CHALMERS

Obtaining Origin/Destination-matrices from cellular network data

Master's Thesis in Engineering Mathematics

Erik Mellegård

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology
Gothenburg, Sweden 2011

Master's Thesis 2011:

Abstract

Mobile network operators are collecting a lot of data on people's calls, like who they call, for how long they call etc., that are stored for billing and network purposes. Among that data, there is also information on which base station they are connected to, something that could be used to obtain valuable information about people's movements. The main reason that this is not being done today is that the operators are afraid of what would happen if someone would mistreat the data and use it to track people. This thesis presents a method for obtaining Origin/Destination-matrices from the mobile network data in a way that keeps the individuals' privacy. Since the operators are reluctant to let us use any real data, the method has been applied to synthetic data and some call data records. The method is applied to each individual separately, which makes it possible to run it in parallel on a cluster of computers, something that is very important when having months of data on million of users. Running the algorithm on call data records, this thesis shows that it is feasible to obtain Origin/Destination-matrices from mobile network data.

Acknowledgements

Without the help and support of a number of people, the making of this thesis would not have been possible. Most thanks I owe to my supervisor at Ericsson, **Simon Moritz**, who has given me a lot of help and feedback. Other people at Ericsson supporting me include **Apostolos Georgakis**, **Muhamed Zahoor** and **Marika Stålnacke**. I would also like to thank the members of the Consider8 project working at SICS; **Olof Görnerup**, **Pedro Sanches** and **Martin Nordström**.

Working at Ericsson would not have been as fun as it was without you, **Jonathan Magnusson**, **Viktor Kärnstrand** and **Abhiroop Gupta**.

Erik Mellegård
Göteborg
October 4, 2011

Contents

List of Figures

1

Introduction

This thesis is part of a project called Consider8. Consider8 is a collaboration between Ericsson AB and Swedish Institute of Computer Science (SICS) that aims at finding human mobility patterns from mobile network data.

When doing all sorts of traffic planning, it is important to have information on people's movements. One of the more fundamental kind of information you can have on people's movements is what is called Origin/Destination-matrices (O/D-matrices). An O/D-matrix is a list of places and how many people travelled from place A to place B (within a given time frame). A simple example of an O/D-matrix can be seen in table 1.1. Even though O/D-matrices seem very simple, they are actually quite useful in many applications. It is also remarkably hard to obtain accurate O/D-matrices that are up-to-date. Today, O/D-matrices are produced in one of two ways, either by surveys or by using measurements of traffic link counts, from cameras and magnetic sensors positioned along the road.

Table 1.1: An example of what an O/D-matrix looks like. From this matrix one can see that three people have travelled between Kista and Solna.

	Kista	Solna	Bromma	Gamla stan	Norrmalm
Kista	-	3	2	9	7
Solna	14	-	12	6	4
Bromma	5	2	-	5	2
Gamla stan	1	2	9	-	7
Norrmalm	4	3	7	9	-

1.1 Traffic link counts

Since the hardware required for measuring traffic counts is relatively inexpensive today, that technique is mostly used today. Usually, a lot of different O/D matrices can reproduce the same traffic counts; therefore the problem of deducing the O/D-matrix from traffic link counts is underdetermined. There are a lot of different techniques for finding the most likely O/D matrix from the link traffic counts, for example; maximum likelihood, generalized least square and Bayesian inference [13]. This way of estimating the O/D-matrix will only give you an estimation on the roads on which you have measuring equipment. Furthermore, the equipment is expensive and requires a lot of maintenance.

1.2 Surveys

Household surveys are conducted from time to time on a country level. People are asked to fill in forms, answering questions about where and how they travel. Household surveys can produce good O/D-estimation matrices if a lot of people participate. The problem is that people might not remember correctly when they fill in their logs and they might have reasons for not being honest when doing so. Going through all these logs are expensive, which means that these surveys are not conducted on a regular basis, resulting in O/D-matrices that are often to old to reflect recent changes in peoples movements.

2

Background

THE MOBILE CARRIERS around the world collect a lot of information on peoples calls as part of their normal operation, like who they call, for how long they call etc. This is collected for billing and network purposes. Among the information the carriers collect, there is information on which base station the phone is connected to during the call.

2.1 Data

When a phone is in active mode, either in a call or sending data, the base station the phone is connected to is logged twice a second in the network carrier systems. When the phone is in idle mode, the information on which base station the phone is connected to is only logged about once an hour (this vary between carriers, but is usually between 20 minutes and 2 hours). Among this data, there is information on the cell id of the base station the phone is connected to and a time stamp. This kind of data is what the method described in this thesis has been developed for.

2.1.1 Available data

The mobile operators are very reluctant to release any data of the kind described above for privacy reasons. However, Ericsson has access to some call data records. In addition to that data, some data has been collected from an Android application and some synthetic data has been generated. The three sets of data are described below.

Call data records

The first set of data is call data records from a mobile network operator. Call data records contains data on all outgoing calls that has been made from a network operator, including time and cell id at the start of every call. The call data records used in this thesis contains data from two million subscribers during one month's time. Unfortunately the data does4 not contain any passive mode information or any information on which base station the phone has been connected to during the call, which in reality would also be consider as input information.

Android application, installed on Ericsson and Swedish Institute of Computer Scien

Application data

The second set of data has been collected using an Android application, installed on Ericsson and Swedish Institute of Computer Science (SICS) employees' phones. This application collects information on the base station the phone is connected to, but also the GPS-position. This data can therefore be used to compare the results from the method with what we can manually deduce from the GPS data. The application collects information on which cell the phone is connected to at all times, meaning that this data differs from the data collected in the mobile operators' networks. Therefore, the data has to be modified to look like network data before using it to test and evaluate the method.

Synthetic data

The third set of data is synthetic data created by Viktor Kärnstrand, who worked at Ericsson during this summer. The method generating this data by creating an O/D-matrix from demographic data and then finds the closest route between all points in the O/D-matrix using an A^*-algorithm. From these routes, synthetic network data and GPS data are created. Since there is information on the actual O/D-matrix used to create the network data, this data be used to evaluate methods developed for obtaining O/D-matrices from mobile network data.

2.2 Mobile networks

A mobile network is organized in an hierarchical structure, with what is called base transceiver station (BTS) as the lowest component in the hierarchy (see appendix C.1). The area a BTS covers is usually called a cell. Each cell has a cell id that is unique within a certain location are code (LAC) for a specified operator and country. In this thesis it will be important to know the location of cell from its cell id (and its LAC, country and operator). There are a few APIs online from where you can get the longitude and latitude of a large part of the cells around the world, for example location-api [5] , open cell id [7] and Ericsson Labs [1].

Cell Id sources

2.3 Reverse geocoding APIs

Reverse geocoding is any process that puts a name or a place to a point (i.e. a longitude and latitude). This includes everything from just identifying the country a certain point belongs to, to identifying what road is closest to that point. There are a few freely available reverse geocoding APIs available online, for example the Google Geocoding API [3] and Openstreetmap [6]. In this thesis, openstreetmap has been used to find all places with one of the tags: ”city”, ”town”, ”village” and ”suburb” in the areas of interest.

3

Related work

Several projects have used cellular network data to analyze traffic in real time. This is usually achieved using cell handover information from active phones to calculate the current speed on a road network. The purpose of this is to be able to give the drivers information on the current speed and congestions on the route they are traveling. Mobile network data as input to traffic analysis has been used by TomTom [12], in the STRESS project [15] and by AirSage [10] among other. Since their work is not that related to the aim of this thesis, their approach will not be explained in any more detail.

3.1 Simulations

There have not been made that many attempts at analyzing traffic in a more long-term perspective from mobile network data, for example obtaining O/D-matrices. Some simulations and studies have been made on the subject, for example Caceres N., Wideberg J.P. and Benitez F.G [17] concluded from their simulation that it is possible to extract O/D-matrices from their simulated cellular network data. Sohn Keemin and Kim Daehyun [22] concluded the same thing in their paper. Their simulation also showed how the result depends on some network characteristics, such as network penetration (proportion of people using the network in question) and the cell size. They both used handover information to estimate the link traffic volume and then used that to estimate O/D-matrices. One of the drawback with this approach is that it can only use data from phones in active mode (when it is in a call or sending/receiving data). It can also only give an approximate solution, since the problem of finding O/D-matrices from traffic link counts is underdetermined. In their simulation they also assumed that a phone is always connected to the closest base station, which is a simplification which is not true. From the data we have access to we can see that a phone can switch between several base stations even while being stationary. This kind of behavior of the network would make it very hard to obtain O/D-matrices with the approach they used.

a phone can switch between several base stations even wh

3.2 Using real data

Some studies on real data have been made in smaller scale, to determining the feasibility of acquiring O/D-matrices from a cellular network. Ahas Rein, Aasa Anto, Silm Siiri and Tiri Margus [14] published a case-study on 277 persons living outside the city of Tallinn. The study was conducted over a time period of 8 days in 2006. The current cell each of the phones where connected to where recorded once every 15 minutes. With this information they were able to determine where people live and where they work and how much time they spend at each place.

early study 2006.

Solomon Charles, Yehuda Gur J, Shlomo Bekhor and Leonid Kheifits published an article describing how cell phone data can be used to help transport planning in Israel [23]. From the network carrier Orange Salomon et al. obtained data from 160 000 persons. One week from each person, spread out over 16 weeks. The 2200 antennas Orange had in the country was were divided into 600 traffic analysis zones. A person was defined as stationary when more than 20 minutes were spent in the same zone. Their were unable to obtain an accurate O/D-matrix down to the zone-level, but they claim to have been able to obtain an accurate O/D-matrix on a district level (there are 34 districts in Israel).

Israel

3.3 Other related work

The project ”Geographic Privacy-aware Knowledge Discover and Delivery” (GeoPKDD) [2] may be one of the more ambitious projects of mapping human mobility. It was a collaboration between over 40 persons from around Europe that ended in 2008. They publiched a lot of papers regarding how to find out what routes people travel in privacy-aware manner, using mobile network data as well as GPS data. They did not, however, do much work on O/D-matrices.

Kang Jong Hee, Welbourne William, Stewart Benjamin and Borriello Gaetano [20] and Asakura Yasuo and Eiji Hato [16] have proposed algorithms for deciding if a user is stationary or moving from positioning data. They used GPS data collected on the mobile device, as opposed to cell id data collected in the cellular network. Using cell id data is much harder since the accuracy of the measurements is lower, the measurements are not evenly spaced in time and they are usually less dense in time.

4

Use cases

O/D-matrices by themselves has a lot of applications, but could also be used together with other kinds of information on people's movements, or together with demographic data. Following are some examples of what O/D-matrices can be used for.

4.1 Building new metro and mono rail in India

India is a country with many fast growing cities. One of them, Chennai, is planning to build metros and mono rails to support the inhabitants need of transportation. The aim is to build metros connecting the most important hubs of Chennai, with mono rails connecting the hubs to less travelled parts of Chennai. In order to make informed decisions on where to put the metros and the mono rails, accurate O/D-matrices are needed. The information on how people travel is already inside the operators' networks, but the methods for extracting information is not yet in place.

4.2 Traffic planning

Accurate O/D-estimations is essential when planning new infrastructure. Measuring traffic link flows is a good way of finding out what roads are inadequate for the amount of traffic traveling on them. However, the best solution might not always be to build a bigger road, but instead build a new road along another stretch. In order to know if a new road might lead to less traffic on an existing road, you need to know between what places people travel and when they travel. As an example, there has been talk about building a new road connecting Nacka with the northern parts of Stockholm, to divert traffic from the center of Stockholm and the heavily trafficked Söderleden. In order to know if this road will divert traffic from the center of Stockholm, you need to know how many of the people leaving from Nacka will travel to the northern parts of Stockholm.

4.3 Bridges and overpasses

Today, the decision of which crossings should be replaced by overpasses in India is based on information collected by people manually counting cars in crossroads. This is costly and takes a long time. Accurate O/D-matrices could be used to determine what crossings are the most important ones.

4.4 Unprotected railroad crossings

A problem many countries are facing is the large number of unprotected railroad crossings [19]. India has the largest railway network in the world with over 30000 level crossings, nearly 15000 of them being unmanned [9]. Since it is too expensive to make all of the crossings safer, it would of interest to have information on how often each crossing is crossed, to know which crossings to focus on. Since it is too costly to place a measuring device or a person at every crossing, O/D-matrices may be of use, even though it might not give a definite answer to this question.

5

Methodology

5.1 Initial study

As a first step in the work on this thesis, literature and commercial applications were studied in order to deduce what approaches to O/D-estimation had already been conducted and how to best build on these and make them better.

When this thesis work started, the only data available was the data collected with the android application. This data necessarily shares some characteristics with real data, even though the sample frequency is not the same as in real data. This data was modified to look like data obtained from a mobile network and then studied in Matlab to determine the best way of obtaining O/D-matrices from it.

5.2 Method development

When a rough picture of what the final framework would look like were becoming clearer, the methods were implemented in Java instead. A paper was written and sent to the ”International Workshop on Spatial and Spatiotemporal Data Mining” [4] and a patent application protecting the method developed was filed.

5.3 Implementation and analysis

When it became clear that the method could run in parallel, it was altered so that it could run on a Hadoop cluster. This allows for faster calculations which is of great importance if this should ever be implemented in a commercial application, since the application would have to be able to handle data on millions of mobile users, amounting to several terabytes.

During the summer of 2011 Viktor Kärnstrand worked at Ericsson. His main goal was to create synthetic data with the same characteristics as real network data that could be used by me and other people working with the Consider8 project to evaluate our methods. When everything else was in place, this data, together with the android application data, was used to evaluate the method developed in this thesis.

6

Framework & Results

This section describes the method developed for obtaining O/D-matrices, how it has been implemented and how to evaluate it. It also describes the program developed for visualizing the O/D-matrices.

6.1 Framework

A schematic overview of the method used for calculating O/D-matrices can be seen in figure 6.1. After converting from cell id to location using a cell id database, each user is analyzed separately. The trips each user has made is calculated using only positioning data, no knowledge of the location of any cities is used at this step. After that, a reverse geocoding API is used to identify the places the user has been traveling to/from. The steps of the method (depicted in 6.1) are explained below.

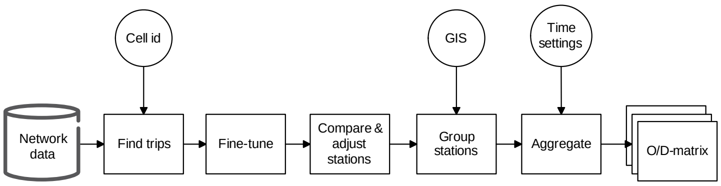


Figure 6.1: An overview of the method for obtaining O/D-matrices.

6.1.1 Convert from cell id

The input data only contains information on which base station the phone was connected to, not the location of that base station. Fortunately, there are free APIs online that you can query for that location of base stations (or, equivalently, cells). They collect the location of cells using applications installed on mobile devices, registering the gps-position and the cell it is currently connected to. Therefore, they do not have the actual place of the base station, but rather the average location the mobile devices have been at when connected to that base station (which actually suits our need better).

There are a few public cell id databases online, for example Ericsson labs [1], opencellid [7] and location-api [5]. Location-api is the api that has the location of the most cells, at the moment of writing they have the location of more than 17 million cells around the world. Figure 6.2 shows the cells in Sweden that has been used in this thesis (there are more, but they were not needed, so the location of them has not been downloaded).

6.1.2 Finding trips

The first step towards finding movement patterns is to determine if an individual is moving or not. This might seem like an easy task, but the data is very sparse and there are a lot of fluctuations occurring even if an individual is perfectly still. In the data collected by Ericsson and SICS, a stationary phone has been noticed to jump between up to 5-6 base stations with only a few seconds on each base station. In [20] they define an easy algorithm for determining if a person is stationary or moving. The method used in this thesis is similar, with a few modifications. A basic description of the algorithm can be seen in figure 6.3. As input, the method takes all the location data points on a subscriber. The output from the method is a list of all the points the subscriber has been stationary at. These stations are then used to find all the trips made by a subscriber. The algorithm has three parameters, two spatial and one temporal, whose optimum values have to be decided empirically. Figure 6.4 shows an example of how the algorithm works.

Parameters

The 3 algorithm PARAMETERS!

The algorithm has three parameters; two of them are spatial parameters and one is a temporal parameter. The first spatial parameter defines the longest distance a point can be from the previous point to be classified as stationary (in case the previous point was classified as moving). The second spatial parameter defines the maximal distance a point can be from a mean in order for the new point to be classified as stationary (in case the previous point was stationary). The time parameter defines the minimum time a person has to be stationary in order to count it as a station.

These parameters could either be fixed or dynamic. The main argument for making them dynamic is that the optimal values for the parameters might differ between urban and rural areas.

Fine-tune

The method for finding trips described above is good at finding the origin and the destination, but unfortunately has a harder time of finding the exact time of departure and arrival. This is a result from the construction of the algorithm, and from the fact that the data is very sparse in time. One way of improving the accuracy of the arrival

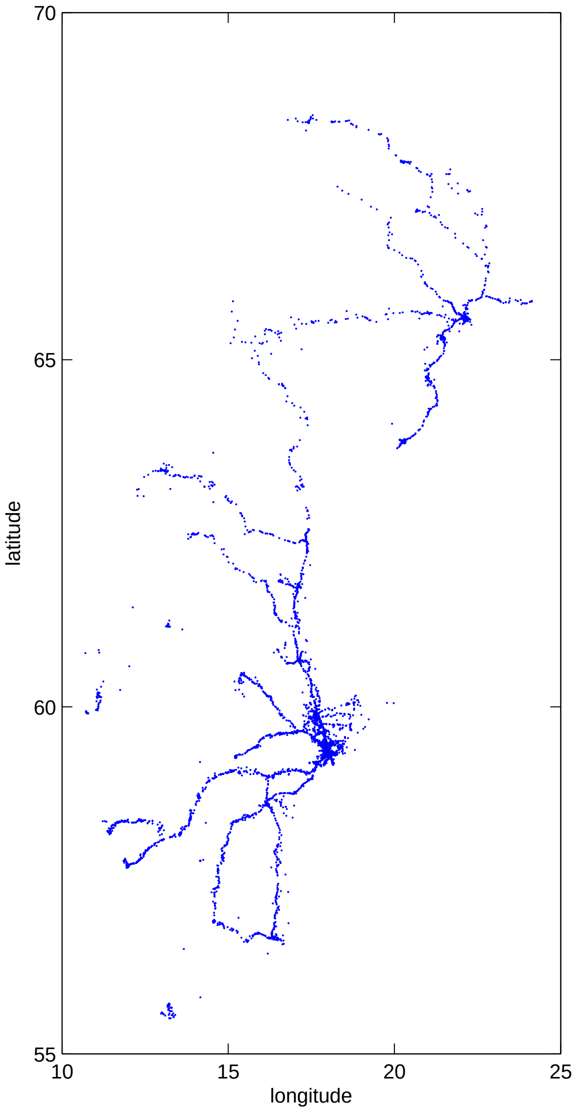


Figure 6.2: All the cells in Sweden that have been used in this thesis.

time would be to do a linear extrapolation from the points between the stations, thereby getting a better estimation of the arrival time.

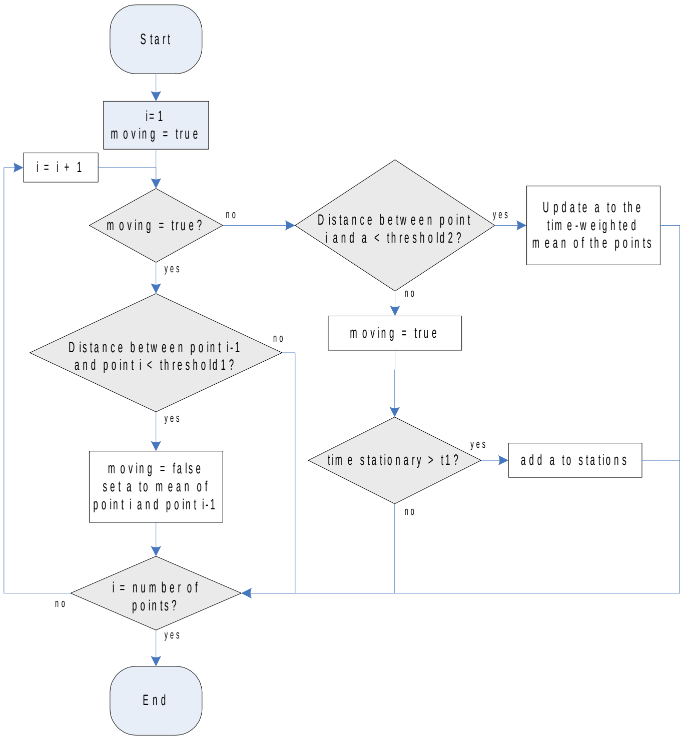

Start
i=1
moving = true
i = i + 1
moving = true?
no
yes
Distance between point i and a < threshold2?
yes
Update a to the time-weighted mean of the points
no
moving = true
Distance between point i-1 and point i < threshold1?
no
yes
moving = false
set a to mean of point i and point i-1
time stationary > t1?
yes
add a to stations
no
i = number of points?
no
yes
End

Figure 6.3: Flowchart describing how to find trips made by an individual

6.1.3 Compare and adjust stations

Since the data is very sparse, a station identified by the algorithm may contain as few as two points. This may result in a bad accuracy of the station. However, a person is likely to visit a certain place more than once. This can be used to get a better approximation of the exact location a user was stationary at. In this thesis a clustering algorithm called mean-shift clustering [21] has been used to cluster the stations. An explanation of how this algorithm is used in this thesis is described in appendix A. The reason for choosing the mean shift clustering algorithm is that it is easily implemented and it does not require any prior knowledge of the number of clusters. An example of how the algorithm clusters a set of points can be seen in figure 6.5. The figure also displays the name of the place each cluster belongs to, which is not known at this stage of the algorithm.

6.1.4 Group stations using open street map

For each trip, the location of the origin and the location of the destination are mapped to places defined by Open Street Map [6]. All the places in Sweden can be seen in figure 6.6. After this step, all trips have a name on both the origin and the destination, as can be seen in figure 6.5. Now, a matrix can be created from the trips, with the element at index i,j representing the number of trips from place with index i to the place with index j.

6.1.5 Aggregate

The O/D-estimation can either be aggregated into a big O/D-matrix or many O/D-matrices representing different times. The trips can be grouped by month, weekday/holiday, or by what time of the day they were made. A combination of those is probably the most interesting. In this thesis, a separate O/D-matrix has been created for each hour of the day and by what month the trip was made. This means that for three months data, $3 \cdot 24 = 72$ O/D-matrices would be created.

6.2 Implementation

swarm optimization algorithm

The framework described above has been implemented in Java with location-api [5] as cell-id database and Open Street Map [3] as reverse geocoding API. The parameters in the algorithm for finding the trips has been implemented as stationary. In order to find the optimal values for the parameter, a swarm optimization algorithm has been used minimize the error between the trips calculated by the program and the trips manually observed in the data. Since the data from the android application have information on the GPS-position, this data has been used for this optimization. The step called fine-tune in the framework has not been implemented.

6.2.1 Architecture

Since every user is evaluated separately, the calculations can be made in parallel on a cluster of computers. In this implementation, Hadoop MapReduce has been used. It is a framework for doing calculations on a cluster of computers developed by Google in 2004. In short, MapReduce requires two user defined functions, a ”mapper” and a ”reducer”, both of them using key-value pairs as both input and output. A more thorough description of how MapReduce work can be found in appendix B. The procedure of finding O/D-matrices is split up into four parts; find unique cell-ids; look up location of cells; sort data; find O/D-matrices (the main part). These steps will be explained below.

Find unique cell-ids

procedure of finding ODMs

The input data this method uses only contains the cell id the user is connected to, not the location of the cell. Fortunately, there are a lot of free APIs online that you can query for the location of a cell from a cell id. Querying such an API more than once for the same cell id would be unnecessary and the APIs usually have restrictions on how many queries you are allowed to make. Therefore, the first step of finding O/D-matrices uses MapReduce to find all unique cell ids in the data, so that the location of each unique cell only has to be queried for once.

Look up location of cells

In this step, the location of all unique cells are downloaded from a cell id database. In this project, location-api [5] has been used.

Order

~~Sort~~ data

The input data is typically sorted by time and not by user. A rather simple MapReduce program sorts all the data by user, and also adds location, collected in the previous step, to all data points.

Find O/D-matrices

This is obviously the main part of the program. It uses MapReduce to run the algorithm described in 6.1. An explanation of how it is implemented is provided in appendix B.

6.3 Visualization

Even though the O/D-matrices is the main result of this thesis, they are rather boring to look at and hard to get any real knowledge from. Therefore, a program for visualizing the O/D-matrices has been developed. The program is just an example of how you can use the O/D-matrices to get more knowledge from them then by just looking at a list of numbers.

A Tomcat server was used to run the method of obtaining O/D-matrices described above and to return the O/D-matrices in xml-format. A client running javascript and using the Google Earth web plugin was created, displaying the O/D-matrices as shown in 6.7. A client was also created displaying the O/D-matrices in Google Maps 6.8. Since both of these were rather slow when a lot of points were added to them, a third alternative were created, displaying the O/D-matrices in a java application instead. The java application is also capable of showing how each individual has moved and can be used to evaluate how well the method called ”finding trips” works.

6.4 Results

6.4.1 Finding trips

The method ”finding trips” correctly classifies all the trips made in the simulated gps data, where the time stationary in both the origin and destination is greater then around 10 minutes (this time depends on value of the time parameter used in the method, which in this case is set to 10 minutes).

Values

The optimal values for the parameters used in the algorithm called ”Finding trips” were calculated using a swarm optimization algorithm running on the data collected by the Android application. The optimal values obtained was about 1 km for the spatial parameters and about 10 minutes for the time parameter.

Algorithm parameters values10 minutes1 km

6.4.2 Simulated data

The simulated data used to evaluate the method contains data on 513 users for about 3 hours time. The first run showed that the method could only find 775 out of 1985 trips. When investigating why this was, it turned out that the simulated data only started saving data points for an agent from when he first started moving until he stopped moving for the last time. The method for obtaining O/D-matrices will then not detect that the agent has been stationary at those places. Taking this into account, it turns out that the method is capable of finding 775 out of 959 trips. Investigating this further, one finds that in the trips the method is not capable of finding, the agent has been stationary for less than around 10 minutes at one of the stations.

6.4.3 Real data

COMPARE: the distribution of trips over the day

It is hard to evaluate the real data, since there are no O/D-matrices to compare the results to. However, you can study how many trips each person makes and how they are distributed over the day. This does not give a definite answer to whether or not the method actually does what it is supposed to do, but it gives a hint at the answer. In figure 6.9 the distribution of trips over the day obtained from the real data set is plotted. In total there are 867209 trips, which is very low considering that it is data from roughly two million users over one months time. Of course, this data is only call data records, which is much less dense in time compared to the data the method was written for. It should also be mentioned that location-api only had the location of less than 10% of the cells in the country the call data records originated from, making this analysis even harder. Still, from figure 6.9, one can draw the conclusion that the distribution of the trips over the day seems reasonable, which is an indication that the method is working.

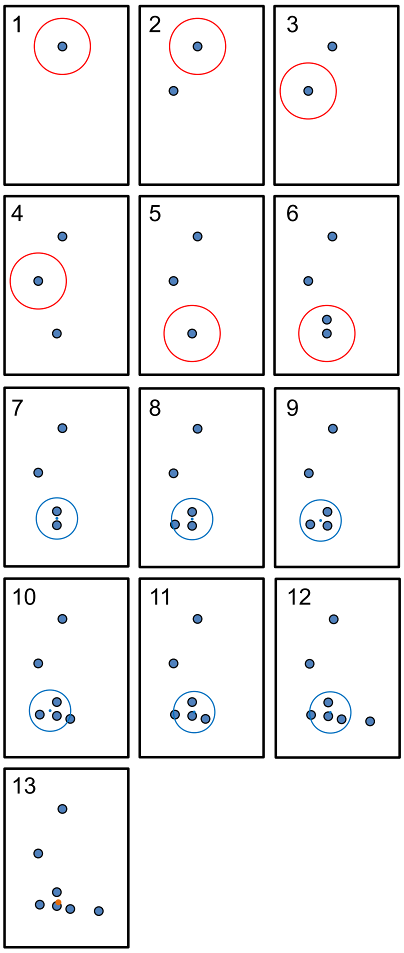


Figure 6.4: An illustration the algorithm for finding the trips made by an individual. The dots (or rather: smaller, filled circles) represent a location of a cell the subscriber has been connected to before the time the box represents. In the first box, the subscriber has only been at one cell. If the next data point is within the circle, the point will be classified as temporarily stationary. In this case, the next point is outside the circle, as can be seen in box 2, and the point is therefore classified as moving. The same procedure is then repeated in box 3 and 4. In box 6 however, the next point is within the circle. The point is therefore classified as stationary, and a mean point is created (box 7). As long as the subsequent points are within (the now smaller) radius, the points keep being classified as stationary. As soon as one point falls outside of the circle (box 12), the new point is classified as moving again. If the time between the first and the last stationary point (the point added in box 4 and the point added in box 10) is larger than a certain threshold, the mean point (weighted by the time spent in each point) is added as a station. When two stations have been added without any longer time between data points (e.g. the phone has been off for two days) in between the two stations, a trip will be created. This figure also explains the three parameters of the algorithm, the two radiuses and the time threshold.

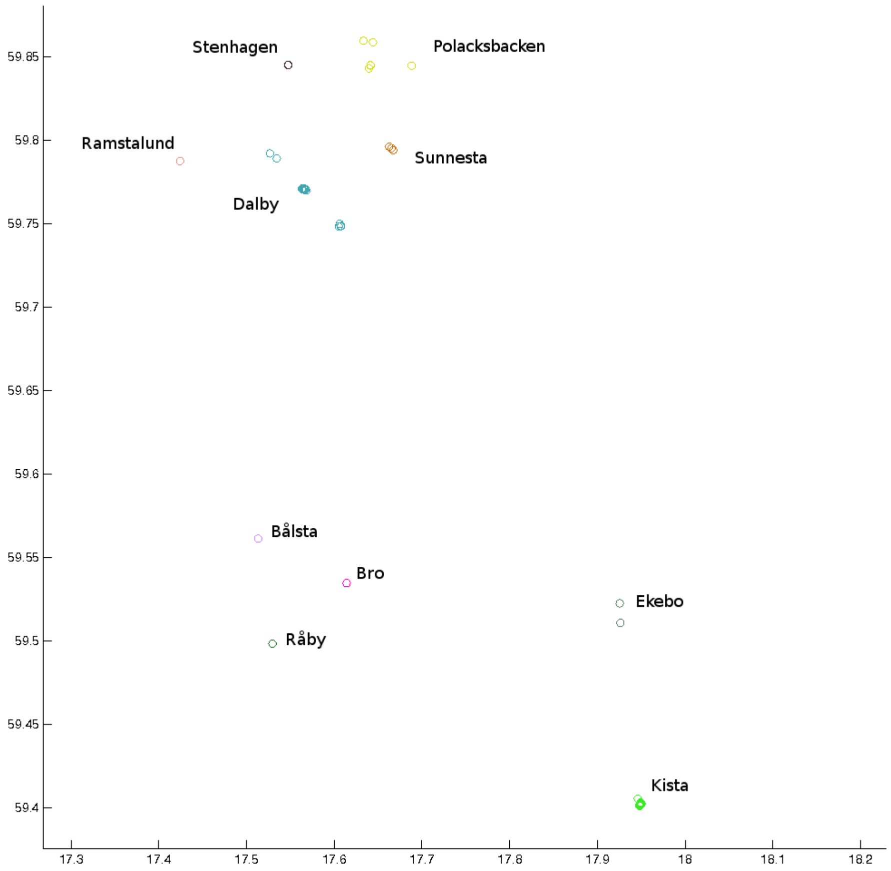


Figure 6.5: All stations for one user of the Android application and how they were clustered.

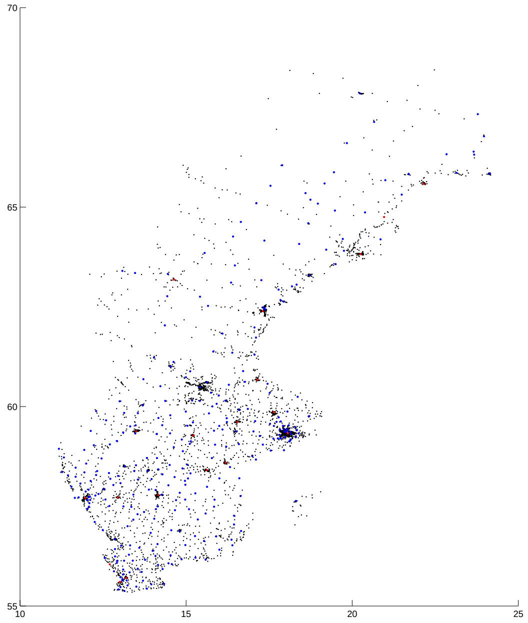
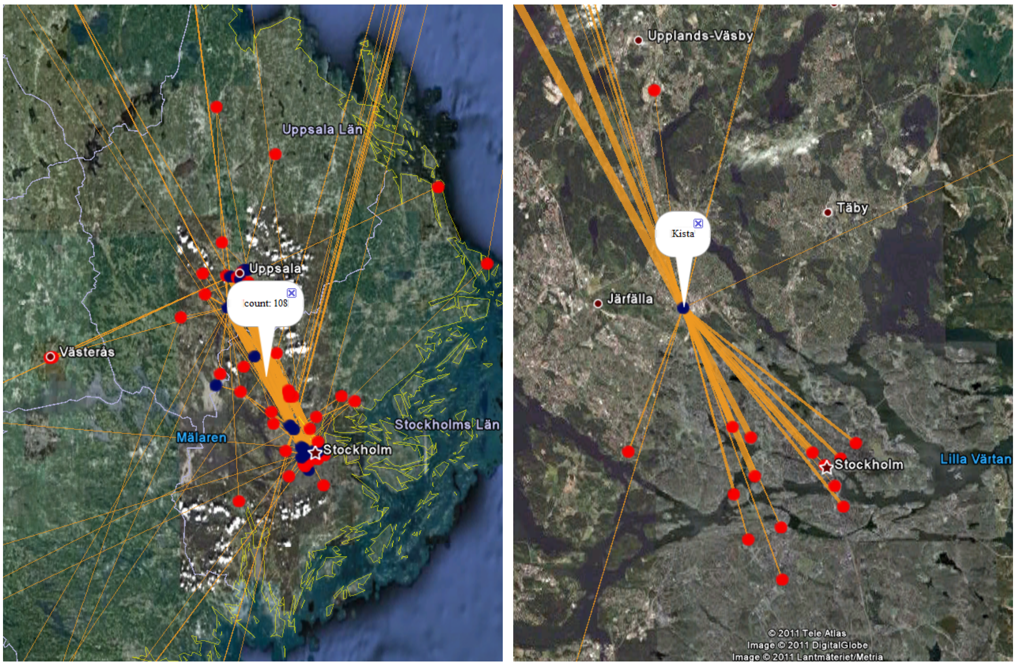


Figure 6.6: All the ”places” in Sweden, extracted from Open Street Map [6].



(a) **(b)**

Figure 6.7: The O/D-matrices displayed in a web browser using Google Earth. The red and blue circles represents origins and destinations and the lines represents the trips from one place to another. The more trips that were made, the thicker the line. Clicking on a place shows the name of the place and clicking on a line shows the total number of trips between the two places.

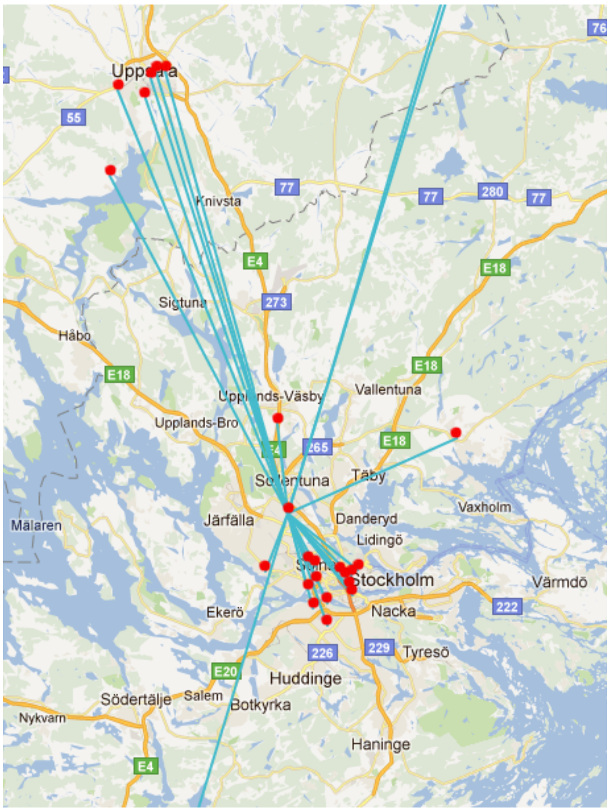


Figure 6.8: The O/D-matrices displayed in a web browser using Google Maps.

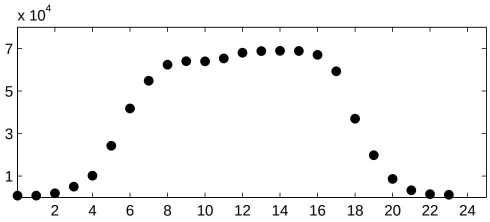


Figure 6.9: The distribution of trips over the day from the real data set.

7

Discussion

This chapter will discuss the advantages and disadvantages of the method developed and the results from running this method on the data available to Ericsson now.

7.1 Method

7.1.1 Advantages

a person is likely to visit the same place more than once

The methodology proposed has a lot of advantages. Comparing all stations for a single person makes it possible to get a better approximation of the location the user is actually at. The reason this is possible is that a person is likely to visit the same place more than once. Comparing the stations a person has visited and clustering them therefore makes it possible to get a better prediction of where the person actually was. Another advantage of treating each person individually is that it makes it possible to parallelize the problem using for example Hadoop and thereby making it a lot faster.

Privacy

k-anonymity- HOW TO ADDRESS!

When creating the O/D-matrices, all information that could be used to identify an individual is lost. This is however not enough to guarantee that an individual cannot be identified [18]. Another way of assuring anonymity is k-anonymity, which means that every individual should be indistinguishable from k-1 other, where k is a predefined number [18]. From the O/D-matrices obtained by the method described in this thesis, it is impossible to tell what day someone traveled from one place to another, you can only tell which month the trip took place. If you know at what time a person left a place and there are less than k-1 other people leaving that place during that month within the same time interval, then the method would not be k-anonymized. K-anonymity could be enforced either by removing information from the O/D-matrices that does not satisfy k-anonymity, or the matrices could be modified to enforce k-anonymity. Less than k trips from a place within a certain time frame would typically mean that the place is small village and the time interval is in the middle of the night. K-anonymity could then be enforced by either clustering villages together or by making the time intervals longer when fewer people travel.

7.1.2 Disadvantages

The biggest disadvantage of the method has to do with the definition of a trip. In the method developed in this thesis, a trip is defined as a travel between two places where the person spend at least 10 minutes. But taking the subway to the train station, just to leave the city by train 15 minutes later would in that case be split up into two trips. This would not be considered two separate trips in reality. But taking the car to the supermarket and then going back 15 minutes later would be consider two trips (both in reality and by the method). This is a difficult problem, and it cannot be solved by simple tweaking the parameters of the method. What you could do is to make this time limit vary depending on how much time you spent traveling before and after as well as the place where the person were stationary.

Another, somewhat related problem, is that some trips will not be identified because of the sparsity of the data. If a subscriber does not use his or her phone while shopping at the supermarket, chances are that there will not be any data collected in the mobile network of his/her presence there. This is not really a disadvantage of the method, but rather the data itself. It could however be dealt with, for example by analyzing data from the same user over a long time. That way, you might be able to draw the conclusions like: if he/she left his/her home and passed that certain place just to return home half an hour later, he/she probably visited this supermarket, since that has happened three times before (when there were more data available).

longer time -> paterns in user activity

7.1.3 Improvements

The values of the parameters in the result, the spatial parameter at 1 km and the time parameter at 10 minutes, should optimally not be fixed, but should vary depending on where the user is. They should be smaller in densely populated areas, where the distance between base stations are shorter. These values are an early result and more work is needed to find the optimal values of these parameters.

One could also consider adding more parameters to the algorithm for finding trips. You could for example add a parameter specifying the minimum travel time or a minimum travel distance. This way, two stations would be considered the same if they are not sufficiently separated in time or space. Tries were made using these parameters, but they were deemed unnecessary for the classification error of the trips.

7.2 Result

Running the algorithm on the call data records from two million users has shown that it possible to get O/D-matrices from real cellular network data. This result was from using call data records, containing only location information for a user at the beginning of each phone call. The algorithm is constructed for using on much more fine-grained data, containing data throughout the phone call as well as passive data collected when the phone is in idle mode. With this data, the method would hopefully be able to perform better.

Unfortunately, no real O/D-matrices obtained in some other way has been used to verify and compare the results. However, when studying the data and the movement distribution over the days, the results seems fair.

The more often a person uses his/her phone, the more accurate O/D-estimation it is possible to make. Smart phones generate much more data, since they are more often connected to the network. This makes it easier to make O/D-estimations. The usage of smart phones is only increasing, which will make this method even better in the future.

predictred future in 2011

8

Conclusion and future work

There are a few companies and organizations working on obtaining patterns of human mobility from mobil network data, and it will be interesting to see what company comes out ahead. There are also a lot of location-based services that are growing fast, for example four-square, facebook places, Google places etc. These are not really competitors to the method proposed in this thesis, since they do not collect information detailed enough for this kind of analysis. However, the data collected by Apple and Google latitude might have the resolution needed to do this kind of analysis. Whether or not they are working on projects similar to Consider8 is not known.

In the case of Google latitude, people are giving up the information voluntarily, Apple however saved information on a persons location in an iPhone without the users consent. Even though they never sent this information to be collected somewhere, this started some commotion [11], and they were blamed of tracking people. It is therefore apparent that it will be of great importance that it is impossible to track a single individual from the O/D-matrices obtained by the method developed in this thesis work. It is also likely that the personal privacy of the first attempt of releasing patterns of human mobility will determine the future of this kind of location based information.

This thesis has shown that it is feasible to obtain O/D-matrices from mobile network data while keeping the privacy of the subscribers. The most important step that remains before this method could be used in a commercial application is to convince the mobile operators that they can run this method without risking to lose the subscribers trust, so that the method can be tested on real data. The ultimate goal here would be to sell a program containing this method to the operators, so that they themselves can run the method and release the O/D-matrices. That way, the sensitive data does not have to leave the operators networks.

Bibliography

[1] Ericsson labs. https://labs.ericsson.com.

[2] Geographic privacy-aware knowledge discovery and delivery. http://www.geopkdd.eu.

[3] google geocoding. http://code.google.com/apis/maps/documentation/geocoding/.

[4] International workshop on spatial and spatiotemporal data mining.

[5] location-api. http://www.location-api.com.

[6] Open street map.

[7] Opencellid. http://opencellid.org.

[8] Your movements speak for themselves: Space-time travel data is analytic super-food! http://jeffjonas.typepad.com/jeff_jonas/2009/08/your-movements-speak-for-themselves-spacetime-travel-data-is-analytic-superfood.html, August 2009.

[9] 50 percent railway crossings are unmanned. http://facenfacts.com/NewsDetails/12307/50-percent-railway-crossings-are-unmanned.htm, July 2011.

[10] Airsage. http://www.airsage.com, 2011.

[11] Apple under pressure over iphone location tracking. http://www.telegraph.co.uk/technology/apple/8466357/Apple-under-pressure-over-iPhone-location-tracking.html, April 2011.

[12] Traffic transformation part deux gps probes vs. handoff data. http://blogs.strategyanalytics.com/auto/?p=109, May 2011.

[13] Torgil Abrahamsson. Estimation of origin-destination matrices using traffic counts - a literature survey, 1998.

[14] Rein Ahas, Anto Aasa, Siiri Silm, and Margus Tiru. Daily rhythms of suburban commuters movements in the tallinn, 2009.

[15] Andreas Allstrˆm and David GrundlegÂrd. Stockholm mobile millennium - modeller fˆr realtidsestimering av restider. s.l., January 2011.

[16] Yasuo Asakura and Hato Eiji. Tracking survey for individual travel behaviour using, 2003.

[17] N. Caceres, J.P. Wideberg, and F.G. Benitez. Deriving origin-destination data from mobile phone network, 2007.

[18] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-anonymity, 2007.

[19] Emily Ford. Salisbury city council debates unprotected railroad crossings. `http://www.salisburypost.com/News/031711-Salisbury-City-Council-debates-unprotected-railroad-crossings-qcd`, March 2011.

[20] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations, 2005.

[21] G. Derpanis Konstantinos. Mean shift clustering, August 2005.

[22] Keemin Sohn and Daehyun Kim. Dynamic origin-destination flow estimation using cellular communication system.

[23] Charles Solomon, Gur J Yehuda, Bekhor Shlomo, and Kheifits Leonid. Intercity person trip table for nationwide transportation planning in israel, 2009.

A

Mean shift clustering

The mean shift clustering algorithm does not need any prior knowledge of the number of clusters the points belong to, as apposed to many other clustering algorithms. This appendix will only explain how the algorithm works for the specific purpose it is used for in this thesis. For a general explanation of the algorithm, see [21].

The aim of the algorithm is cluster n points in a two dimensional space into an unknown number of cluster. We define a density:

$$f(x) = c \sum_{i=1}^{n} k(\mathbf{x} - \mathbf{x_i}),$$

where c is a constant and

$$k(x) = e^{-ax^2},$$

where a is another constant. The function $f(x)$ has a number of local maxima, where the density is (locally) higher. In such a point,

$$0 = \nabla f(x) = \sum_{i=1}^{n} -a(\mathbf{x} - \mathbf{x_i})\mathbf{e^{a(x-x_i)^2}} = \mathbf{a}\left[\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{e^{-a(x-x_i)^2}}\right]\left[\frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i e^{-a(x-x_i)^2}}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{e^{-a(x-x_i)^2}}} - \mathbf{x}\right].$$

The first term is a scalar, therefore we only have to consider the second term when deciding the direction of maximum increase. Define

$$\mathbf{m(x)} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i e^{-a(x-x_i)^2}}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{e^{-a(x-x_i)^2}}} - \mathbf{x}$$

and

$$\mathbf{x^{t+1}} = \mathbf{x^t} + \mathbf{m(x^t)}. \tag{A.1}$$

The sequence in A.1 will converge. All the points that converge to the same point are considered to belong to a cluster.

B

Hadoop MapReduce

Hadoop MapReduce is a framework developed by Google in 2004. It was developed for being able to do calculations on large amounts of data. It does the calculation in two steps, called ”map” and ”reduce”, both of them using key-value pairs as input and output.

Figure B.1 describes how MapReduces was used to find O/D-matrices. As input, the user id is used as key, and a list of all data points (longitude, latitude and timestamp) are used as value. The MapReduce framework then takes care of sending all data with the same key to one computer, and then start the ”mapper” function, where the main calculations happen. Output from the mapper is the O/D-matrices for that specific user in the form of key-value pairs, where the key is the position in the O/D-matrix (the id of the origin, the id of the destination and the time of departure) and the value is the number of trips that user made with that key. Again, MapReduce makes sure all key-value pairs having the same key is copied to the same computer and then starts the ”reducer” function. In this case, the reducer function is very simple, the output key being the same as the input key, and the output value the sum of all input values. The output is therefore the actual O/D-matrices we were trying to obtain.

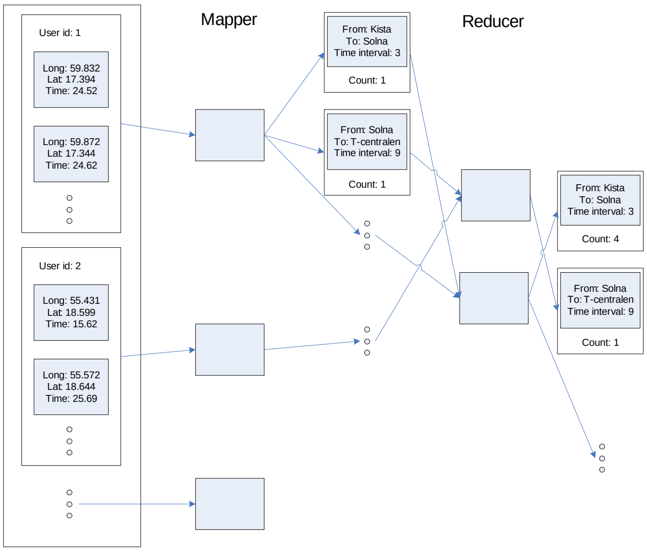


Figure B.1: An illustration of how hadoop operates on key-value pairs.

C

Data collection

The accuracy with which one can determine where a person is depends a lot on what kind of data is available, specifically the frequency and accuracy of the positioning for the mobile phones. The accuracy with which one can pinpoint a certain phone depends on where in the mobile network the information is gathered. Figure C.1 displays a simplification of the mobile network as it is in Europe.

A number of Base Transceiver Stations (BTS) are connected to a Base Station Controller (BSC). A number of BSC is in turn connected to a Mobile Switching Center (MSC). To be able to reach a phone when someone calls it, the network needs to know where the phone is situated. Therefore, all mobile devices are required to give there position to the mobile network about once an hour (or somewhere between 20 minutes and two hours depending on the mobile operator).

During the call, or when using some other kind of service, for example surfing the web, the MSC gets an update on which BTS the phone is connected to twice a second. This information is then stored for billing and network purposes.

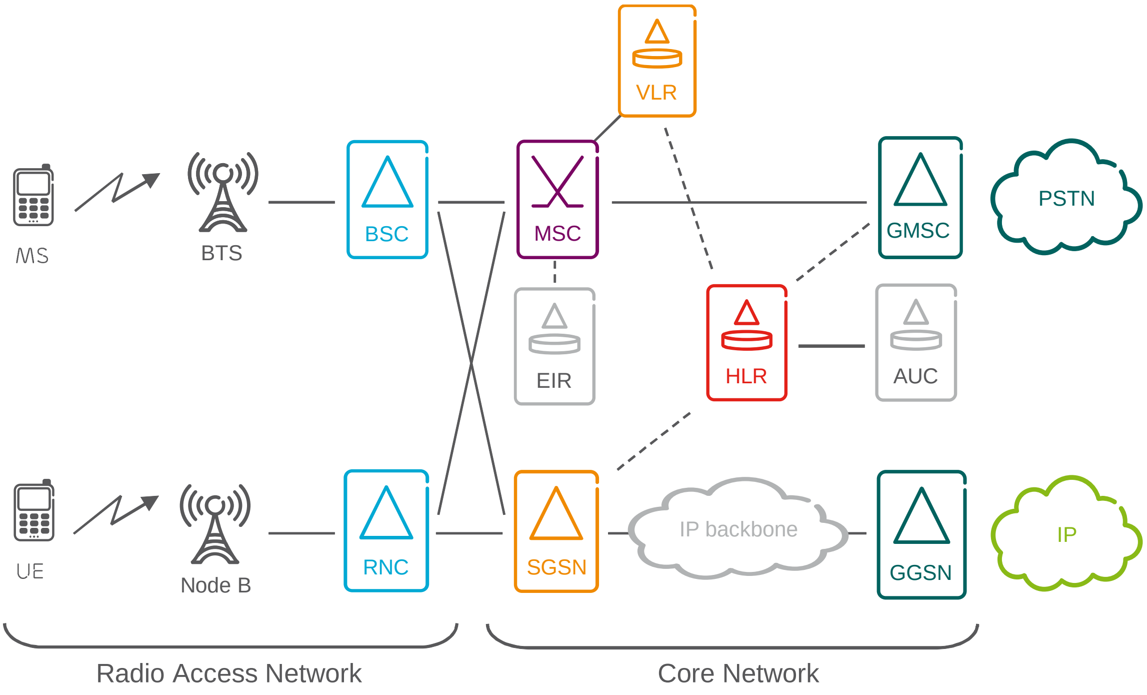


Figure C.1: An illustration of where the data is collected in the mobile network. The upper part of the figure shows the GSM (or 2G, second generation) network, while the lower part shows the corresponding functions for the 3G network.