states can be considered. These selected states are then passed through the prediction model which provides an output for the verification. This can be achieved by observing the data-set and the resultant states obtained from the prediction model. This step will be repeated several times to compare the results and the frequently occurring states in a graph and check the accuracy of the prediction.

3.4.3 **Response time**

Calculating the execution time of the model is another important aspect which requires checking in the verification part. By implementing the timestamps, the time taken for the model to provide a successful prediction can be calculated. This can be achieved by implementing timestamps at each input and output section of the model. The results can be used in order to improve the prediction model by comparing each value from the time-stamp in each prediction.
4 Implementation

This chapter describes the implementation of the proposed solution. The solution was explored by taking different alternative solutions under consideration and the following solution was suggested.

4.1 Implementation overview

There are, in general, mainly four layers of implementation required in the suggested solution of this thesis. The overall layer in relation to the layer architecture of the implementation model is illustrated in the figure 4.1. Layer one is the sensor layer for input which will access the required sources as input from the real world. The new generation of smart phones have the required sensors in-built which are able to access the input sources. Layer two deals with the application layer which acts as the interface between the user and the program. Layer three is the main part of the implementation in which the prediction with regards to the discrete data will be processed. The discrete data will be processed with the Markov chain model. Layer four is the database layer, which saves the input discrete data and the data from the Markov chain prediction model.

![Figure 4.1: Implementation Overview](image-url)
4.2 Proposed Implementation

The proposed implementation has been categorized into 3 different parts. The first phase is the External-Real world which deals with real world entities such as a Cell phone tower, mobile phones and discrete data such as Cell ID or WiFi. The second phase is the Internal-Prediction world which deals with the actual implementation process. It has a prediction engine, current state/predicted states and a database in which to store all the data. The third phase is the Application which is used as an interface between the External-Real world and the Internal prediction world. The overall diagram explaining the proposed implementation model is given in figure 4.2 below.

![Proposed Implementation Diagram](image)

Figure 4.2: Proposed Implementation Diagram

4.2.1 External-Real World

The external-real world phase has 3 parameters, in the main and these are the cell phone tower, cell ids and mobile phone. Cell ID is the main discrete data used as the input in this project for the Markov chain model. All the mobile phones work on this Cell ID which is being obtained from a nearby Cell Phone Tower. As the mobile phone moves, the Cell Ids will also change accordingly.
4.2.2 Internal-Prediction World

The internal-prediction world phase is the main part of the implementation. It has a prediction engine, current state, predicted state and a database. The prediction engine performs all the prediction mechanism. The Markov chain is the prediction method adopted in this project. The current input Cell ID is considered as the current state in the prediction world. The input is processed in the prediction engine and then the output will be given as the predicted state or the next Cell ID state.

4.2.3 Application Phase

The third phase of the implementation is the application phase. It is the user interface which connects the external-real world and the internal-prediction world. The application phase will provide the user with the means to store the current Cell ID, display the probabilities of the stored Cell ID and thus predict the next cell that is to be used by the cell phone.

4.3 Entity-Relationship Model

A graphical representation of the entities and the relationship between the entities used in this project is given below in the Figure 4.3.

![E-R Diagram for mobileCellPrediction](image)

Figure 4.3: E-R Diagram

The functions are all entities used in the E-R diagram and are detailed in the following sections.
4.3.1 Predict Context

This is the output from the prediction engine after the processing of the input states such as Cell IDs in the prediction model. The predict context will provide the next future state of a mobile device. It is connected to the prediction engine via a predict Cell ID attribute.

4.3.2 GSM Cell Location

The GSM cell location is the function which will provide the location of the current state of the mobile device. It has two parameters namely the GSM location and Cell ID. This function is the main source for obtaining the inputs from the real world to the prediction engine.

4.3.3 Prediction Engine

The prediction engine is the main part of the project. It will access the input from the sensors and process the input in the prediction engine and provide results as the next context state. It has two parameters namely cell id and a unique cell id which is used for predicting the next future state.

4.3.4 State Prediction

The state prediction is a function which is the sub model of the prediction engine. It has three parameters which are Cell ID count, total Cell ID count and unique Cell ID count which are necessary for the context prediction. The state prediction function verifies the probability Cell ID function and predicts in the prediction engine.

4.3.5 Probability of Cell Ids

The probability of a cell id is the function that verifies the parameters from the state prediction and which calculates the Markov chain probability from the transition matrix. The transition matrix consists of two parameters which are multiply Matrix and next State.

4.3.6 Store Values

The input values, which are accessing from the sensors, are stored in the database under the function Store Values. This function has three parameters namely insert cell id, cell id count and delete cell id. The store values function is connected to the prediction engine via the store cell id information attribute.
4.4 Application Model

The overall screen shots of the implemented application are given in this section. The screen shot for the home page is given in figure 4.4. The home page has 3 options and these could be, showing the current Cell ID, the probability of the overall Cell IDs and the next Cell ID.

![Home page screenshot]

Figure 4.4: Home page
The screen shot that is showing the probability of all Cell ID is given below in figure 4.5. This page will show the total number of unique Cell IDs and the chances/probabilities associated with each Cell ID.

Figure 4.5: Probability Page
The screen shot is showing the next cell ID after calculating the probability from the Markov chain model and this is given below in figure 4.6.

![Next Cell ID page](image)

**Figure 4.6: Next Cell ID page**

### 4.5 Android prototype

As discussed in the previous sections, in order to implement the Markov chain algorithm process and random generation of cell IDs, a system of android application is proposed. The detailed explanation of the code snippet and processes are explained below.
The class file `mainactivity` module of the prototype includes three main functions which are responsible to read, get and store cell IDs. In order to obtain the current cell ID and local area code, android has some specific functions, `getCid()` and `getLac()`. The services of the telephony manager and cell location are retrieved using the android APIs and the network operator is identified. This will provide the information as to whether the current cell is a UMTS (3G) cell or not depending upon the padding numbers. After this, the cell id is retrieved and stored in the database. This entire process is coded in `storeCellIds.java` file. The snapshot showing the Calculate Cell ID list is given in figure 4.7 below.

```
public void callCellIDList() {
    String s = "";
    int ct = 0;
    for (int i = 0; i < uniquellid.size(); i++) {
        s = "";
        ct = 0;
        for (int j = 0; j < lst.size() - 1; j++) {
            if (uniquellid.get(i).equalsIgnoreCase(lst.get(j))) {
                s += lst.get(j) + 1 + ",";
                ct += 1;
            }
        }
        cellidcstarray.add(ct);
        cellidseachashmap.put(uniquellid.get(i), s.trim());
        //System.out.println("After = " + uniquellid.get(i) + " = " + s);
    }
    //System.out.println("ct size = " + cellidseachashmap.get(2));
}
```

Figure 4.7: Calculate cellIDlist

The database is crucial in this prototype as it stores the cell ids which is very useful in relation to comparison and prediction. The table consisting of the Cell ID and its mac number is stored, retrieved and modified whenever it is necessary. The cell Ids can be inserted using the method, `insertCellid(String CellId, String Mac)` which will update the cell id table in the database. The method, `getcellIds()` and `getUniqueCellIds()` are used to obtain the normal and unique cell Ids respectively. The method `calCellIDsList()` is used to calculate the number of Cell ID lists which is helpful for the Markov chain process.

The figure that showing the database class is given below in figure 4.8.
The class file, CalcProb, is responsible for handling the Markov chain process for cell id prediction and for the creation of the transition matrix. The method, getUniqCellID(), returns the cell id with a unique number and it retrieves from the database.

For the Markov chain algorithm process, it is necessary to multiply the initial probability of cell ids list by the transition matrix probability. This results in a probability of the future state for prediction, which is intuitively the result from the Markov chain algorithm.

The getInitialProb() method provides the probability of obtaining the most likely cell id from the existing current cell id list. As an example, if there are over 10 probabilities from a cell id list in relation to achieving the next state, and if any three are unique, then this result will be combined with the transitional matrix probability in order to obtain the future predicted state.
The figure showing the transitional probability matrix multiplication is given below. This probability estimated result is then compared with the random probability generator for the evaluation process. The results from the evaluation of the accuracy of the prediction model are explained below in the following sections.

```java
public static void main() {
    double sum = 0;
    double multiply[][] = new double[10][10];
    for (int x = 0; x < r; x++) {
        for (int c = 0; c < r; c++) {
            for (int k = 0; k < r; k++) {
                sum = sum + matrix[c][k] * array[k][d];
            }
            multiply[c][d] = sum;
            sum = 0;
        }
    }
    endresult = outputArray(multiply);
}
```

Figure 4.9: Matrix multiplication
5 Results

This thesis is focused on developing an application to store the current state of a mobile device and predict its future states. To achieve a better prediction result, a Markov chain model is used for predicting the future states. The context prediction approach has been implemented in Java and the result has been tested using Transition Matrix tables generated from the Markov chain algorithm and these tables are compared using Matlab software. This evaluation part of the project followed two different steps in order to evaluate the accuracy of the proposed prediction model.

Evaluation

The accuracy of the system is evaluated by comparing the future and previously predicted contexts. The prediction methodology will be fruitful with the usage of discrete context information. In order to achieve the evaluation, a simulation of a fixed Markov chain model with some known cell ID states (eight cell ID states) has been considered. This simulation of the evaluation method provides accurate results by comparing some set of cell IDs, each consisting of eight cell ID states with the probability. This intuitively results in the accuracy of the prediction model.

5.1 Creating Transaction Matrix Tables

The simulation process is conducted with the assistance of Matlab software by comparing the future and previously generated transition matrix tables. Each Markov chain prediction model has transition matrix tables that are generated from the Markov chain algorithm. A transition matrix table shows the entire probability distribution of the Markov chain prediction model. With transition matrix tables it becomes possible to evaluate the accuracy of the prediction of the future state of a mobile device. In relation to this, consideration was given to eight fixed cell ID states \{1,2,3,4,5,6,7,8\}. Different steps used for the evaluation and the probabilities for state changes are detailed in the following sections.

Step-1

As a first step towards checking the accuracy of the prediction model, an area which has eight different Cell ID states \{1,2,3,4,5,6,7,8\} is considered. The result of this is that it is possible for a user to travel to all eight places within the area at a time T1. For this purpose, a fixed Markov chain prediction model with eight known Cell ID states \{1,2,3,4,5,6,7,8\} is used. The probability distribution of user movements in that fixed area is generated in a transition matrix table which is displayed in figure 5.1. At this time, there is only one transition matrix probability distribution data in the Markov chain database which is gen-
erated at time T1. The transition matrix table of the probability distribution generated at the time T1 is given in figure 5.1.

<table>
<thead>
<tr>
<th></th>
<th>CELL ID-1</th>
<th>CELL ID-2</th>
<th>CELL ID-3</th>
<th>CELL ID-4</th>
<th>CELL ID-5</th>
<th>CELL ID-6</th>
<th>CELL ID-7</th>
<th>CELL ID-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELL ID-1</td>
<td>0.06</td>
<td>0.12</td>
<td>0.24</td>
<td>0.06</td>
<td>0.18</td>
<td>0.12</td>
<td>0</td>
<td>0.24</td>
</tr>
<tr>
<td>CELL ID-2</td>
<td>0.13</td>
<td>0.13</td>
<td>0.09</td>
<td>0.22</td>
<td>0.13</td>
<td>0.64</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>CELL ID-3</td>
<td>0.07</td>
<td>0.14</td>
<td>0.03</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
<td>0.17</td>
<td>0.14</td>
</tr>
<tr>
<td>CELL ID-4</td>
<td>0</td>
<td>0.16</td>
<td>0.24</td>
<td>0.68</td>
<td>0.64</td>
<td>0.2</td>
<td>0.24</td>
<td>0.04</td>
</tr>
<tr>
<td>CELL ID-5</td>
<td>0.04</td>
<td>0.12</td>
<td>0.24</td>
<td>0.2</td>
<td>0.16</td>
<td>0.64</td>
<td>0.04</td>
<td>0.16</td>
</tr>
<tr>
<td>CELL ID-6</td>
<td>0.14</td>
<td>0.09</td>
<td>0.18</td>
<td>0.18</td>
<td>0.09</td>
<td>0.14</td>
<td>0.05</td>
<td>0.14</td>
</tr>
<tr>
<td>CELL ID-7</td>
<td>0.04</td>
<td>0.04</td>
<td>0.09</td>
<td>0.09</td>
<td>0.13</td>
<td>0.17</td>
<td>0.13</td>
<td>0.3</td>
</tr>
<tr>
<td>CELL ID-8</td>
<td>0.17</td>
<td>0.09</td>
<td>0.14</td>
<td>0.06</td>
<td>0.11</td>
<td>0.66</td>
<td>0.11</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Figure 5.1: Transition Matrix table at time T1.

**Step-2**

In the second step of the evaluation, new user movements in the same area of fixed Cell Id states are considered and a new transition matrix is generated at time T2. This time, there is probability distribution data from two transition matrices, which is in the Markov chain database received from the Markov chain prediction mechanisms in step-1 and step-2. Thus, all the probability distribution data from the transition matrix table generated at time T2 is added to the old transition matrix data generated at time T1. The transition matrix table of the probability distribution generated at the time T2 is given in figure 5.2.

<table>
<thead>
<tr>
<th></th>
<th>CELL ID-1</th>
<th>CELL ID-2</th>
<th>CELL ID-3</th>
<th>CELL ID-4</th>
<th>CELL ID-5</th>
<th>CELL ID-6</th>
<th>CELL ID-7</th>
<th>CELL ID-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELL ID-1</td>
<td>0.08</td>
<td>0.15</td>
<td>0.27</td>
<td>0.12</td>
<td>0.12</td>
<td>0.68</td>
<td>0.04</td>
<td>0.15</td>
</tr>
<tr>
<td>CELL ID-2</td>
<td>0.17</td>
<td>0.14</td>
<td>0.06</td>
<td>0.22</td>
<td>0.11</td>
<td>0.66</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>CELL ID-3</td>
<td>0.04</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.11</td>
<td>0.2</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>CELL ID-4</td>
<td>0.07</td>
<td>0.19</td>
<td>0.17</td>
<td>0.1</td>
<td>0.1</td>
<td>0.17</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>CELL ID-5</td>
<td>0.05</td>
<td>0.08</td>
<td>0.21</td>
<td>0.21</td>
<td>0.16</td>
<td>0.65</td>
<td>0.13</td>
<td>0.11</td>
</tr>
<tr>
<td>CELL ID-6</td>
<td>0.13</td>
<td>0.07</td>
<td>0.17</td>
<td>0.2</td>
<td>0.1</td>
<td>0.13</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>CELL ID-7</td>
<td>0.05</td>
<td>0.07</td>
<td>0.1</td>
<td>0.07</td>
<td>0.2</td>
<td>0.15</td>
<td>0.15</td>
<td>0.22</td>
</tr>
<tr>
<td>CELL ID-8</td>
<td>0.15</td>
<td>0.1</td>
<td>0.15</td>
<td>0.1</td>
<td>0.1</td>
<td>0.65</td>
<td>0.12</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Figure 5.2: Transition Matrix table at time T2.
Step-3

In the third step of the evaluation, new user movements in the same area of fixed Cell Id states are considered and a new transition matrix is generated at a given time, T3. It is now the case that the probability distribution data exists for three transition matrices and these are in the Markov chain database received from the Markov chain prediction mechanisms in step1, 2 and 3. Thus, all the probability distribution data from the transition matrix table generated at time T3 is added to the old transition matrix data generated at times T1 and T2. The transition matrix table generated from the fixed Markov chain prediction model at time T3 is given in figure 5.3.

<table>
<thead>
<tr>
<th></th>
<th>CELL ID-1</th>
<th>CELL ID-2</th>
<th>CELL ID-3</th>
<th>CELL ID-4</th>
<th>CELL ID-5</th>
<th>CELL ID-6</th>
<th>CELL ID-7</th>
<th>CELL ID-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELL ID-1</td>
<td>0.07</td>
<td>0.1</td>
<td>0.24</td>
<td>0.1</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>CELL ID-2</td>
<td>0.18</td>
<td>0.11</td>
<td>0.07</td>
<td>0.25</td>
<td>0.09</td>
<td>0.05</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>CELL ID-3</td>
<td>0.07</td>
<td>0.14</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.14</td>
<td>0.17</td>
<td>0.12</td>
</tr>
<tr>
<td>CELL ID-4</td>
<td>0.06</td>
<td>0.15</td>
<td>0.17</td>
<td>0.08</td>
<td>0.13</td>
<td>0.19</td>
<td>0.15</td>
<td>0.08</td>
</tr>
<tr>
<td>CELL ID-5</td>
<td>0.09</td>
<td>0.08</td>
<td>0.2</td>
<td>0.18</td>
<td>0.16</td>
<td>0.08</td>
<td>0.12</td>
<td>0.1</td>
</tr>
<tr>
<td>CELL ID-6</td>
<td>0.11</td>
<td>0.11</td>
<td>0.16</td>
<td>0.18</td>
<td>0.11</td>
<td>0.09</td>
<td>0.13</td>
<td>0.11</td>
</tr>
<tr>
<td>CELL ID-7</td>
<td>0.1</td>
<td>0.06</td>
<td>0.1</td>
<td>0.08</td>
<td>0.17</td>
<td>0.13</td>
<td>0.13</td>
<td>0.23</td>
</tr>
<tr>
<td>CELL ID-8</td>
<td>0.16</td>
<td>0.11</td>
<td>0.12</td>
<td>0.11</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Figure 5.3: Transition Matrix table at time T3.

Step-4

In the fourth step of the evaluation, new user movements in the same area of fixed Cell Id states are considered and a new transition matrix is generated at a given time, T4. There is now probability distribution data for four transition matrices and this is in the Markov chain database received from the Markov chain prediction mechanisms from step1, 2, 3 and 4. Thus, all the probability distribution data from the transition matrix table generated at time T4 is added to the old transition matrix data generated at times T1, T2 and T3. The transition matrix table generated from the fixed Markov chain prediction model at time T4 is given in figure 5.4.
Step-5

In the fifth step of the evaluation, new user movements in the same area of fixed Cell Id states are considered and a new transition matrix is generated at a given time, T5. There is now probability distribution data from five transition matrices and these are in the Markov chain database received from the Markov chain prediction mechanisms from steps 1, 2, 3, 4 and 5. Thus, all the probability distribution data from the transition matrix table that are generated at time T5 is added with the old transition matrix data generated at times T1, T2, T3 and T4. The transition matrix table generated from the fixed Markov chain prediction model at time T5 is given in figure 5.5.

![Transition Matrix table at time T4.](figure.png)

![Transition Matrix table at time T5.](figure.png)

Now we have five different transition matrix tables that have been generated at five different times T1, T2, T3, T4, T5. In order to check the accuracy of the proposed prediction model, it is necessary to compare the values received from all five different transition matrix tables. Each row in the transition matrix represents the probability distribution of each cell id to the same and other seven cell ids. Thus each row represents that particular cell id's probability measurement in order to change from that current cell id to the next (total 8 cell ids in this case) cell ids. The transition table comparison is conducted with the assist-
5.2 Transition Matrix comparison

In this section, the probability results of each cell id from all the transition matrices are plotted in a graph and compared using Matlab. The probability results of the transition matrix are generated by considering a particular area that has eight different fixed cell ids \{1,2,3,4,5,6,7,8\}. The prediction model accuracy is evaluated from the results of a comparison of all eight cell ids that are taken from the five transition matrix. The graphs obtained after the comparison are given below.

The X-axis represents the total number of cell ID states. In this case, there are eight fixed states \{1,2,3,4,5,6,7,8\} being considered. The Y-axis represents the probability measurements between 0 and 1. Firstly the probability distribution graph of Cell Id-1 from all five transition matrices that are generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison graph of Cell Id-1 from all five transition matrix is given in figure 5.6.

![Figure 5.6: Transition Matrix comparison for Cell ID-1](image)

Now, the probability distribution graph of Cell Id-2 from all five transition matrices that are generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-2 from all five transition matrix is given in figure 5.7.
Now, the probability distribution graph of Cell Id-3 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-3 from all five transition matrix is given in figure 5.8.

Now, the probability distribution graph of Cell Id-4 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-4 from all five transition matrix is given in figure 5.9.
Figure 5.9: Transition Matrix comparison for Cell ID-4

Now, the probability distribution graph of Cell Id-5 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-5 from all five transition matrix is given in figure 5.10.

![Transition Matrix comparison for Cell ID-5](image)

Figure 5.10: Transition Matrix comparison for Cell ID-5

Now, the probability distribution graph of Cell Id-6 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-6 from all five transition matrix is given in figure 5.11.

![Transition Matrix comparison for Cell ID-6](image)

Figure 5.11: Transition Matrix comparison for Cell ID-6

Now, the probability distribution graph of Cell Id-7 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-7 from all five transition matrix is given in figure 5.12.
Now, the probability distribution graph of Cell Id-8 from all five transition matrices generated at five different time intervals T1, T2, T3, T4, T5 is plotted. The probability result comparison of Cell Id-8 from all five transition matrix is given in figure 5.13.

From this overall comparison of the results, it clearly shows that the accuracy of the probability distribution in each transition matrix is satisfyingly similar. The eight Cell IDs from the five transition matrices show a good probability distribution mechanism. Since the variation of probabilities are minimal it is possible to state that this Markov chain model can be used in the real world scenario.
6 Conclusion

This project's overall concrete goals were, “To predict the future local context of a mobile device” and “To implement an idea of designing Model context as a finite state machine and then, by generating a Markov chain, to predict the next context state”. This paper presented a model for storing and retrieving the discrete context data for predicting the future state of a mobile device. The primary focus of this thesis was to study the existing context prediction models and then to propose a solution based on the best existing model and then test it with a simple implementation.

To create a data set and to record the current state of a mobile device, Cell ID was used as input from the sensors in mobile devices. Then a prediction model is selected after studying different context prediction models. This selected prediction model is used to predict the future local context of a mobile device. Finally the accuracy of the prediction model is checked by comparing the predicted contexts with previously predicted contexts.

6.1 Problem statements

The problem statement asked to propose a system to create a data set by recording the current state of a device and its state transitions. In addition the requirement was to provide a prediction model which is based on discrete context data to predict the future local context of a mobile device.

The first question is solved by creating a data set and recording the current state of a mobile device with its state transitions. This is achieved with the assistance of sensors that are built into the new generation of smart phones. The second question is answered with the assistance of a Markov chain by creating a prediction model to predict the next context of a mobile device. The accuracy of the system has been checked by comparing the probability results from transition matrix tables. Finally the result has been plotted in a graph using Matlab and the result has been evaluated. This is detailed in the results chapter. These results have also been verified by running the developed application on an android mobile phone.

6.2 Results

The evaluation results shows that the accuracy of the prediction model is good as they are satisfyingly similar for each transition matrix table. This evaluation is conducted using eight sets of fixed cell id states and their associated probability distribution transition matrix model. The eight cell ids from the five transition matrices all display a good probability distribution. Since the variation of probabilities is minimal, that this Markov chain model can be used in the real world scenario. The section 5.2 shows a comparison of the accuracy of the prediction model.
6.3 Future work

All the conducted tests show that the accuracy of the prediction is increased after the learning associated with each input state. Therefore it is possible to say that the implemented prediction model is working with the real world context data in the real time scenario. With this approach as a future work, it is possible to send this model to different users in the media sense platform. It can be used to share the predicted context information to find each user's next location.

Another suggested future work is to enable an illusion of evolving context information whenever a source is temporarily unavailable. This assist the user to access the resource even if there is an outage, which makes an illusion that the current resources being accessed are still active.
References

http://www.ssw.uni-linz.ac.at/Teaching/PhDTheses/Beer/

http://www.wpnc.net/fileadmin/WPNC05/Proceedings/Definition_and_Implementation_of_Context_Information.pdf

[3] AT&T Developer program “Cell ID”,
http://developer.att.com/developer/tier2page.jsp?passedItemId=3100144


http://students.ee.sun.ac.za/~gshmaritz/gsmfordummies/arch.shtml

[6] Webopedia, ”WiFi “,
http://www.webopedia.com/TERM/W/Wi_Fi.html

http://en.kioskea.net/contents/wifi/wifimodes.php3

http://en.wikipedia.org/wiki/Markov_chain


http://en.wikipedia.org/wiki/Android%28operating_system%29

http://www.android-app-market.com/android-architecture.html

[12] Ingrid E. Burbey “Predicting Future Locations And Arrival Times of Individuals”


Mobile phone context prediction using Markov chains
Emil Skariah

Introduction to Hidden Markov Models.pdf
Appendix A: Documentation of own developed program code

The main program code will be available upon request through e-mail. Send request to the e-mail “emsk1003@student.miun.se”.

The MatLab code for plotting the graph

The MatLab code that are used to plot the graph of the results are shown below;

clear all;

% Give the number of nodes and the inputs for each nodes to the function

n=4;

prob_markov=zeros(n*n);

prob_random=zeros(n*n);

%For Client1

prob_markov(1) =0 ;
prob_markov(2) =0.25 ;
prob_markov(3) =0.5 ;
prob_markov(4) =0.25 ;
prob_markov(5) =1 ;
prob_markov(6) =0 ;
prob_markov(7) =0 ;
prob_markov(8) =0 ;
prob_markov(9) =0 ;
prob_markov(10) =0.125 ;
prob_markov(11) =0.625 ;
Mobile phone context prediction using Markov chains
Emil Skariah

prob_markov(12) = 0.25;
prob_markov(13) = 0.2;
prob_markov(14) = 0.2;
prob_markov(15) = 0.2;
prob_markov(16) = 0.4;

% For Client 1
prob_random(1) = 0;
prob_random(2) = 0;
prob_random(3) = 1;
prob_random(4) = 0;
prob_random(5) = 0.33;
prob_random(6) = 0;
prob_random(7) = 0.67;
prob_random(8) = 0;
prob_random(9) = 0;
prob_random(10) = 0.22;
prob_random(11) = 0.56;
prob_random(12) = 0.22;
prob_random(13) = 0.2;
prob_random(14) = 0.2;
prob_random(15) = 0;
prob_random(16) = 0.6;
save output.mat;

clear all;

load output.mat;

i=1:16;

plot(i, prob_markov(i), '-b*',i, prob_random(i), '-ro');

hleg1 = legend('prob markov','prob random');

xlabel('Number of States');ylabel('Probability');

title('Prob of Markov Chain');