GRS Master Thesis

Google Timeline Geolocation Accuracy

Andrea Macarulla Rodríguez

TU Delft

GRS Master Thesis

Google Timeline Geolocation Accuracy

by

Andrea Macarulla Rodríguez

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday June 1, 2017 at 1:00 PM.

Student number:	4421949	
Project duration:	March 7, 2016 – June 1, 2017	
Thesis committee:	Prof. dr. ir. Ramon Hanssen,	TU Delft, Chair
	Dr. ir. Christian Tiberius,	TU Delft, supervisor
	Drs. Roel van Bree,	Nederlands Forensisch Instituut, supervisor
	Dr. ir. Gerard Janssen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

Google Location History Timeline could be used in the future to track mobile devices of users with a Google account. The Department of Forensic Digital Technology in the Netherlands Forensic Institute might consider using it as available data for evidence in its investigations.

A part of this research is to assess the accuracy of the locations given by Google Location History Timeline. Google informs that any registered mobile device was at a certain time at a certain position, and provides a measure of the accuracy.

To study the veracity of the information provided by Google, a series of experiments were carried out. During these experiments the true position was recorded with a reference GPS device with a superior order of accuracy.
Subsequently, the accuracy values given by Google were studied and analyzed based on various parameters, such as the configuration of mobile device connectivity, speed of movement, environment, traffic density and weather.
The distance between Google provided position and actual position (determined with a more precise device) is computed and called *Google Error*. Then this error was compared with the Google provided accuracy to have a measure of Google data quality.

Additionally, linear least squares multivariate models were developed with the purpose of calculating the precision that Google would provide a priori together with its positioning error.

When studying the variability of the Google accuracy and Google error, in the experiments it was found that these variables are dependent on the configuration of the mobile device, the environment, and the means of transport, but weather and traffic have no influence on these variables.

To quantify the performance of the values provided by Google, a Hit is defined as the observation in which the actual error committed by Google is less than the accuracy provided. The configuration that has the largest Hit rate is the GPS connection, with a 52% success. Then 3G and 2G go with 38% and 33% respectively. The WiFi connection only has a Hit rate of 7%. Regarding the means of transport, when the connection is 2G or 3G, the worst results are in Still with a Hit rate of 9% and the best in Car with 57%.

For predicting values for Google accuracy and Google error, six multivariate linear models were defined.
The model input variables were the distances and angles from the position of the device to the three nearest cell towers, and the categorical (non-numerical) variables of *Environment* and *means of transport*.
The signal strength received by the device from the base stations were treated as possible input variables too, but not sufficient correlation was obtained, on top these models would not be useful to study future forensic cases, since these measures are not usually available.
To evaluate the utility of a model, a *Model Hit* is defined when the actual observation is within the 95% confidence interval provided by the model.
The model that shows the best results was the one that predicted the accuracy when the used network is 2G, with 76% of *Model hits*. The next one had only a 23% success (accuracy 3G).

As a conclusion from the performed experiments the assurance of Google providing the correct position can not be given. The accuracy radius Google provides when using exclusively telephony networks (2G or 3G) is overbounding the actual position error only in about 35% of the experiments; in the other 65% of the experiments, the actual error is larger than the given accuracy radius. For an accuracy measure to be of practical meaning, the confidence level should be much larger, for instance 95%.
Even when using WiFi and GPS, Google gives accurate locations but the accompanying accuracy measure is too optimistic (small radii) and hit rates are very low.

The linear models developed in this thesis gave results which were not satisfactory enough yet. Further research in the parameters involved and a major collection of data is required.

Acknowledgments

After an intensive period of one year, I am glad to say this note of thanks is the finishing touch on my thesis. It has been a cycle of intense learning for me, not only in the scientific arena, but also on a personal level.

I would like to reflect on the people who have supported and helped me so much throughout these years.

I would first like to thank my supervisors dr.ir. Christian Tiberius from TU Delft and drs. Roel van Bree from Netherlands Forensic Institute for their excellent guidance and support during this process.

In addition I would like to thank dr.ir. Gerard Janssen from TU Delft for his valuable guidance.

Prof. dr. ing. Zeno Geradts deserves my infinite gratitude for his patience and for offering me the opportunity to continue as PhD in the NFI.

To my colleagues at NFI: I would like to thank you for your wonderful cooperation as well. It was always helpful to discuss ideas about my research around with you.

I would like to thank my classmates and friends for their support during all these years of university, both in Madrid and Delft.

I would finally like to mention my parents, without their encouragement and values this work would not have been possible.

Thank you very much, everyone!

Andrea Macarulla Rodríguez
Delft, June 2017

Preface

What you are about to read is the Master Thesis "Google Timeline Geolocation Accuracy", the basis of which is a survey on the location data that Google provides for registered mobile devices. It has been written to fulfill the graduation requirements of the Geoscience & Remote Sensing Master Program at the Delft University of Technology (TU Delft). I was engaged in researching and writing this Thesis from March 2016 to April 2017.

The project was undertaken at the request of the Netherlands Forensic Institute, where I undertook an internship.

My research question was formulated together with both my supervisors dr.ir. Christian Tiberius (TU Delft) and drs. Roel van Bree (NFI). The research was arduous, but conducting extensive investigation has allowed me to answer the question that we identified.

The goal of this thesis is to study the behavior of the Google application that provides the location of a mobile device. This information is given in the form of a position and a radius defining a circular area where the device is supposed to be.

To study this behavior experiments were performed with two mobile phones with the Google Timeline application active and a separate GPS device.

Comparative results were analyzed, and multivariate linear models of least squares were developed to try to predict the behavior of Google.

Fortunately, drs. Roel van Bree from NFI and dr.ir. Christian Tiberius and dr.ir. Gerard Janssen from TU Delft, were always available and willing to answer my queries and guide me in my work.

I hope you enjoy your reading.

Andrea Macarulla Rodríguez
Delft, June 2017

Contents

1 Introduction

1.1. Google Timeline Geolocation

Trying to remember were you've been? Google can help. As explained in [32], if you opt into being tracked, Google can record where you've been through Google Maps and your Android phone. Everything is logged in an interactive map called your Timeline that's accessible through your Google account.

To access your Timeline, you have to turn on Location History. It can be enabled or disabled in your Google Settings on phones running Android 2.3 or higher. When you first set up your Android phone, Google will likely ask you to turn Location History on (it's not turned on by default).

Google uses your location, search, and browsing info to make your timeline (source [33]).

To make your timeline, follow the steps below.

1. Open the Google Maps app Google Maps .
2. Tap Menu and then Your timeline .
3. Tap More and then Timeline settings.
4. Under "Location settings," make sure your location and Location History are on. This lets you turn on your Location History so that you can track the routes that you've traveled in your Google Account.

Google tracks your location through Google Maps (see figure 1.1), which also works on the iPhone and the web. You can see your Timeline from your settings in the Google Maps app on Android. It even shows if you walked, drove, or were in a plane.

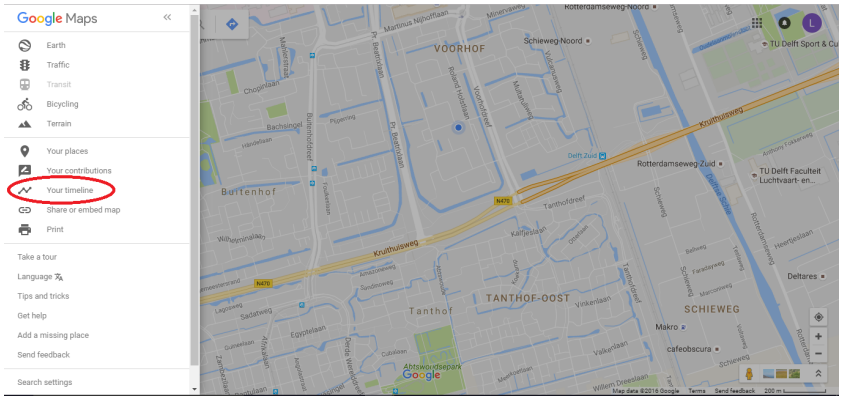


Figure 1.1: Accessing Google Timeline. To see your Timeline from the web, go to Google.com/maps/timeline.

1.2. Netherlands Forensic Institute

According to [2], The Netherlands Forensic Institute (NFI) is one of the world's leading forensic laboratories. Its aim to encourage fact-finding by means of independent forensic investigation. Its mission is to facilitate effective law enforcement and administration of justice from a focus on scientific information positions.

It is located in the Ypenburg quarter of The Hague and it is an autonomous division of the Dutch Ministry of Security and Justice and falls under the Directorate-General for the Administration of Justice and Law Enforcement.

The core duties of the NFI are:

- Forensic investigation in criminal cases
- Research & Development
- Knowledge Lab

In this context, the NFI works for the Public Prosecution Service, the judiciary, the police, and the Special Investigation Services.

Besides these core duties, the NFI has several additional duties. These additional duties include activities such as giving courses to fire brigade personnel and ambulance staff, who – just like clients of core duties such as the police – must often enter the crime scene in their official capacity. Another example is training lawyers to understand NFI reports.

In the interest of law enforcement at the national and international level, the NFI may also be asked to provide services to Dutch and foreign governmental and intergovernmental organizations. Among these duties are, for example, commissions for the Immigration and Naturalisation Service and image and audio analysis for UN Tribunals.

Figure 1.2: Image of the Netherlands Forensic Institute

1.3. Motivation and purpose of the thesis

The Department of Forensic Digital Technology in the NFI ponders the possibility of using Google Location History Timeline in the future as assistance in their investigations. It can help to open possibilities to track mobile devices of suspects, gathering information about their where-abouts.
In order to be able to harness this information in court or for justice purposes, an assessment of the accuracy has to be performed. Google registers that the mobile device of the suspect was at a certain time in a certain position, and it estimates its own error of X meters. Can we estimate this error? Can we affirm that the device was there based solely in Google information? An assessment of the validity of the data must be taken.

For that purpose, the following research questions are formulated in the next section.

1.4. Research questions

To help the NFI with their inquiry, this thesis is posed based on two research questions.

1.4.1. What is the actual accuracy of the location data that Google Location History provides?

As explained in the motivation, Google caters to the user the information of a position (time, longitude and latitude) but not with a specification nor explanation of the methods used. This sets the ground to pose several questions.

How do we quantify Google Geolocation accuracy?

According to he International Organization for Standardization (ISO) accuracy is defined as the closeness of agreement between a test result and the accepted reference value [6]. Inspired by [70] and [63], the approach to assess the accuracy is to compare the position Google grants with another position provided by a GPS device that will be considered as Ground Truth . This device will have an error on its own, but since the order of magnitude of the Google error will always be greater (except when the phone GPS is activated) the error of the device will be neglected because the order of magnitude is better in accuracy than phone GPS. To see the idea behind this quantification, see figure 1.3. In the figure, the term accuracy is short for "Google provided accuracy" and Error is the assessment of the error Google is truly making with respect to the GPS ground truth position.

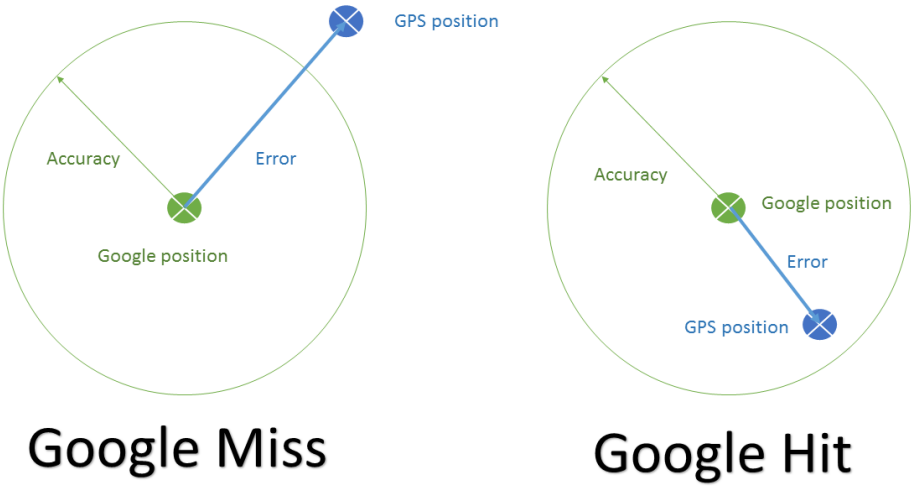


Figure 1.3: Description of what is considered a Google hit/miss. According to Google is inside a circle whose center is the presented position and whose radius is the accuracy studied in this thesis. Measuring the distance to the *Ground Truth* point (location provided by GPS device) we determine if the device was truly inside the circle. If it is, we call it a *hit*, otherwise it is a *miss*.

Does accuracy stated by Google correspond to actual accuracy?

We want to observe how often Google is able to provide a right value of accuracy. For that, we will compare it to the actual positioning error in each experiment. Also, we will study the dispersion or variability that Google provides for its own accuracy when conditions are nominal.

Is there the possibility of doing reverse engineering to determine how Google computes the accuracy?

Google location computation algorithms are unknown to the user. However, we can try to figure out how a set of parameters (e.g. kind of environment, multipath, type of signal) affect the accuracy of the location points. To discern which parameters are significant to those which are not, different experiments with different conditions (weather, traffic, network configuration) will be performed and evaluated.

Where does Google take the information from?

Studying [28] and [64], the three possible sources where Google gets data are most likely Cell Towers, WiFi and GNSS. Understanding how and when each of them is used to get location will elucidate why Google presents sometimes really wide circles and other time narrower ones. It is also of interest to discover if Google uses more than one source at the same time and with which criteria.

How and when does Google store/compute the locations and send them to the server?

Google probably saves and updates more frequently when the smart-phone is moving than when it is still. Using a command-line tool called Android Debug Bridge (ADB) it should be possible to determine its velocity or precision influences the frequency of data acquisition and update [1]. It will also help us to understand if the information provided to Google comes from their own server or the smartphone itself.

1.4.2. Is it possible to perform a prediction of the accuracy radius and error that Google will provide in case there is new experiment incorporated?

If a suspect's smartphone is handed on as evidence containing new data on it, we don't have a *ground truth* anymore to compare. So based on the previous data collected from experiments we aim to provide a confidence interval where the phone could have been.

What information can be extracted from the phone?

It has to be checked if the phone, with nothing installed previously does save parameters such as signal strength, connected WiFi or base stations to which it has been connected to. GPS and application registers in the logcat would also be useful.

With this information, can an algorithm (or several ones) be used to deduce its previous locations?

Instead of an algorithm, in this thesis we will apply a method called multi linear regression [48]. This model is used to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. To apply this method, it is important to determine which parameters affect Google's accuracy and error (subquestion 1.4.1).

Once the model has been developed, is it good enough to be considered accurate?

A separate set of experiments will be set to verify the validity of the model. These experiments will be performed in the exact same way as the experiments which the model is based on, but we won't introduce them as input. Instead, we will evaluate them as if they were "new data" and then compare the results that the model presents versus the data we have registered. With that, we will verify if the predictions are good in the case new data has to be analyzed in the future.

1.5. Thesis overview

During this thesis an evaluation of Google Location Timeline will be performed. It is divided into 8 chapters and 2 annexes. This introduction is the first chapter.

In second chapter of the thesis, a literature review is evaluated. Some of the methods that comprise modern Mobile Location systems are described.

In the third chapter the theoretical principles of Multi-linear models are explained, because these models are used in the calculus with the experimental data.

In the fourth chapter, the field experiments executed during this research are described.

In the fifth chapter, methodology is written. In it, we explain the way the data is handled, according to the mathematics explained in chapter 3. With them the linear model that will help us to find if accuracy and error prediction are possible is developed.

In the sixth chapter, the results of the experiment measurements and analysis of accuracy and error of Google Timeline are exposed and discussed.

In seventh chapter, the results of applying the linear model in the experiments and analysis of said experiments are exposed and discussed.

In eighth chapter, conclusions about the research are drawn.

In Annex A, the Matlab programs to collect data and algorithms are explained.

In Annex B is the instruction manual for the developed software for future researchers.

2

Literature Study

In this chapter, we will study the wireless location technologies that are used nowadays to locate a device. Three sources will be studied: Cell Tower (2G/3G/4G), WiFi and GNSS [10]. The methods to study are: Basic positioning methods (Dead reckoning, Proximity sensing, trilateration, multilaterarion and triangulation), then Satellite positioning systems (GPS and Assisted GPS), Received signal strength, fingerprinting and IP location.

2.1. Introduction: Mobile Location Network

A Mobile Location Network uses a signal from mobile provider. The technology of locating is based on measuring power levels and antenna patterns and uses the concept that a powered mobile phone always communicates wirelessly with one of the closest base stations, so knowledge of the location of the base station implies the cell phone is nearby [8] [25].

Advanced systems determine the sector in which the mobile phone is located and roughly estimate also the distance to the base station. Further determination can be done by interpolating signals between adjacent antenna towers [64].

2.2. Basic positioning methods

2.2.1. Dead reckoning

Dead reckoning or dead-reckoning (also ded for deduced reckoning or DR) is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course [64].

Dead reckoning begins with a known position, or fixed, which is then advanced, mathematically or directly on the chart, by means of recorded heading, speed, and time. Speed can be determined by many methods. Before modern instrumentation, it was determined aboard ship using a chip log. More modern methods include pit log referencing engine speed (e.g. in rpm) against a table of total displacement (for ships) or referencing one's indicated airspeed fed by the pressure from a pitot tube. Distance is determined by multiplying the speed and the time. This initial position can then be adjusted resulting in an estimated position by taking into account the current (known as set and drift in marine navigation). If there is no positional information available, a new dead reckoning plot may start from an estimated position. In this case subsequent dead reckoning positions will have taken into account estimated set and drift [36]. For an illustration of this, see image 2.1.

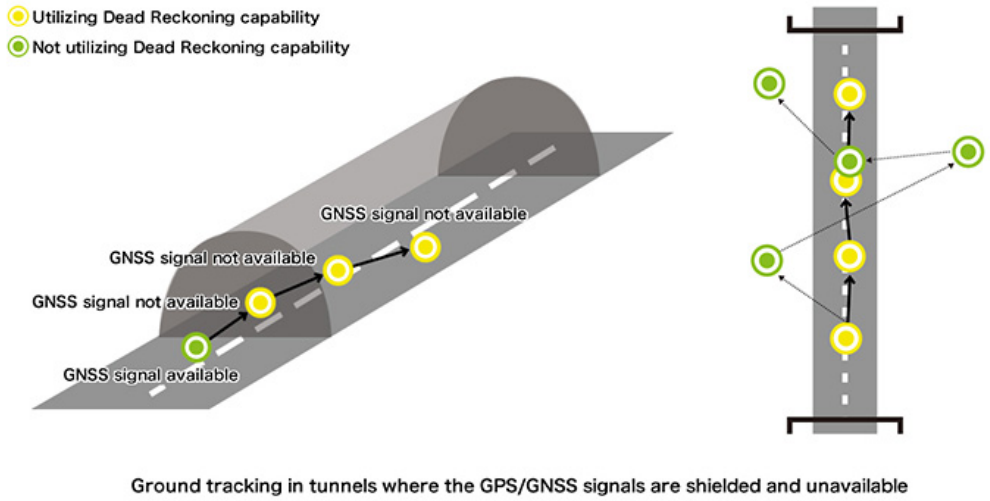


Figure 2.1: Dead reckoning illustration. It enables to keep high accuracy positioning by using information from various sensors (gyro sensor, accelerometer, speed pulse, etc.) to calculate the current position, even when GPS/GNSS only positioning is difficult or impossible. Image extracted from [4]

The equation for computing the new position is (with constant acceleration) [23]:

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 \Delta t + \frac{1}{2}\mathbf{a}\Delta t^2 \tag{2.1}$$

where

x(t) is the position vector of the object at any time t

$\mathbf{x}_0$ is the position vector of the object at the initial time

$\mathbf{v}_0$ is the velocity of the object at the initial time

a is the acceleration of the object (constant vector)

Dead reckoning positions are calculated at predetermined intervals, and are maintained between fixes. The duration of the interval varies. Factors including one's speed made good and the nature of heading and other course changes, and the navigator's judgment determine when dead reckoning positions are calculated.

It is useful because this is the simplest way finding your approximate position, although it is the least accurate method [9].

2.2.2. Proximity Sensing: Signal Signature

The mobile position is derived from base-station coordinates. It is usually determined by tracking signal signatures or cell identity (Cell ID) of neighboring base stations [64] [35].

Every base station has its own signal pattern, which is usually embedded into its pilot and some synchronization channels. It normally comprises: signal signature estimation, neighbor list update and mobile location analysis.

On the other hand, if you are indoor, and using WiFi, this method may be used too. Some WiFi have location services capabilities. WiFi positioning takes advantage of the rapid growth in the early 21^{st} century of wireless access points in urban areas [41].

There are many advantages of the fingerprinting approach [68], including the fact that no special hardware is required on the user mobile station (MS) side. A big disadvantage is that trees or buildings may change the fingerprint that corresponds to each location, requiring an update to the fingerprint database.

The problem of WiFi based indoor localization of a device consists in determining the position of client devices with respect to access points.

A technique is used, called *fingerprinting*. It simply relies on a calibration survey which consists on the recording of the signal strength from several access points in range and storing this information in a database along with the known coordinates of the client device (as an offline phase).

There are many advantages of the fingerprinting approach [68], including the fact that no special hardware is required on the user mobile station (MS) side.
A big disadvantage is that trees or buildings may change the *fingerprint* that corresponds to each location, requiring an update to the fingerprint database.

Another disadvantage is that the calibration survey has to be done beforehand in the zone of the study.

What Google does is to use his 'War-Cars' as they are called and as well as systematically photographing streets and gathering 3D images of cities and towns around the world, Google's Street View cars are fitted with antennas that scan local WiFi networks and use the data for its location services [40]. This has been quite controversial but can explain the accuracies obtained when WiFi is activated but no connection is established.

One obvious approach is to convert the Signal Strength (SS) (See section 2.5) to distance measurements. If three distances between the user receiver and different Access Points (APs) can be obtained, trilateration can be used to estimate the receiver's position [17]. However creating an accurate model to convert SS to distance is difficult. The propagation of radio signals in indoor environments is very complicated. The SS received from an Access Point varies significantly (up to 15 dBm) over time at the same location. Such systems may provide a median accuracy of 0.6 m and tail accuracy of 1.3 m [1] [41] [68].

2.2.3. Trilateration

Trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles [28].

Normally the position of the device is determined using trilateration with Time of Arrival (TOA) [34]. The method is as follows if three cell tower positions are known, computing the time of signal arrival from each of them the distances to the three towers can be computed. In order to achieve it, the clocks of the towers and from the device have to be perfectly synchronized and the ranges from the device to the towers precisely measured. Usually, the solution is not always exact, so the minimum error is searched using the least squares solution. But if we have more equations then unknowns, a direct and low computational cost method can be used [39]. For a graphic illustration of this method, see figure 2.2

[1] Median of 0.6 m means that 50% of the experiments are below 0.6 m and tail of 1.3 m means that only %5 of the experiments are above 1.3 m

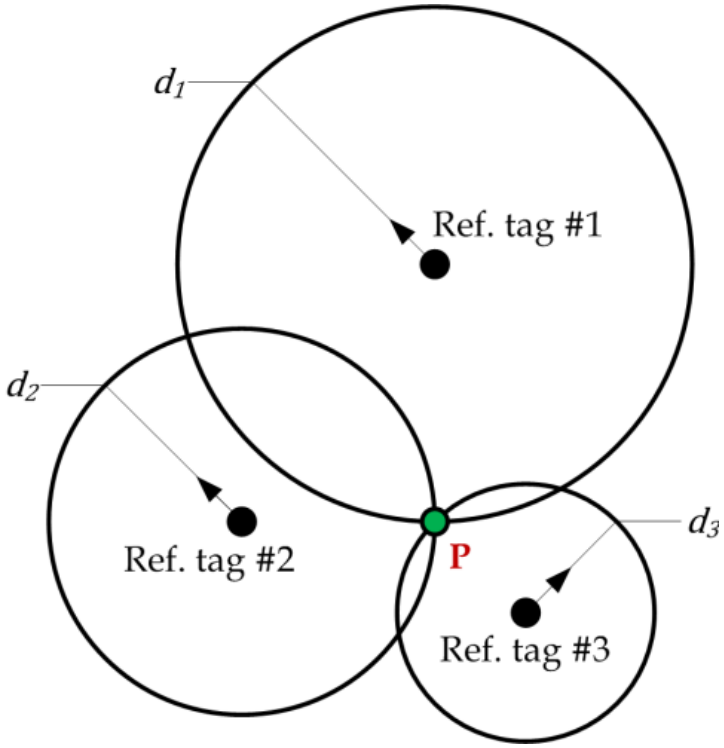


Figure 2.2: TOA Method representation. Extracted from [28]

Using the direct method, the distances are determined by the following equations:

$$d_i = c \cdot t_i$$

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} = c \cdot t_i \tag{2.2}$$

Where d_i is the radius of the circle, x_i and y_i are the coordinates of the Cell Towers device and (x, y) are the coordinates of the distance. The time the signal takes to go from the tower to the device is t_i, and c is the speed of light.

The equations to determine x, y and t are not linear, so the equations can be linearized around a point close to the solution. If more equations than unknowns are available, the equation system maybe inconsistent (it does not have an exact solution which satisfies all the equations). Applying Linear Least Squared Errors method (on the linearized equations) on an iterative way, the final solution can be achieved.

In this case there are simpler and easy to program in computers methods, called direct methods, which get a solution, with the cost of losing information. [47]

So, having into account that the unknown point P have (x, y) as coordinates, and the known Cell ID points are (x_i, y_i). Let's do the example of three Cell towers which radius are d_1, d_2 and d_3. See figure 2.2.

$$\begin{aligned}(x - x_1)^2 + (y - y_1)^2 &= d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 &= d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 &= d_3^2\end{aligned} \tag{2.3}$$

Then expanding into squares:

$$\begin{aligned}x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 &= d_1^2 \\ x^2 - 2x_2x + x_1^2 + y^2 - 2y_2y + y_2^2 &= d_2^2 \\ x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 &= d_3^2\end{aligned} \tag{2.4}$$

Subtracting second equation from the first in system (2.4) we obtain:

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$$

And from the same system (2.4) we subtract third from the second:

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = d_2^2 - d_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2$$

This is a system with two equations with two unknowns.

$$\begin{aligned} Ax + By &= C \\ Dx + Ey &= F \end{aligned} \tag{2.5}$$

Being:

$$\begin{aligned} A &= 2(x_2 - x_1) \\ B &= 2(y_2 - y_1) \\ C &= \frac{t_1^2 - t_2^2}{c^2} - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ D &= 2(x_3 - x_2) \\ E &= 2(y_3 - y_2) \\ F &= \frac{t_2^2 - t_3^2}{c^2} - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{aligned}$$

The position of the device can be then extracted and then the solution to system (2.6) [38][28]:

$$\begin{aligned} x &= \frac{BF - EC}{BD - EA} \\ y &= \frac{CD - FA}{BD - EA} \end{aligned} \tag{2.6}$$

2.2.4. Time Difference of Arrival (TDOA)

This method follows the same principle as TOA, but this time the measurement is difference in the arrival times between two stations. In this way the location to look is some point of a branch of a hyperbola. Repeating the process with a third tower, another hyperbola is obtained. The intersection of both branches gives the location of the point [28][50].

To see an illustration of the method, see figure 2.3.

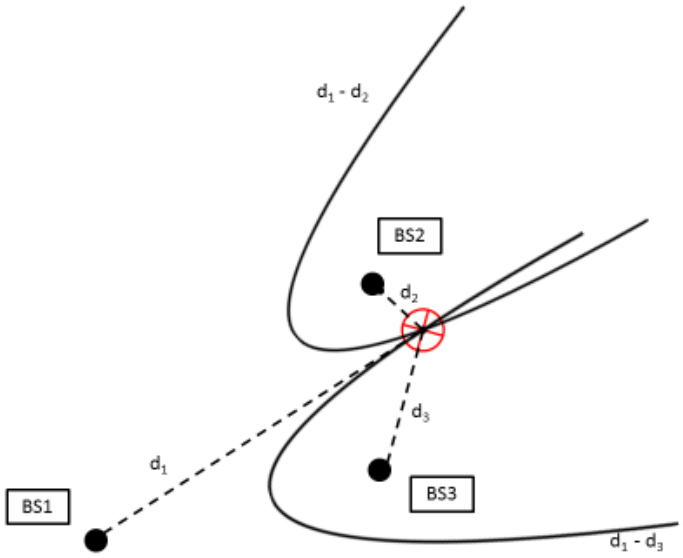


Figure 2.3: Time Difference of Arrival (TDOA). Figure based on one appearing in [28].

With this method, the receiver's clock does not have to be synchronized with the network time, because it's the difference in time that is measured.

The equations that determine the location are:

$$\begin{aligned} d_{21} &= d_2 - d_1 = c(t_2 - t_1) \\ d_{31} &= d_3 - d_1 = c(t_3 - t_1) \end{aligned} \tag{2.7}$$

For convenience, the coordinates of $BS1$ are taken as $(0,0)$. Then we have:

$$\begin{aligned} c(t_2 - t_1) = d_{21} &= \sqrt{(x_2 - x)^2 + (y_2 - y)^2} - \sqrt{x^2 + y^2} \\ c(t_3 - t_1) = d_{31} &= \sqrt{(x_3 - x)^2 + (y_3 - y)^2} - \sqrt{x^2 + y^2} \end{aligned} \tag{2.8}$$

Equation systems (2.7) and (2.8) are solved in x and y. After some mathematical manipulation (for whole development see reference [28]), the equation to which we arrive is the following:

$$\Theta = d_1 H^{-1} a + H^{-1} b \tag{2.9}$$

Where the fixed terms are

Device position $\Theta = \begin{bmatrix} x \\ y \end{bmatrix}$. This is the information we are looking form.

Cell Tower positions $H = \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}$

Linear term $a = \begin{bmatrix} -d_{21} \\ -d_{31} \end{bmatrix}$. These are the time differences converted into distances

Quadratic term $b = \frac{1}{2} \begin{bmatrix} x_2^2 + y_2^2 - d_{21}^2 \\ x_3^2 + y_3^2 - d_{31}^2 \end{bmatrix}$

And the distance to the first tower (BS1)

$$d_1 = c \cdot t_1 = \sqrt{x^2 + y^2} \tag{2.10}$$

The main challenge is to determine how to compute d_1, because the distance is not known. The method to compute it will be iterative until the solution reaches the real one. First we do an estimation of d_1 and we introduce it in equation (2.9) and we obtain a first solution of Θ. Using equation (2.10) we obtain a new value of d_1. We repeat the process until d_1 converges and then we have the solution.

2.2.5. Angle of Arrival (AOA)

This method is based on the measurement of the angle of arrival (AoA) of the signal. Two (or more) oriented bases with directional antennas are necessary. These antennas are capable of measuring the signal arrival angle from the device, and subsequently communicate the information to it. With a simple calculation, the device can determine its own position [28].

Multiple receivers on a base station would calculate the AoA of the cell phone's signal, and this information would be combined to determine the phone's location on the earth [31]. See figure 2.4

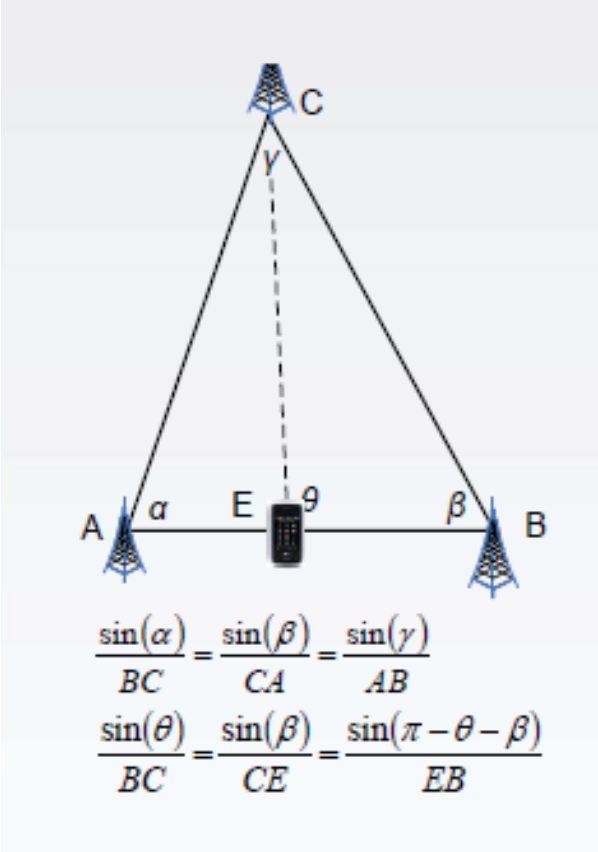


Figure 2.4: Angle of Arrival. The device is normally located by several antennas and a base station. Figure extracted from [64].

Generally this measurement is made by measuring the difference in received phase at each element in the antenna array. The delay of arrival at each element is measured directly and converted to an AoA measurement [26].

This can be thought of as antenna in reverse. In beamforming, the signal from each element is delayed by some weight to "steer" the gain of the antenna array.

Consider, for example, a two element array spaced apart by one-half the wavelength of an incoming RF wave. If a wave is incident upon the array at boresight, it will arrive at each antenna simultaneously. This will yield 0° phase-difference measured between the two antenna elements, equivalent to a 0° AoA. If a wave is incident upon the array at broadside, then a 180° phase difference will be measured between the elements, corresponding to a 90° AoA. For illustration see figure 2.5

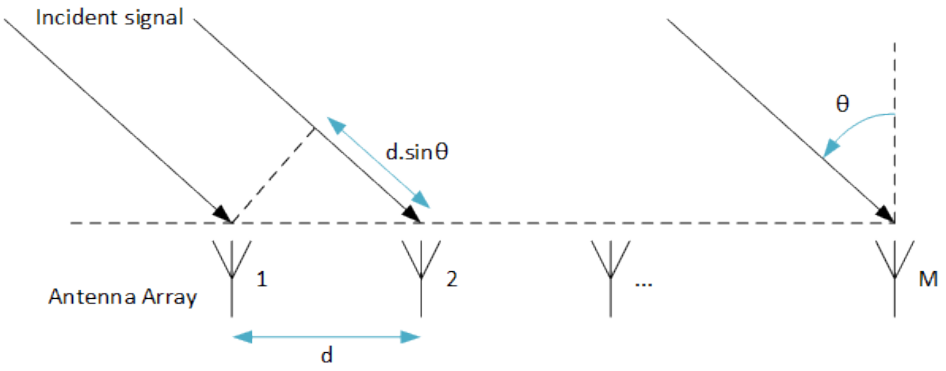


Figure 2.5: Angle of arrival with directional antenna. Figure extracted from [41]

2.3. Location by GPS

The Global Positioning System (GPS) is a space-based radionavigation system owned by the United States government and operated by the United States Air Force. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites [47].

The concept is based on time and the known position of specialized satellites. The satellites carry very stable atomic clocks that are synchronized with one another and to ground clocks. Any drift from true time observed in any satellite is corrected daily from the ground station. Likewise, the satellite locations are known with great precision. GPS receivers have clocks as well; however, they are usually not synchronized with true time, and are less stable. The difference of time between the device and the GPS network is taken as another unknown to add to the three spacial coordinates to calculate.

GPS satellites continuously transmit their current time and position. A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time).

2.3.1. System structure

GPS consists of three segments - the satellite constellation, ground control network, and user equipment. The satellite constellation comprises satellites in medium earth orbit that provide the ranging signals and navigation data messages to the user equipment. The ground control network tracks and maintains the satellite constellation by monitoring satellite health and signal integrity and maintaining the satellite orbital configuration. Furthermore, the ground control network also updates the satellite clock corrections and ephemerides as well as numerous other parameters essential to determining user position, velocity and time (PVT). The user equipment receives signals from the satellite constellation and computes user PVT [5]

Satellite Constellation

The baseline satellite constellation consists of 30 satellites positioned in six earth-centered orbital planes with four operation satellites and a spare satellite slot in each orbital plane. The system can support a constellation of up to thirty satellites in orbit. The orbital period of a GPS satellite is one-half of a sidereal day or 11 hours 58 minutes. The orbits are nearly circular and equally spaced about the equator at a 60-degree separation with an inclination of 55 degrees relative to the equator. The orbital radius (i.e. distance from the center of mass of the earth to the satellite) is approximately 26,600 km [47].

With the baseline satellite constellation, users with a clear view of the sky have a minimum of four satellites in view. It's more likely that a user would see six to eight satellites. The satellites broadcast ranging signals and navigation data allowing users to measure their pseudoranges[2] in order to estimate their position, velocity and time, in a passive, listen-only mode [5].

[2]The pseudorange (from pseudo- and range) is the pseudo distance between a satellite and a navigation satellite receiver (see GNSS positioning calculation) —for instance Global Positioning System (GPS) receivers.

Ground Control Network

At the heart of the Ground Control Network is the Master Control Station. The MCS operates the system and provides command and control functions for the satellite constellation.

The satellites in orbit are continuously tracked from six USAF (United States Air Force) monitor stations spread around the globe in longitude: Ascension Island , Diego Garcia, Kwajalein , Hawaii , Cape Canaveral and Colorado Springs. The monitor stations form the data collection component of the control network. A monitor station continuously makes pseudorange measurements to each satellite in view. There are two cesium clocks referenced to GPS system time in each monitor station. Pseudorange measurements made to each satellite in view by the monitor station receiver are used to update the master control station's precise estimate of each satellite's position in orbit [47] [5].

User Equipment

The user equipment, often referred to as "GPS receivers", captures and processes L-band signals from the satellites in view for the computation of user position, velocity and time [47].

2.3.2. Method

Each GPS satellite continually broadcasts a signal (carrier wave with modulation) that includes:

Pseudorandom code Sequence of ones and zeros. It is known to the receiver. By time-aligning a receiver-generated version and the receiver-measured version of the code, the time of arrival (TOA) of a defined point in the code sequence, called an epoch, can be found in the receiver clock time scale. For an illustration of this, see figure 2.6.

Message that includes the Time of Transmission It is the TOT of the code epoch (in GPS system time scale) and the satellite position at that time.

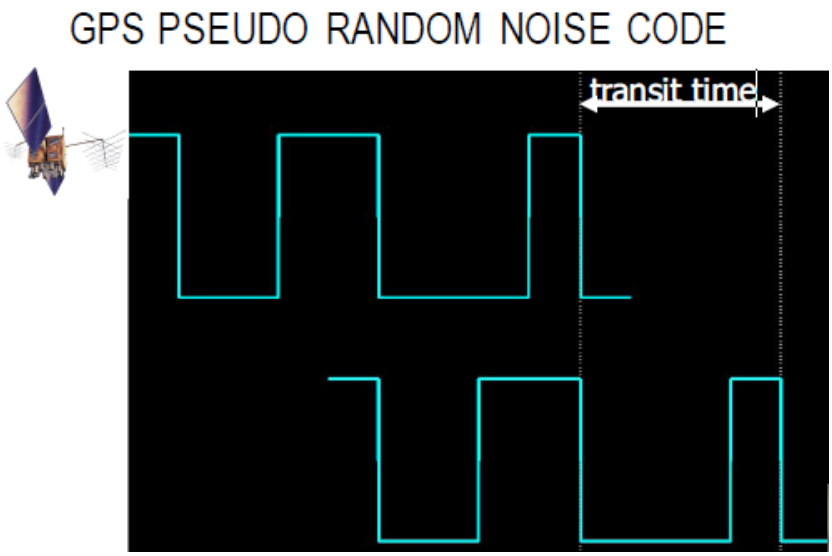


Figure 2.6: Pseudo random code. Satellite and receiver generate same code at same time. Once satellite signal arrives, receiver checks how long ago the received code was generated. Figure taken from [18]

The method is the following: the receiver measures the TOAs (according to its own clock) of four satellite signals. From the TOAs and the TOTs, the receiver forms four time of flight (TOF) values, which are (given the speed of light) approximately equivalent to receiver-satellite range differences. The receiver then computes its three-dimensional position and clock deviation from the four TOFs.
The computation is summarized with the equation (2.11), where:

ρ^s is the pseudo.range of each satellite (at least four).

x^s, y^s, z^s are the Earth centered satellite coordinates, already known thanks to the ephemerides.

b is the clock error (converted into distance).

x,y, z the device coordinates.

$$\rho^s = \sqrt{(x^s - x)^2 + (y^s - y)^2 + (z^s - z)^2} + b \tag{2.11}$$

In practice the receiver position (in three dimensional Cartesian coordinates with origin at the Earth's center) and the offset of the receiver clock relative to the GPS time are computed simultaneously, using the navigation equations to process the TOFs [13].

For an illustration of GPS trilateration, see figure 2.7.

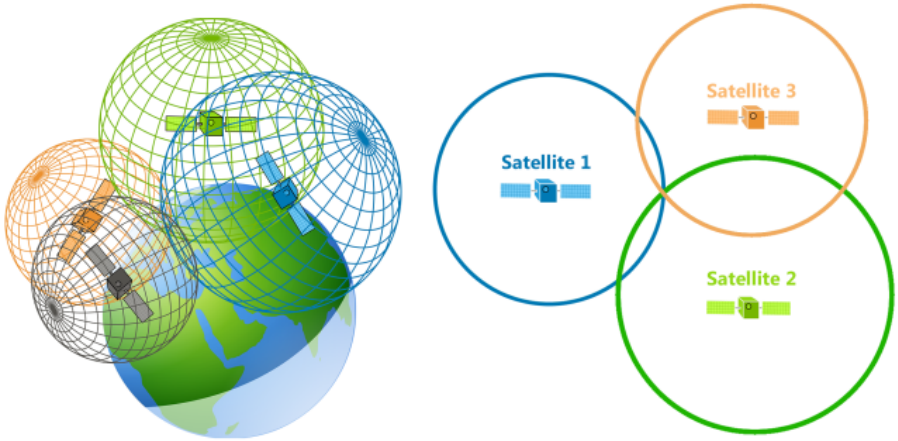


Figure 2.7: Sketch of GPS Trilateration. The intersection point in the three spheres determines the device position. Figure extracted from [12].

2.4. Assisted GPS

Assisted GPS, also known as A-GPS or AGPS, enhances the performance of standard GPS in devices connected to the cellular network. A-GPS improves the location performance of cell phones (and other connected devices) in three ways [69]:

- By helping obtain a faster "time to first fix" (TTFF). A-GPS acquires and stores information about the location of satellites via the cellular network so the information does not need to be downloaded via satellite.
- By helping position a phone or mobile device when Assisted GPS signals are weak or not available. GPS satellite signals may be impeded by tall buildings, and do not penetrate building interiors well. A-GPS uses proximity to cellular towers to calculate position when GPS signals are not available.
- Obtaining time synchronization with the Mobile Network.

Standalone GPS provides first position in approximately 30–40 seconds. A standalone GPS needs orbital information of the satellites to calculate the current position. The data rate of the satellite signal is only 50 bit/s, so downloading orbital information like ephemerides and the almanac directly from satellites typically takes a long time, and if the satellite signals are lost during the acquisition of this information, it is discarded and the standalone system has to start from scratch. In A-GPS, the network operator deploys an A-GPS server. These A-GPS servers download the orbital information from the satellite and store it in the database. An A-GPS capable device can connect to these servers and download this information using mobile network radio bearers such as GSM, CDMA, WCDMA, LTE or even using other wireless radio bearers such as WiFi. Usually the data rate of these bearers is high, hence downloading orbital information takes less time [57].

2.5. Received Signal Strength Indication

Received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal [55]. The RSS values are measured in dBm and have typical negative values ranging between 0 dBm (excellent signal) and -110 dBm (extremely poor signal) [41].

The distance is estimated in relation to the strength of the received signal. Estimating the distance to three nearby towers and using trilateration, the position is obtained [28].

Let's suppose that s(t) of power P_t is transmitted through a given channel. The received signal r(t) of power P_r is averaged over any random variations due to shadowing. We define the linear path loss of the channel as the ratio transmit power to receiver power [56] [53].

$$P_L = \frac{P_t}{P_r} \tag{2.12}$$

And defined in dB:

$$P_L[dB] = 10\log_{10}\frac{P_t}{P_r} \tag{2.13}$$

The RSS decreases (not linearly) with the distance between the node that is receiving the signal and the device that is transmitting the signal.
The RSS detected by the nodes are affected by many factors, including [3]:

- The antenna of the device that is transmitting.
- The antenna of the node itself.
- The number of walls and other obstructions in proximity of the nodes.
- The presence of water in proximity of the nodes.
- The material of the objects inside the environment.
- The number of people.

This is due to:

Reflection Change in direction of a wavefront at an interface between two different media so that the wavefront returns into the medium from which it originated [42].

Diffraction Bending of light around the corners of an obstacle or aperture into the region of geometrical shadow of the obstacle. It occurs when the radio path between transmitter and receiver is obstructed by a surface that has sharp edges [56].

Scattering The radio wave is forced to deviate from a straight trajectory by one or more paths due to localized non-uniformities in the medium through which they pass [22].

Considering there are no obstacles, the free space propagation is:

$$P_r(d) = P_t K\left(\frac{d_0}{d}\right)^{\gamma} \tag{2.14}$$

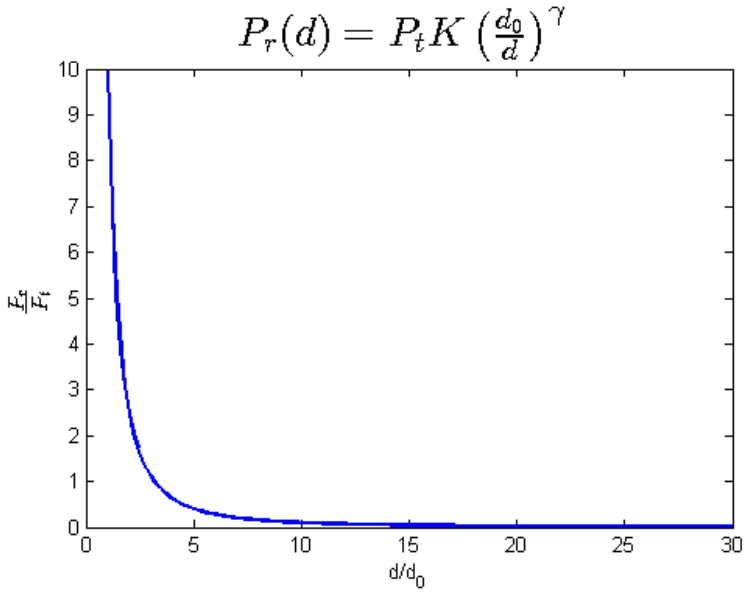


Figure 2.8: Free space propagation. This graph is equation (2.14) representation for a value of $K = 10$ and $\gamma = 2$.

K is a dimensionless constant that depends on the antenna characteristics and free space path loss up to distance d_0. γ is the path loss coefficient [53].

This equation in decibels is:

$$P_r[dBm] = P_t[dBm] + K[dB] - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) \tag{2.15}$$

Observing the received power, and knowing the emitting power and the loss factor (i.e transmitting power is provided by the telecom company and power loss factor is taken from tables of previous researches), the distance could be determined comparing the measurements to a reference.

2.6. IP Address Location

This one will detect your location based on nearest Public IP Address on your devices. They can be your computer, your router, or your ISP provider. Depend on the IP information available, but in many case where the IP is hidden behind Internet Service Provider NAT, the accuracy is in level of city, region, or even country [10].

There is a public database, Wigle, which stores SSID of wireless networks, linked to their locations. Google can use this database or its own to locate mobile devices based on the WiFi networks nearby.

3

Multi linear regression theory

In this chapter theoretical concepts applied in Methodology and Results are explained. First what a multi linear regression model is, and second the five steps followed in methodology to generate it.

3.1. Multiple regression model

3.1.1. Brief introduction to Linear regression

The simplest idea of linear regression summarizes the relationship between a quantitative predictor variable (x) and a quantitative response variable (y) with a straight line [24]. This model can be extended to handle:

- Several explanatory variables
- Categorical independent variables and interactions between independent variables
- Non linear relationships.

It is important to note that regression models with observational data can only describe outcomes of processes, but they cannot explain them.

3.1.2. Description and assumptions

The structure of the data is established by the following equation [60](process that generated the observations):

$$y_i = \alpha + A_{i1}x_1 + A_{i2}x_2 + \cdots + A_{in}x_n + e_i \qquad \text{for i = 1,2,...,m} \tag{3.1}$$

Where m is the number of observations and n the number of predictor variables or regressors represented by A_{ij}.
A_{ij} is the value of the regressor j in the experiment i.
The parameters (x_i) represent partial effects. Each slope is the effect of the corresponding regressor holding all other predictor variables in the model constant. These parameters define a hyperplane.
e_i is the perturbation error term and α the intercept. e_i is the distance from the observations $\mathbf{Y}$ to the hyperplane. The objective of the least squares method is to find the hyperplane that better adjusts to the observations.

To clarify with an example, y_i is the accuracy provided by Google, A_{i1}, A_{i2}, and A_{i3} are the distances to the three nearest cell towers and α, x_1, x_2 and x_3 the coefficients we have to calculate to define the model. i would represent the experiment, which is each entry in the Google Timeline History. This will be explained in chapter 5.

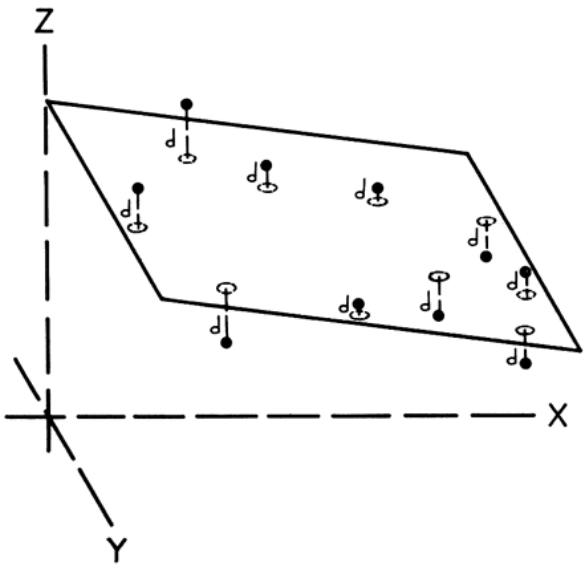


Figure 3.1: Graphic interpretation of the adjusted hyperplane.This figure represents the linear adjustment of z variable for 10 observations. The x, y variables are the regressors. The distances "d" from observations to the plane represent the residuals e and the projections of the observations on the plane are the predicted values. Picture taken from [30]

Then it is necessary to adopt the assumptions of the ordinary least squares regression (OLS) about the errors:

1. $E(e_i) = 0$ for i = 1, 2 , . . . , m. - Expectancy equals zero.
2. $e_i \sim N(0, \sigma^2)$ for i = 1, 2 , . . . , m. - Normally distributed errors.
3. A variables have to be independent of errors.

If these assumptions are met, the OLS estimators are unbiased and efficient estimates of population parameters.

With this model we can not only do approximations with the original variables, but we also can extend it to quadratic, cubic, etc terms or even cross products. In some cases, this will allow for a better fit of the model.

3.1.3. Ordinary Least Squares

If we substitute α with x_0, the general linear model used in equation (3.1) takes the following form [48] [24]:

$$y_i = x_0 + A_{i1}x_1 + A_{i2}x_2 + \cdots + A_{in}x_n + e_i \qquad \text{for i = 1,2,. . . ,m} \tag{3.2}$$

The equivalent version in matrix form would be:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} 1 & A_{11} & \dots & A_{1n} \\ 1 & A_{21} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & A_{m1} & \dots & A_{mn} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix} \tag{3.3}$$

Or:

$$\underset{m\times 1}{\mathrm{y}} = \underset{m\times(n+1)}{\mathbf{A}} \times \underset{(n+1)\times 1}{x} + \underset{m\times 1}{e} \tag{3.4}$$

A is called the model matrix, because it contains all the values of the explanatory variables for each observation in the data.

Every coefficient $x_i (i \geq 1)$ measures the marginal effect that over the response variable y when a predictor variable A_i is incremented leaving the rest of the variables A_j constant, with $j \neq i$.

With the assumptions of subsection 3.1.2 ($e_i \sim N(0,\sigma^2)$), now in vector form converts to $\underline{e} \sim N_n(0,\sigma_e^2\mathbf{I_n})$.

Since the $\underline{e}$ are dependent on the conditional distribution of $\underline{y}$, $\underline{y}$ is also normally distributed with mean and variance as follows [60]:

$$\begin{aligned}\mu &\equiv E(\underline{y}) \\ &= E(\mathbf{A}x + \underline{e}) \\ &= \mathbf{A}x + E(\underline{e}) = \mathbf{A}x\end{aligned} \tag{3.5}$$

$$\begin{aligned}D(\underline{y}) &= E[(\underline{y}-\mu)(\underline{y}-\mu)^T] \\ &= E[(\underline{y}-\mathbf{A}x)(\underline{y}-\mathbf{A}x)^T] \\ &= E(\underline{ee}^T) = \sigma_e^2\mathbf{I_m}\end{aligned} \tag{3.6}$$

Therefore $\underline{y} \sim N_m(\mathbf{A}x, \sigma_e^2\mathbf{I_m})$.

The fitted linear model is then:

$$\begin{aligned}\hat{y} &= \mathbf{A} \times \hat{x} \\ \hat{e} &= y - \hat{y}\end{aligned} \tag{3.7}$$

$$\underline{y} = \mathbf{A}\hat{x} + \underline{e} \tag{3.8}$$

Where $\hat{x}$ is the vector of fitted slope coefficients and $\underline{e}$ is the vector of residuals. The purpose of OLS is to minimize the residual sum of squares:

$$\begin{aligned}S(\hat{x}) &= \sum_{i=1}^{m} {e_i}^2 = \underline{e}^T\underline{e} \\ &= (\underline{y}-\mathbf{A}\hat{x})^T(\underline{y}-\mathbf{A}\hat{x}) \\ &= \underline{y}^T\underline{y} - (2\underline{y}^T\mathbf{A})\hat{x} + \hat{x}^T(\mathbf{A}^T\mathbf{A})\hat{x}\end{aligned} \tag{3.9}$$

To minimize $S(\hat{x})$, we have to equal the partial derivative with respect to $\hat{x}$ to zero.

$$\frac{\partial S(\hat{x})}{\partial \hat{x}} = 0 - 2\mathbf{A}^T\underline{y} + 2\mathbf{A^T}\mathbf{A}\hat{x} \tag{3.10}$$

If $\mathbf{A^T}\mathbf{A}$ is not singular (rank of n+1) we can uniquely solve for the least-squares coefficients:

$$\underline{\hat{x}} = \begin{pmatrix}\hat{x_0} \\ \hat{x_1} \\ \vdots \\ \hat{x_n}\end{pmatrix} = (\mathbf{A^T}\mathbf{A})^{-1}\mathbf{A}^T\underline{y} \tag{3.11}$$

$\mathbf{A^T}\mathbf{A}$ is always a squared and symmetric matrix. The rank of $\mathbf{A^T}\mathbf{A}$ is equal to the rank of $\mathbf{A}$. This leads to two criteria that must met in order to ensure that $\mathbf{A^T}\mathbf{A}$ is not singular, and thus obtain an unique solution 48:

1. $m \geq n+1$ - We need at least as many observations as there are coefficients in the model.
2. Columns of $\mathbf{A}$ must not be linearly related (i.e. $\mathbf{A}$ variables must be independent).
3. We have to consider that no regressor other than the constant can be invariant. An invariant regressor would be a multiple of the constant.

3.1.4. Distribution of the least-squares Estimator

We can say now that $\underline{\hat{x}}$ is a linear estimator of x.

$$\underline{\hat{x}} = (\mathbf{A^T A})^{-1}\mathbf{A}^T \underline{y} = \mathbf{M}\underline{y} \tag{3.12}$$

Establishing the expectation of $\underline{\hat{x}}$ from the expectation of $\underline{y}$, we see the $\underline{\hat{x}}$ is an unbiased estimator of x:

$$E(\underline{\hat{x}}) = E(\mathbf{M}\underline{y}) = \mathbf{M}E(\underline{y}) = (\mathbf{A^T A})^{-1}\mathbf{A}^T(\mathbf{A}x) = x \tag{3.13}$$

Solving for the variance of $\underline{\hat{x}}$, we find that it depends only on the model matrix and the variance of the errors:

$$\begin{aligned} D(\underline{\hat{x}}) &= \mathbf{M}D(\underline{y})\mathbf{M}^T \\ &\text{Using equation (3.6):} \\ &= [(\mathbf{A^T A})^{-1}\mathbf{A}^T]\sigma_e^2\mathbf{I}_n[(\mathbf{A^T A})^{-1}\mathbf{A}^T]^T \\ &\text{Taking into account that } \mathbf{A^T A} \text{ is symmetric:} \\ &= \sigma_e^2(\mathbf{A^T A})^{-1}(\mathbf{A^T A})(\mathbf{A^T A})^{-1} \\ &= \sigma_e^2(\mathbf{A^T A})^{-1} \end{aligned} \tag{3.14}$$

The variances of the $\underline{\hat{x}}$ elements are expressed in terms of the elements of the estimators of the inverse of the $\mathbf{A^T A}$ matrix. The inverse of $\mathbf{A^T A}$ times the constant σ^2 represents the variance matrix of the regression coefficients $\hat{x}$. The diagonal elements of $\sigma^2(\mathbf{A^T A})^{-1}$ are the variances of the estimators $\underline{\hat{x}}$. The off-diagonal elements of this matrix are the covariances [48].

$$\begin{aligned} \mathbf{C} &= (\mathbf{A^T A})^{-1} \\ D(\underline{\hat{x}_i}) &= \sigma_e^2 C_{ii} \\ cov(\underline{\hat{x}_i}\underline{\hat{x}_j}) &= \sigma_e^2 C_{ij} \end{aligned} \tag{3.15}$$

The value of the variance helps to indicate the precision of the estimation of the model. It has to be compared with the x value of the coefficients. The smaller it is, the coefficients $\underline{\hat{x}_i}$ are calculated with better precision.

Finally, if $\underline{y}$ is normally distributed, the distribution of $\underline{\hat{x}}$ is [48]:

$$\underline{\hat{x}} \sim N_{n+1}[x, \sigma_e^2(\mathbf{A^T A})^{-1}]$$

3.1.5. Dummy variables

Linear regression can be extended to accommodate categorical variables (factors) using dummy variable regressors (or indicator variables). For example, in the following equation, a categorical variable is presented by a dummy regressor D (coded 1 for one category, 0 for the other)[27][48]:

$$y_i = \alpha + A_i x + \gamma D_i + e_i \tag{3.16}$$

This fits two regression lines with the same slope (A_i) but different intercepts (i.e. coefficient γ represents the constant separation between the two regression lines). This is used to represent two models with a simple difference (the value of a qualitative variable) in a single one.

Linear regression can be extended to accommodate categorical variables (factors) using dummy variable regressors.

3.1.6. Wilkinson notation

To describe a linear model without specifying the coefficient values, Wilkinson notation is used [67]. With this notation one can specify the response variable and the regressors used to define a linear model. One can define a model using the regressors themselves, and combination of products among regressors, or powers of the variables (regressors). This notation is used in model calculation software like Matlab®and will be used

in this thesis. For example, to define a linear model of response variable *y* and regressors *var1* and *var2*, the normal notation would be:

$$y = x_0 + var1 \cdot x_1 + var2 \cdot x_2 \tag{3.17}$$

in Wilkinson notation would be:

$$y \sim 1 + var1 + var2 \tag{3.18}$$

The variable (or variables) at the left of ~ sign is (are) the response variable(s). On the right there are the regressor names. The *1* term means that the model will have the intercept (x_0). By default the intercept is included, so the *1* can be omitted. The + sign indicates a variable to be considered in the model, and a minus (-) sign means that variable is not included in the model. For example, if three variables are available (*var1, var2, var3*) but we want to build a model with only *var1, var3*, in normal notation would be:

$$y = var1 \cdot x_1 + var3 \cdot x_3 \tag{3.19}$$

in Wilkinson notation would be:

$$\begin{aligned} &y \sim -1 + var1 - var2 + var3 \\ &\quad or \\ &y \sim -1 + var1 + var3 \end{aligned} \tag{3.20}$$

The "*" operator (for interactions) and the "^" operator (for power and exponents) automatically include all lower-order terms.

To include only an interaction (product of two variables) without including the factor variables ":" sign is used[59]. See table 3.1.

Terms to add to the model	Wilkinson notation	Optional notation
intercept, v1, v2	v1+v2	1+v1+v2
$v1, v1^2$	$v1^2$	$v1 + v1^2$
v1, v2, v1 · v2	v1*v2	v1+v2+v1:v2
v1 · v2	v1:v2	v1*v2 - v1 - v2
v1, v1 · v2, v2 · v3, v1 · v3	v1*v2*v3 -v2-v3 - v1:v2:v3	

Table 3.1: Wilkinson notation examples. In this table some examples are shown of which factors are considered when using "*", ":", and "^" operators. By default, *intercept* term is included. "*" and "^" include al lower-order terms

3.2. Generating a model for multi-linear regression

To develop the multi linear model, a road map of five steps was followed:

1. Check the data
2. Select variables
3. Test model
4. Correct model problems
5. Validate model

3.2.1. Check the data

The first step to develop a linear model regression, is to prepare and check the data. The data is checked because it is not read directly from a source. It is read from different sources and then combined into a single table. Any mistake done during conversion has to be detected as soon as possible. In order to do that, we'll manage the concepts of mean, median, RMS, and percentile.

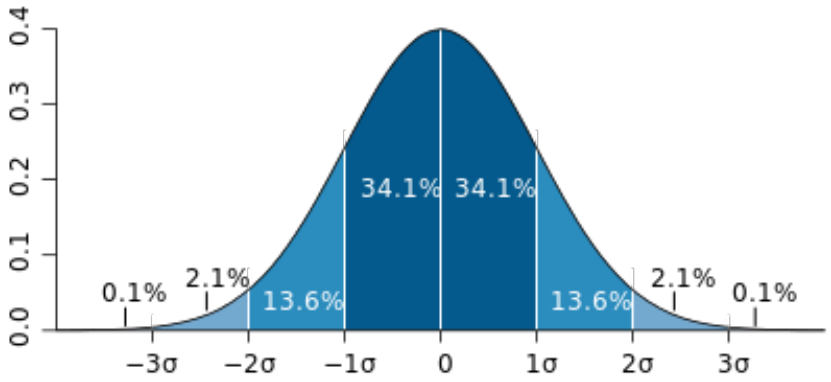


Figure 3.2: Standard deviation. A plot of normal distribution (or bell-shaped curve) where each band has a width of 1 standard deviation. Image extracted from [66]

Mean and Standard deviation

In statistics, the standard deviation (σ or S) is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values [60].

$$\sigma = \sqrt{\frac{\sum_1^n (z_i - \mu)^2}{n-1}} \tag{3.21}$$

Being the mean (μ):

$$\mu = \frac{\sum_1^n z_i}{n} \tag{3.22}$$

Median

The median of a set of values (z_i for example) is the value separating the higher half of the data sample, a population, or a probability distribution, from the lower half. The reason to use the median in describing data instead of the average ($\overline{z} = \frac{z_1+z_2+\cdots+z_m}{m}$) is because the average is more skewed by extremely large or small values than the median, and it may give a better idea of a "typical" value[48][46].

Root Mean Square

Abbreviated as RMS, it is the square root of the arithmetic mean of the squares of the values. In econometrics the root mean square error of an estimator is a measure of the imperfection of the fit of the estimator to the data[48][16].

$$RMS = \sqrt{\frac{z_1^2 + z_2^2 + \cdots + z_m^2}{m}} \tag{3.23}$$

Percentile

A percentile is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations fall. For example, the 20^{th} percentile is the value (or score) below which 20% of the observations may be found. The median explained above corresponds to percentile 50 [51].

Purpose

When the experiment data is retrieved from different sources (Google Timeline, GPS receiver), the data is evaluated using these statistics to find reading errors. For example, if all the points have a mean latitude of 51 degrees, with a σ of 0.05 degrees, a value less than 50.80 (outside of the interval mean $\pm$ 3 σ) is considered an error in the measurement.

In the first chapter of results 6 we use these statistical concepts to see the aspect of the data collected and the variables under study. If there are any outstanding outliers or excessive dispersion it would become easier to see with these computations.

3.2.2. Select variables

After checking that data is applicable, with enough quality, and of decent amount, then starts the building of the model. In this part, we choose the best variables for the model (i.e.; the variables that have the most direct relationships with the chosen response variable). The aim when selecting variables is to collect the maximum amount of information possible from a minimum number of variables[48].

If two independent variables are measured in exactly the same units, we can assess their relative importance in their effect on **y** quite simply, the larger the coefficient, the stronger the effect. But explanatory variables often are not all measured in the same units, making it difficult to assess relative importance. This problem is solved using standardized dimensionless variables.

This search will be done evaluating the terms of R^2, R^2 adjusted or $\tilde{R^2}$, C_p and S.

Test for significance of regression

The test of significance regression is a test to determine whether a linear relationship exists between the response variable y and a subset of the regressor variables $A_1, A_2, \ldots, A_n$.

Intermezzo: Hypothesis testing

In order to undertake hypothesis testing you need to express your research hypothesis as a null and alternative hypothesis. The null hypothesis and alternative hypothesis are statements regarding the differences or effects that occur in the population. You will use your sample to test which statement (i.e., the null hypothesis or alternative hypothesis) is most likely (although technically, you test the evidence against the null hypothesis) [48] [58].

The null hypothesis assumes that whatever you are trying to prove did not happen, it is normally represented by H_0.

The level of statistical significance is often expressed as the so-called p-value. Depending on the statistical test you have chosen, you will calculate a probability (i.e., the p-value) of observing your sample results (or more extreme) given that the null hypothesis is true.

The p-value is the probability that the test statistic will take on a value that is at least as extreme as the observed value of the statistic when the null hyphothesis H_0 is true. Thus, a p-value conveys much information about the weight of evidence against H_0, and so a decision maker can draw a conclusion at any specified level of significance.

Whilst there is relatively little justification why a significance level of 0.05 is used rather than 0.01 or 0.10, for example, it is widely used in academic research. However, if you want to be particularly confident in your results, you can set a more stringent level of 0.01 (a 1% chance or less; 1 in 100 chance or less).

The appropriate hypothesis are:

$$\begin{aligned} &H_0 : x_1 = x_2 = \cdots = x_n = 0 \\ &H_a : x_i \neq 0 \text{ for at least one } i \end{aligned} \tag{3.24}$$

Rejection of H_0 implies that at least one the regressor variables contributes significantly to the model.

The test for significance of regression is performed studying the variance of the errors. The total sum of squares is partitioned into a sum of squares due to regression and a sum of squares due to error.

$$\begin{aligned} SS_T &= \sum_{i=1}^{m} (y_i - \bar{y})^2 = \hat{e_0}^T \hat{e_0} \\ SS_E &= \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \hat{e}^T \hat{e} \\ SS_R &\equiv SS_T - SS_E = \sum_{i=1}^{m} (\hat{y}_i - \bar{y})^2 \end{aligned} \tag{3.25}$$

Now, if H_0 is true, SS_R/σ^2 is a chi-square with n degrees of freedom. The number of degrees of freedom for this chi-square random variable is equal to the number of regressor variables in the model except the

	Degrees of Freedom
SS_T	m-1
SS_E	m-n-1
SS_R	n

intercept x_0. It can also be shown that SS_E/σ^2 is a chi-square random variable with m-n-1 degrees of freedom. SS_E and SS_R are independent.

To determine if we should reject H_0, we will use the F-test. In statistics, F-test (or Snedecor test) is a test where the estimate follows a F distribution if the null hypothesis cannot be rejected. F distribution is defined as:

$$F = \frac{U_1/d_1}{U_2/d_2} \tag{3.26}$$

Where U_1 and U_2 are chi-squared distributions with d_1 and d_2 degrees of freedom respectively. Also, they have to be statistically independent.

Using the expressions:

$$\begin{aligned} MS_R &= \frac{SS_R}{n} \\ MS_E &= \frac{SS_E}{m-n-1} \end{aligned} \tag{3.27}$$

The test statistic for H_0 is:

$$F_0 = \frac{\frac{SS_R}{\sigma^2}/n}{\frac{SS_E}{\sigma^2}/(m-n-1)} = \frac{MS_R}{MS_E} \tag{3.28}$$

We should reject H_0 if the computed value of the test statistic in equation (3.28), f_0, is greater than $f_{\alpha,n,m-n-1}$. Being α the confidence interval.

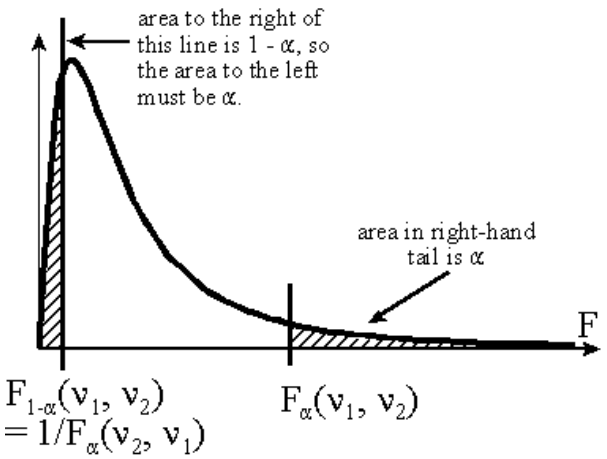


Figure 3.3: Example of F distribution with explanation of confidence areas α. Picture extracted from [54]

R - Correlation coefficient

The ratio of SS_R to SS_T gives us the proportional reduction in squared error associated with the regression model. This also defines the square of the correlation coefficient[48]:

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} \tag{3.29}$$

Note that R is dimensionless.

Properties:

- $0 \leq R^2 \leq 1$.
- When $R^2 = 1$, there is an exact relation between response y and n regressor variables. That is to say that all the observations fall in the hyperplane.
- When $R^2 = 0$, $\hat{x}_0 = \bar{y}$ and $\hat{x}_1 = \cdots = \hat{x}_n = 0$. There is no linear relation between y and $\mathbf{A}_i$.

Adjusted R^2

R^2 will always rise as more explanatory variables are added to a model. It will never decline. As an alternative, there is an adjusted R^2 that corrects for the degrees of freedom (i.e. the number of explanatory variables):

$$\tilde{R}^2 \equiv 1 - \frac{S_E^2}{S_Y^2} = 1 - \frac{\frac{SS_E}{m-n-1}}{\frac{SS_T}{m-1}} \tag{3.30}$$

Because $\frac{SS_E}{m-n-1}$ is the error mean square and $\frac{SS_T}{m-1}$ is a constant, it only depends on the observations and their average, $\tilde{R}^2$ will only increase when a variable is added to the model if the new variable reduces the mean square of the residual errors.

The adjusted R^2 statistic essentially penalizes the analyst for adding terms to the model. It is an easy way to guard to over-fitting (i.e. including regressors that are not really significant). Consequentially, it is quite useful in comparing and evaluating competing regression models[48].

C_p statistic

If a set of p regressors are selected from n available, C_p is a measure for the total mean square for the regression model compared to the model which contains all the available regressors. C_p is defined as:

$$C_p = \frac{SS_{E(p)}}{\frac{\hat{e}^T\hat{e}}{m}} - m + 2p \tag{3.31}$$

Being:

$SS_{E(p)}$ the sum of squared residuals for the model with p regressors.

$$SS_{E(p)} = \hat{e_p}^T \hat{e_p} \tag{3.32}$$

The smaller the C_p value, the less total mean square error, and the p chosen regressors give a better estimate of the model coefficients[48].

Then, to select the good set of variables, we will mainly focus on which $\tilde{R}^2$ is best. To do that, we will use all-possible-regressions to test all possible subsets of potential predictor variables. With the all-possible-regressions method, numerical criteria will be examined as follows:

R^2 The set of variables with the highest R^2 value are the best fit variables for the model.

$\tilde{R}^2$ The sets of variables with larger adjusted R^2 values are the better fit variables for the model.

C_p The smaller the C_p value, the less total mean square error, and the model is more precise.

To select the appropriate regressors, first, the model with all n regressors is calculated. After that, all possible models with all possible p regressors are calculated, and the values of R^2, $\tilde{R}^2$, C_p are compared and then the regressors are chosen.

Test model

Once the set of variables is selected, then the model has to be tested. For this, we will check the global F-test.

Global F-test

Same method as explained in 3.2.2. We also will check $\tilde{R^2}$ of the full model. To choose which $\tilde{R^2}$ is good enough for the model, we have to consider that in some situations the variables under consideration have very strong and intuitively obvious relationships, while in other situations you may be looking for very weak signals in very noisy data. Roughly speaking, the error measures become percentages rather than absolute amounts. Lomax and Hahs-Vaughn [43]

Moreover, variance is a hard quantity to think about because it is measured in squared units. It is easier to think in terms of standard deviations, because they are measured in the same units as the variables and they directly determine the widths of confidence intervals. So, it is instructive to also consider the "percent of standard deviation explained," i.e., the percent by which the standard deviation of the errors is less than the standard deviation of the dependent variable. This is equal to one minus the square root of 1-minus-R-squared.

$$\frac{\sigma_T - \sigma_E}{\sigma_T} = 1 - \frac{\sigma_E}{\sigma_T} = 1 - \sqrt{\frac{SS_E}{SS_T}} = 1 - \sqrt{1 - R^2} \tag{3.33}$$

3.2.3. Testing model assumptions and outliers

Removing outliers

To detect outliers, we will mainly focus on computing Cook's distance. Cook's distance is a commonly used estimate of the influence of a data point when performing a least-squares regression analysis [20] and [19]. In a practical ordinary least squares analysis, Cook's distance can be used in several ways: to indicate influential data points that are particularly worth checking for validity; or to indicate regions of the design space where it would be good to be able to obtain more data points. Cook's distance is useful for identifying outliers in the observation values (observations for independent and dependent variables). Cook's distance of an observation is the product of the distance of the observation to the centroid of the rest of observations (how far it is from the rest) and the variation of the predicted value for this observation with respect of the same prediction when that observation is not considered in the model (how influential this observation is). So it shows the influence of each observation on the fitted response values and its leverage. An observation with Cook's distance larger than three times the mean Cook's distance might be an outlier.

Cook's distance is defined as:

$$D_i = \frac{\sum_{j=1}^{m} (\hat{y}_j - \hat{y}_{j(i)})^2}{n * MS_E} \qquad \text{With i = 1,2,... m} \tag{3.34}$$

Where:

$\hat{y}_j$ It's the j^{th} fitted response value.

$\hat{y}_{j(i)}$ It's the j^{th} fitted response value, where the fit does not include observation *i*.

MS_E It's the mean squared error, see equation (3.27).

n It is the number of coefficients in the regression model without including the intercept.

This method is equivalent to *w-test*, under the condition that σ is not known [58].

First we have the statistic T_q for H_a: Being the null Hypothesis H_0 to take all observations and the alternative Hypotesys H_a, to exclude one observation (the i^{st}).

$$\begin{aligned} \underline{T}_q &= \underline{\hat{e}}^T Q_{yy}^{-1} c_y (c_y^T Q_{yy}^{-1} Q_{\hat{e}_0 \hat{e}_0} Q_{yy}^{-1} c_y)^{-1} c_y^T Q_{yy}^{-1} \underline{\hat{e}} \\ &= (\underline{\hat{y}}_0 - \underline{\hat{y}}_a)^T Q_{yy}^{-1} (\underline{\hat{y}}_0 - \underline{\hat{y}}_a) \end{aligned} \tag{3.35}$$

Where:

$\underline{\hat{e}}$ the observation minus the predicted values $\underline{y} - \underline{\hat{y}}$

Q_{yy} is variance matrix, $\sigma^2 I_m$

c_y is the canonical vector to express the i^{st} observation as an outlier (0,0,...,1,...,0)

$Q_{\hat{e}_0\hat{e}_0}$ is $P_A^{\perp}Q_{yy}$ with $P_A^{\perp} = I_m - A(A^TQ_{yy}^{-1}A)A^TQ_{yy}^{-1}$

When σ is not know it can be considered as a new parameter to determine, and the new value for T_q is:

$$T_q = \frac{(\underline{\hat{y}}_0 - \underline{\hat{y}}_a)^T Q_{yy}^{-1}(\underline{\hat{y}}_0 - \underline{\hat{y}}_a)}{q\hat{\sigma_a}^2} = \tag{3.36}$$

Inserting
Q_{yy}^{-1} as $\frac{I_m}{\sigma^2}$
q as n
$\hat{\sigma}^2 = MS_E$ See reference [48]
We obtain:

$$T_q = \frac{\sum_{i=1}^{m}(\underline{\hat{y}}_{0i} - \underline{\hat{y}}_{ai})^2}{n\sigma^2} \tag{3.37}$$

which is equivalent to equation (3.34). With this it is proven that Cook's distance method and w-test are equivalent for removing outliers.

Other method to detect outliers is to look for the observations which have largest errors. To determine what a "big error" is, Pearson's residuals are used. As the error magnitude depends on the response variable itself, an adimensional error measure is needed.

Pearson's residuals are the Raw residuals divided by the square root of the mean squared errors [21]. So, it is a dimensionless magnitude and it is a common practice to define observations with Pearson's residual, in absolute value, greater than 3 as outliers.

$$\hat{e}_{pr} = \frac{\hat{e}}{\sqrt{MS_E}} = \frac{\hat{e}}{\sqrt{\frac{\hat{e}^T\hat{e}}{m-n-1}}} \approx \frac{\hat{e}}{\hat{\sigma}_{\hat{e}_i}} \tag{3.38}$$

The pr in the equation stands for Pearson error.

Testing model

The linear model must meet a set of assumptions.If any of these assumptions is violated (i.e., if there are nonlinear relationships between dependent and independent variables or the errors exhibit correlation, heteroscedasticity, or non-normality), then the forecasts, confidence intervals, and scientific insights yielded by a regression model may be (at best) inefficient or (at worst) seriously biased or misleading [65].

Violations of linearity or additivity If you fit a linear model to data which are nonlinearly or non additively related, your predictions are likely to be seriously in error, especially when you extrapolate beyond the range of the sample data. It is usually most evident in a plot of observed versus predicted values or a plot of residuals versus predicted values. The points should be symmetrically distributed around a diagonal line. Evidence has to be searched of a "bowed" pattern, indicating that the model makes systematic errors whenever it is making unusually large or small predictions. To fix it, it can be considered applying a nonlinear transformation to the dependent and/or independent variables if a transformation that seems appropriate is possible. Another possibility is adding another regressor that is a nonlinear function of one of the other variables (i.e. regress y on both $\mathbf{A}$ and $\mathbf{A^2}$). It may also be that some entirely different independent variable has been overlooked, or interactions among variables have been ignored.

Violations of independence The best test for serial correlation is to look at a residual time series plot (residuals vs. row number) and a table or plot of residual autocorrelations. Ideally, most of the residual autocorrelations should fall within the 95% confidence bands around zero. It can indicate that there is some room for fine-tuning in the model.

Violations of homoscedasticity It is difficult to gauge the true standard deviation of the forecast errors, usually resulting in confidence intervals that are too wide or too narrow. In particular, if the variance of the errors is increasing over time, confidence intervals for out-of-sample predictions will tend to be unrealistically narrow. Heteroscedasticity may also have the effect of giving too much weight to a small subset

of the data (namely the subset where the error variance was largest) when estimating coefficients [29]. It is seen in a plot of residuals versus predicted values and, in the case of time series data, a plot of residuals versus time. It is necessary to be alert for evidence of residuals that grow larger either as a function of time or as a function of the predicted value. Because of imprecision in the coefficient estimates, the errors may tend to be slightly larger for forecasts associated with predictions or values of independent variables that are extreme in both directions, although the effect should not be too significant. What is expected is not to see errors that systematically get larger in one direction by a significant amount.

Violations of normality They create problems for determining whether model coefficients are significantly different from zero and for calculating confidence intervals for forecasts. Sometimes the error distribution is "skewed" by the presence of a few large outliers. Since parameter estimation is based on the minimization of squared error, a few extreme observations can exert a disproportionate influence on parameter estimates. Calculation of confidence intervals and various significance tests for coefficients are all based on the assumptions of normally distributed errors. If the error distribution is significantly non-normal, confidence intervals may be too wide or too narrow. the best test for normally distributed errors is a normal probability plot or normal quantile plot of the residuals.

3.2.4. Validating the model

The model will be validated using the cross validation k-fold method.

Cross-validation is a model assessment technique used to evaluate a machine learning algorithm's performance in making predictions on new datasets that it has not been trained on [52]. This is done by partitioning a dataset and using a subset to train the algorithm and the remaining data for testing. Because cross-validation does not use all of the data to build a model, it is a commonly used method to prevent overfitting during training.

Each round of cross-validation involves randomly partitioning the original dataset into a training set and a testing set. The training set is then used to train a supervised learning algorithm and the testing set is used to evaluate its performance. This process is repeated several times and the average cross-validation error is used as a performance indicator.

In the technique **k-fold**: partitions data into k randomly chosen subsets (or folds) of roughly equal size. One subset is used to validate the model trained using the remaining subsets. This process is repeated k times such that each subset is used exactly once for validation. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

4

Experiments

This chapter is meant to explain and describe the experiments carried out during the study to answer the research questions. Equipment, methodology, measured parameters and routes are explained. Google radius accuracy and error made compared to "Ground truth data" is measured. These measures are statistically analyzed in the following chapters (chapter 6) and used as an input for the linear regression model (chapter 7).

4.1. Motivation

The goals of this thesis is to find out if the location provided by Google Location History is more precise or not. That is to say if when Google Timeline says that a mobile phone was inside a determined area, it is likely to be true or not. This information can be divided in three sets:

Provided accuracy The provided accuracy is the radius Google Timeline application gives when registering a position. It is expressed in meters, and it is represented as the radius of the circle around the provided location.

Google error Google error is the real distance between the location provided by Google Timeline and the actual position at that moment. It is calculated based on the position provided by a reliable GPS device that will be considered as *ground truth*.

Hit or Miss A location provided by Google with a certain accuracy, it is considered to be a *Hit* if the distance between the actual position and the location is less than the accuracy provided by Google. That is to say, that the real location falls inside the circle whose center and radius are the Google location, and the provided Google accuracy. Otherwise, we'll say it is a *Miss*. See figure 4.1

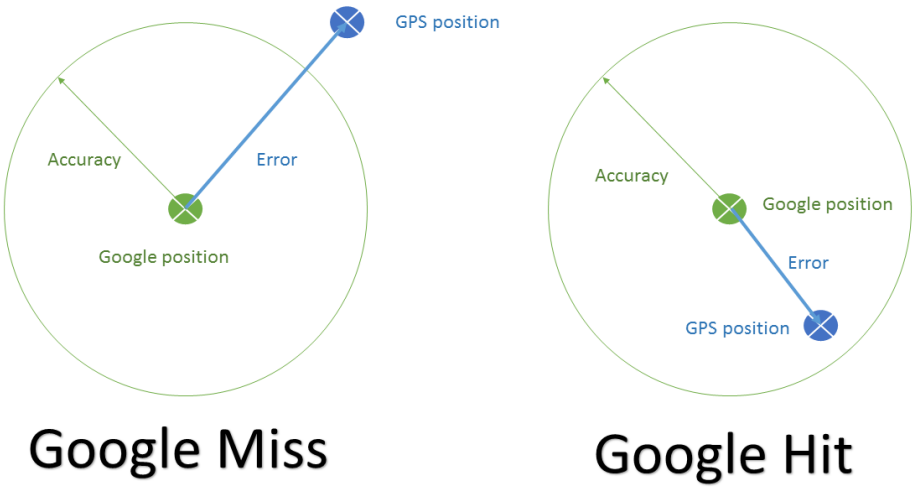


Figure 4.1: Description of what is considered a Google hit/miss. Google gives a radius of accuracy in meters where it is possible to find the device at a given time. Measuring the distance to the *Ground Truth* point (location provided by GPS device) we determine if the device was truly inside the circle. If it is, we call it a *Hit*, otherwise it is a *Miss*.

To obtain this information, the main steps to execute are:

1. Retrieve the position and accuracy provided by Google for two test mobile phones.
2. Comparing these positions retrieved from Google with those considered as ground truth, obtained from GPS devices.
3. Executing experiments under different circumstances. The results are grouped according to classifications of parameters defined, as source of signal, environment, means of transport, weather and traffic density.
4. When location is obtained from 2G or 3G connection, it is likely that Google uses a positioning algorithm based on signal strength and position of connected cell tower. In these experiments, this information will be taken from the mobile phones and with a simple power interpolation a third position will be calculated and compared to real one to obtain this method's error and compare it to Google's.

4.2. Equipment

For the study of the Google Timeline location data accuracy, it is necessary to collect data from different sources. The data to collect come from different devices and a series of experiments have been defined for this task.

4.2.1. Electronic devices

Google Location History timeline data

Google Location History is an application that registers the location of the users' smartphones. The first hypothesis we assume is that Google location performance depends of the mobile phone configuration. In order to study the data collection, two phones **Huawei G6-U10** with Vodafone SIMCARDs are arranged. See figure 4.2. Once the Google accounts are registered and logged in the application and activating location history option, Google automatically starts collecting data. These data can be retrieved at any moment from the Google Maps website. The name of the used accounts are:

location.test2016@gmail.com First Google account.

location.test2016.2@gmail.com Second Google account.

Figure 4.2: Huawei G6-U10. Two mobile phones of this model were used in the experiments.

Ground truth data

The *Ground truth* are the locations registered by an independent and reliable device (handheld GPS). For the experiments not based on the phone GPS capabilities (2G, 3G and WiFi connection), the device used was a GARMIN model **GPS Garmin GPSmap 76Cx**. A picture of this device is in figure 4.3a This model records the location and time in its SD memory card, and can be downloaded later to a computer.
For the experiments based on the mobile phone GPS capabilities, the ground truth has to be a more accurate and precise device. Then the device used was a **uBLOX model EVK-M8**. This device is not able to store the data in its own memory. It is connected to a computer, so the experiments with this receiver were run on a car. An image of this device is shown in figure 4.3b

(a) Garmin GPSmap 76Cx

(b) uBLOX model EVK-M8

Figure 4.3: GPS devices

Personal computer

To collect the information from the devices above described, a laptop with Microsoft Windows is used. This computer is used to:

- Connect to Google's web page to download the file with Google Timeline. This is done after the experiment has finished.
- Copy the locations registered by GPS Garmin. This is done offline too.
- Extract information stored in the mobile phones, after the experiment has taken place.
- Calculate and store the location processing by uBLOX GPS device output, during the experiment.

4.2.2. Transport equipment

Depending of the experiment, different means of transport are used. For still experiments no equipment is needed. Apart from this, bike (personal), tram (public transport) and car (NFI van, as shown in figure 4.4) are used.

Figure 4.4: NFI Van used in car experiments

4.3. Experiment execution

For every experiment the procedure is as described in next steps:

1. **Time synchronization.** Switch on mobile phones and GPS device. Check they are synchronized in time.
2. **Phone Configuration.**Change each mobile phone settings to desired configuration.
3. **Experiment conditions.** Note down conditions on logbook.
4. **Logcat registration.** Start terminal emulators in phones and register logs
5. **End experiment.** Note end time in logbook
6. **Data processing.** Retrieve and gather all data and process.

4.3.1. Time synchronization

At the moment of start up, the phones and GPS devices start synchronizing their clocks to network time or satellite's time. An independent study was carried out forcing the mobile phones to be out of time (10 minutes and an hour). The data obtained from Google's file was with the right time, only the local log files in the phones had the time offset. As normal experiments, positions are compared between Google and GPS device, only the time in this apparatus is relevant, and GPS device can not give positions until they are auto calibrated with satellite's constellation.

4.3.2. Phone Configuration

For each experiment the mobile phones have to be configured to connect only to desired type of network (2G, 3G or WiFi). By default they connect to the most convenient at every moment and this automatic adjustment has to be deactivated. Steps in Android devices for each configuration are described below.

Smartphone only connected in 2G

1. Settings/Privacy/Location/WiFi and Mobile networks
2. Settings/WiFi/Deactivate (From everything, including advanced options)
3. Settings/Mobile Networks/Sim mode/ Only GSM

Smartphone only connected in 3G

1. Settings/Privacy/Location/WiFi and Mobile networks
2. Settings/WiFi/Deactivate (From everything, including advanced options)
3. Settings/Mobile Networks/Sim mode/ Only WCDMA

Smartphone only connected in WiFi

1. Settings/Privacy/Location/WiFi and Mobile networks
2. Settings/WiFi/Activate
3. Press *#*#4636#*#* in your phone. Then press the button "No radio".

Smartphone only connected in GPS

1. Settings/Privacy/Location/Only GPS
2. Settings/WiFi/Deactivate
3. Press *#*#4636#*#* in your phone. Then press the button "No radio".

4.3.3. Experiment conditions in logbook

All the relevant data concerning the experiment has to be registered for later analysis in the logbook. This data include:

Date and time To avoid mistakes, UTC (Coordinated Universal Time) is used for everything. Google files and GPS registers use this convention, so only local files in the phones use local time, that has to be translated to UTC before processing. At the end of the experiment the time has to be written too.

Phone 1/2 configuration This means the kind of signal each device will use. For example, Phone 1 with *2G* and phone 2 with *3G*.

Environment Rural or urban.

Weather Clear, cloudy, rainy.

Traffic Light, normal, busy.

Means of transportation These can be divided into:

- **Still** These experiments were at the NFI, and at home, in Delft.
- **Walking** Most of the experiments were between home and the tram stop.
- **Bike riding** Most of these experiments were between Delft and NFI. See figure 4.14
- **Tram travel** See figure 4.13. Same endpoints as bike, but in public transport
- **Car travel in a rural area** . See figure 4.15. This circuit in the zone of Gouda was run several times with different phone configurations.
- **Car travel in a urban area** See figure 4.16. This circuit in The Hague was run several times with different phone configurations.

Once the experiment finishes, the time is written down in the logbook. Afterwards, at the office, the logs from the phones and GPS are copied to a computer, and the logbook entries are written in the excel sheet. See figure 4.5.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date (dd-MM-yyyy)	Starting Time (UTC)	Finishing time (UTC)	Weather	Enviroment	Traffic	Action	2G	3G	Wi-Fi	GPS	Phone 1	Phone 2
2	08/04/2016	15:15	17:00	Cloudy	Urban	Normal	Tram	False	True	False	True	True	False
3	09/04/2016	0:01	23:59	Clear	Urban	Light	Still	False	True	False	False	True	False
4	10/04/2016	0:01	23:59	Clear	Urban	Light	Still	False	True	False	False	True	False
5	11/04/2016	7:00	8:00	Cloudy	Urban	Normal	Bike	False	True	False	True	True	False
6	11/04/2016	8:00	16:00	Cloudy	Urban	Normal	Still	False	True	False	False	True	False
7	11/04/2016	16:00	17:00	Cloudy	Urban	Normal	Bike	False	True	False	False	True	False
8	12/04/2016	0:01	7:00	Cloudy	Urban	Light	Still	False	True	True	False	True	False
9	12/04/2016	7:00	8:00	Cloudy	Urban	Normal	Bike	False	True	True	True	True	False
10	12/04/2016	8:00	16:30	Cloudy	Urban	Light	Still	False	True	True	False	True	False
11	12/04/2016	16:30	17:15	Cloudy	Urban	Normal	Bike	False	True	True	False	True	False
12	12/04/2016	17:15	23:59	Cloudy	Urban	Light	Still	False	True	True	False	True	False
13	13/04/2016	0:01	7:30	Clear	Urban	Light	Still	False	True	True	False	True	False
14	13/04/2016	7:30	8:20	Clear	Urban	Normal	Bike	False	True	True	False	True	False
15	13/04/2016	8:20	16:45	Clear	Urban	Light	Still	False	True	True	False	True	False
16	13/04/2016	16:45	17:30	Clear	Urban	Normal	Bike	False	False	False	True	True	False
17	13/04/2016	17:30	23:59	Clear	Urban	Light	Still	False	True	True	False	True	False

Figure 4.5: Excel experiment table. The fields registered in it are: Date (dd-MM-yyyy), Starting time (hh:mm:ss), Finishing time (hh:mm:ss), weather (Clear, Cloudy, Raining), Environment (Urban, rural), traffic (Light, Normal, Busy), Action (Still,Walking, Bike, Tram, Train, Car), 2G (True/false), 3G (True/False), Wi-Fi (True/False), GPS (True/False), Phone 1 (True/False), Phone 2 (True/False).

Extracting Data

Extract data from mobile phones To extract information of the phone, first rooted access is required. In this case, it was done with *Kingoroot* software.

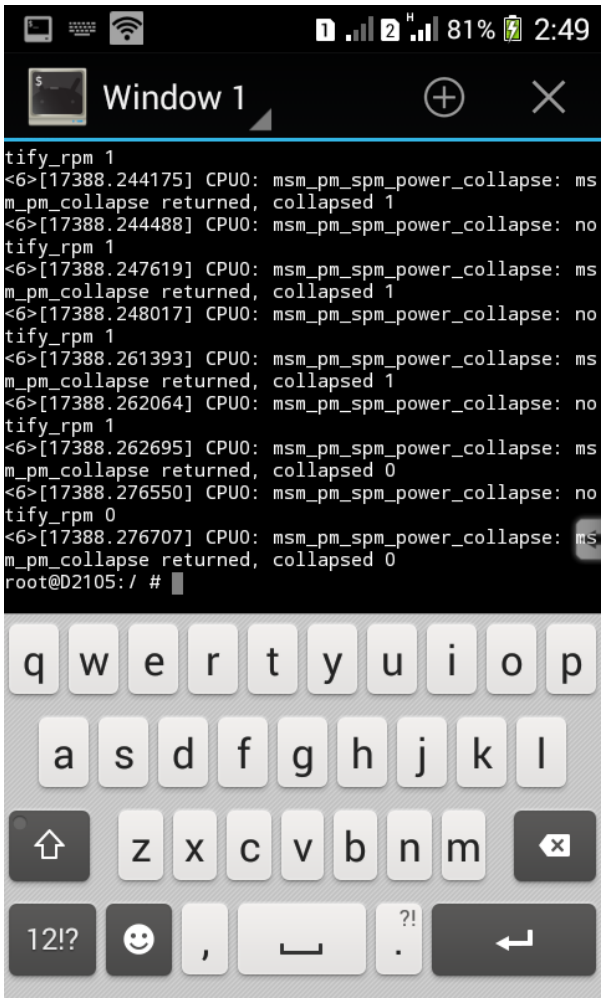


Figure 4.6: Example of Android emulator screen, used as a recompilation of data.

With a terminal in the phone, it is easy to write all the data that is occurring in a text file (see figure 4.6) with the command:

`logcat -v time > LogatFileName.txt` For a general information about the occurrences in the phone.

`logcat -v time - b radio > LogatFileNameRadio.txt` For information about Cell towers and Received Signal Strength of the phone.

It is important to choose specific file names to know which phone they belong to, because after the experiment, the files are copied together to a computer.

Obtain data from Google To obtain the data that Google has stored for each mobile telephone, just enter at webpage *https//www.google.com/maps/timeline*, sign in with the phone account, then select any date in the timeline, and click on the gear to download all the data. See figure 4.7. These raw data is stored in a *.json* file just like the shown in figure 4.8.

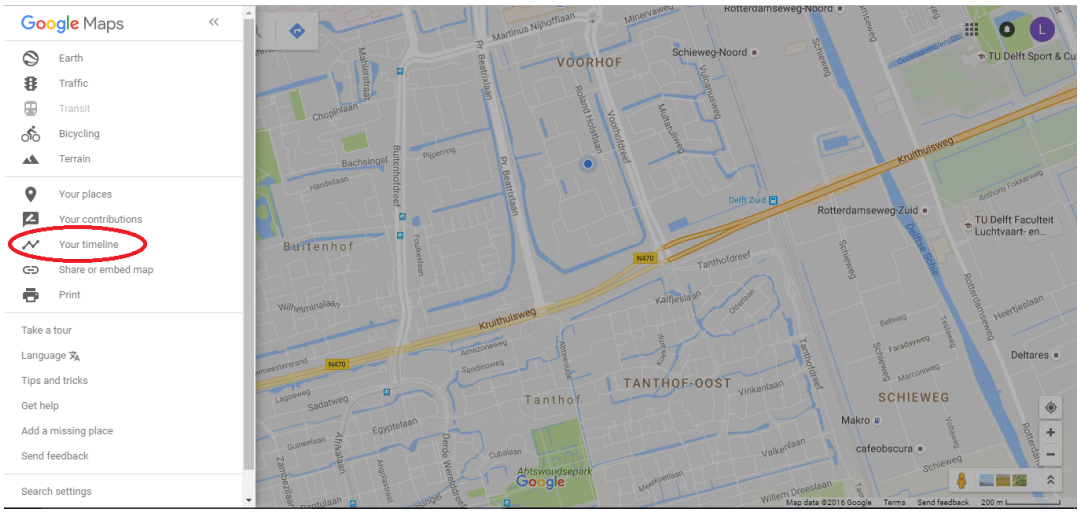


Figure 4.7: Where to find your Google Timeline.For each account, there is *json* file which will provide the data for the accuracy study.


```
  "accuracy" : 22
}, {
  "timestampMs" : "1470476420820",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476387607",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476026837",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476005798",
```


Figure 4.8: Excerpt from the JSON file downloaded form the Google test account. In it, it can be seen timestamp (ms from 1-1-1970), latitude longitude (in degrees x 10^7) and Google accuracy (in meters).

Obtain data from GPS devices The data from GARMIN GPS device is obtained using the free utility *EasyGPS*. The data obtained is in *.gpx* format like shown in figure 4.9.

Figure 4.9: GPX Garmin file. The format is similar to HTML. It can be seen latitudes and longitudes (in degrees), timestamp (yyyy-MM-dd T hh:mm:ss Z in UTC) and elevation (in m).

The data for uBLOX device can not be retrieved offline, so, it has to work connected to a computer. This device is used in car-routes due to the difficulty to use it in other means of transport. The data obtained is in *.ubx* format.

Logbook In a logbook, important data is noted down to afterwards, record it in an excel sheet. See figure 4.10. The information registered is:

Date Current date the experiment takes place

Start time The moment the experiment starts.

Mobile configuration Each mobile is configured to use a unique signal source (2G, 3G, WiFi, GPS)

Environment As the mobile phones use radio signals to communicate and calculate position, the number of cell-towers and obstacles are important parameters to be considered. So, two environment were defined as *rural*, and *urban*.

Weather This circumstance was noted down for each experiment, to check its dependency with the studied variables. Three values were used: *clear*, *cloudy* and *rainy*.

Traffic Also the saturation of traffic was registered to check its influence in the study. Three values were used: *light*, *normal* and *busy*

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date (dd-MM-yyyy)	Starting Time (UTC)	Finishing time (UTC)	Weather	Enviroment	Traffic	Action	2G	3G	Wi-Fi	GPS	Phone 1	Phone 2
2	08/04/2016	15:15	17:00	Cloudy	Urban	Normal	Tram	False	True	False	True	True	False
3	09/04/2016	0:01	23:59	Clear	Urban	Light	Still	False	True	False	False	True	False
4	10/04/2016	0:01	23:59	Clear	Urban	Light	Still	False	True	False	False	True	False
5	11/04/2016	7:00	8:00	Cloudy	Urban	Normal	Bike	False	True	False	True	True	False
6	11/04/2016	8:00	16:00	Cloudy	Urban	Normal	Still	False	True	False	False	True	False
7	11/04/2016	16:00	17:00	Cloudy	Urban	Normal	Bike	False	True	False	False	True	False
8	12/04/2016	0:01	7:00	Cloudy	Urban	Light	Still	False	True	True	False	True	False
9	12/04/2016	7:00	8:00	Cloudy	Urban	Normal	Bike	False	True	True	True	True	False
10	12/04/2016	8:00	16:30	Cloudy	Urban	Light	Still	False	True	True	False	True	False
11	12/04/2016	16:30	17:15	Cloudy	Urban	Normal	Bike	False	True	True	False	True	False
12	12/04/2016	17:15	23:59	Cloudy	Urban	Light	Still	False	True	True	False	True	False
13	13/04/2016	0:01	7:30	Clear	Urban	Light	Still	False	True	True	False	True	False
14	13/04/2016	7:30	8:20	Clear	Urban	Normal	Bike	False	True	True	False	True	False
15	13/04/2016	8:20	16:45	Clear	Urban	Light	Still	False	True	True	False	True	False
16	13/04/2016	16:45	17:30	Clear	Urban	Normal	Bike	False	False	False	True	True	False
17	13/04/2016	17:30	23:59	Clear	Urban	Light	Still	False	True	True	False	True	False

Figure 4.10: Excerpt from Excel experiment table. The fields registered in it are: Date (dd-MM-yyyy), Starting time (hh:mm:ss), Finishing time (hh:mm:ss), weather (Clear, Cloudy, Rainy), Environment (Urban, Rural), traffic (Light, Normal, Busy), Action (Still, Walking, Bike, Tram, Train, Car), 2G (True/False), 3G (True/False), Wi-Fi (True/False), GPS (True/False), Phone 1 (True/False), Phone 2 (True/False). The whole table contains 446 rows.

4.3.4. Logcat registration

With the terminal emulator, two log processes are started in each mobile phone. The result of each log process is called logcat.

- General logcat. This is only for additional information. An important thing in this log is the lines which include the words 'location inserted'. This means that the phone has registered a location in Google Timeline. It is checked that these 'location inserted' in the log correspond in time with the locations retrieved from the *json* file of Google location history. In this work this is not studied but it is interesting to study what happens before or after the locations are inserted, to know what triggers this event.
- Logcat radio. This log contains the identification of cell towers the mobile is connected to, and neighbors (in GSM networks). This information is used to calculate the position in with a third method as explained in this chapter in 4.1

4.3.5. Data processing

After the experiments all available data are collected from:

- GPS GARMIN device, in GPX format.
- GPS uBLOX device, in NMEA format.
- Google Timeline, from the web in JSON format.
- Smartphone logcats (Standard and Radio logcat), in TXT format.
- Logbook from Excel, in XLSX format.

All the data is processed with Matlab in gathered in a table shown in figure 5.6. The way of working to obtain this table is explained in chapter 5 and the appendix A.

4.4. Routes

In these section figures, some routes are shown. In some of them, according to the legend the symbol meaning is

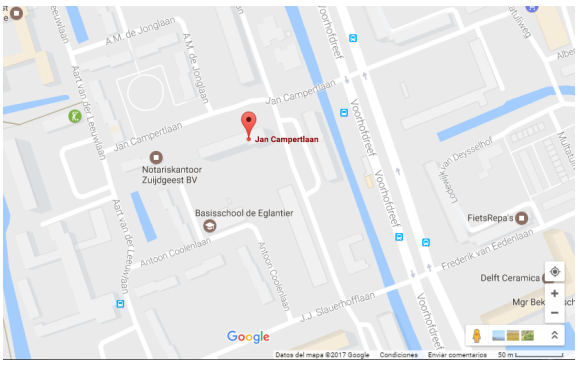
red dots correspond to locations provided by Google.

blue dots correspond to locations provided by GPS device.

magenta triangles correspond to Vodafone telephony towers.

4.4.1. Still

Still experiments were at home (figure 4.11a) and at work place (figure 4.11b).



(a) Still at home

(b) Still at workplace

Figure 4.11: Still experiment places

4.4.2. Walking

The walking experiments were mainly from home to nearest tram stop.

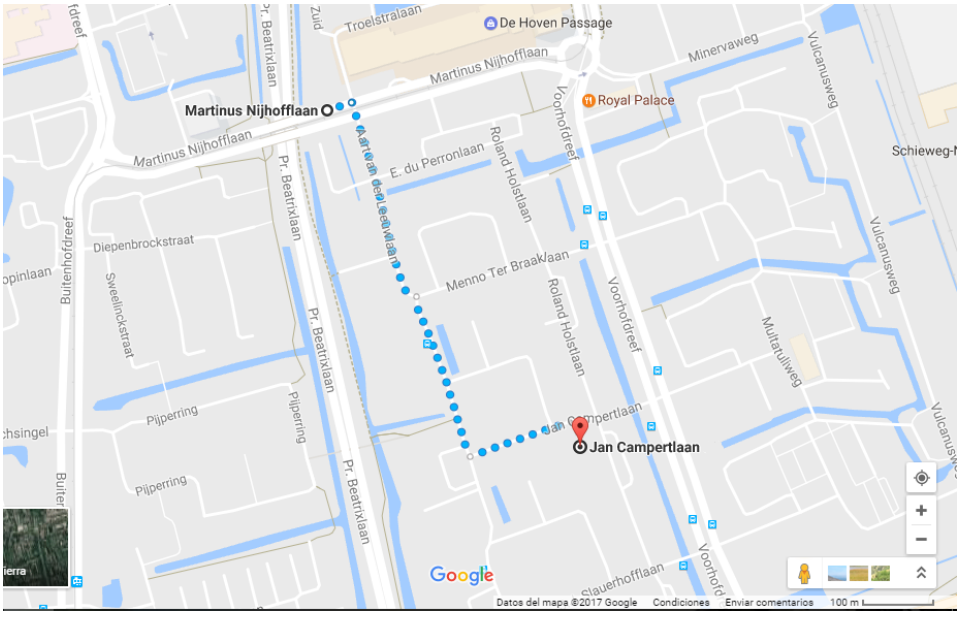


Figure 4.12: Walking experiment, from home to tram stop.

4.4.3. Tram

The routes done in tram are between Martinus Nijhofflaan tram stop in Delft and NFI (Ypenburg).

Figure 4.13: Route taken on October, 12th while traveling on tram

4.4.4. Bike

Most of the routes were between Jan Campertlaan and NFI.

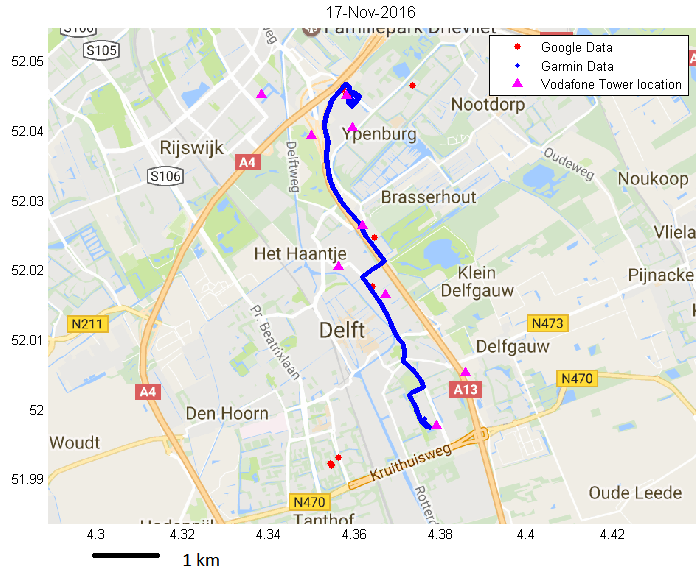


Figure 4.14: Route taken on November, 7th on bike. The red dots indicate locations registered by Google. Blue dots represent the real trajectory, recorded by GPS device.

4.4.5. Car

The rural routes were done in Gouda.

Figure 4.15: Route taken on September 23rd. while traveling by car in a rural environment. We gave 4 returns to the tour showed in the image. the red dots (Google locations) are dispersed around the trajectory

The urban car routes were done in The Hague.

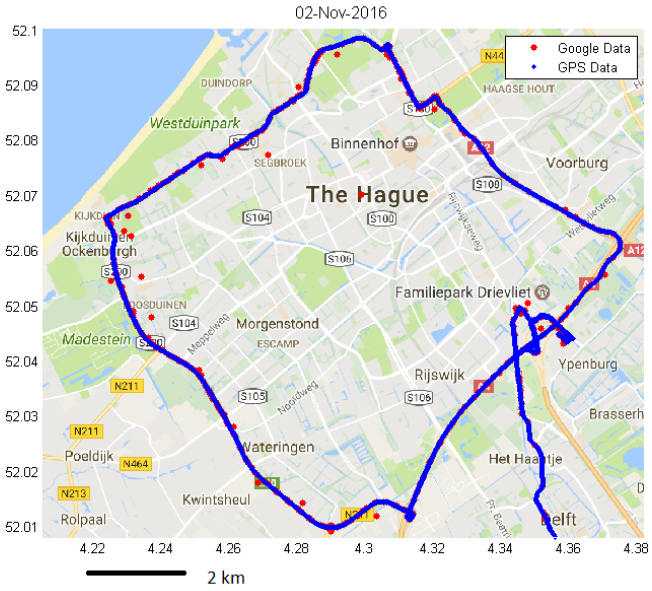


Figure 4.16: Route taken on November, 2nd. while traveling by car in an urban environment. We gave several returns to the tour showed in the image. Most of the red dots (Google locations) are near the trajectory

4.5. Experiment summary

In table 4.1 a summary of invested time in experiments is shown.

Time in Experiments [hh:mm]	Rural	Urban					Total
	Car	Bike	Car	Still	Tram	Walking	
2G	0:28	26:39	4:06	565:41	15:32	1:26	613:52
3G	0:56	32:16	4:06	1230:16	42:36	4:26	1314:36
WIFI	0:58	36:51	2:25	1337:17	52:01	4:02	1433:34
GPS	0:55	9:44	5:10	261:57	18:55	0:10	296:51
Total	3:17	105:30	15:47	3395:11	129:04	10:04	3658:53

Table 4.1: Experiments time summary. Times shown in this table correspond to phone switched on and connected to corresponding network (2G, 3G, WiFi, GPS). For these registered time intervals, logcats were retreived from the phones and Google was able to register locations. It is important to note that the number of observations (Google registered locations) are not proportional to time, so if Still has much more time assigned in experiments, the number of locations registered is not so large compared to the rest of Actions

5

Methodology

For the study of the Google Timeline location data accuracy and Google error, after the experiments have been executed, it is necessary to collect and prepare data, check data, select variables, remove outliers, adjust the model and validate the model.

In each section of this chapter these tasks will be explained.

5.1. Collect data

In order to analyze the behavior of the variables to study, a series of experiments have been defined. During these experiments as much as possible data is collected. This includes the data generated by Google in different configurations of the mobile device (2G and 3G connection, WiFi activated or not...), The "ground truth", that is locations registered by an independent and reliable device (handheld GPS), means of transport, weather, traffic density, recorded manually in a logbook. The experiments classification is described in section 4.3.3.

5.2. Data preparation

For obtaining data, there are five sources corresponding to:

Google Timeline location data Data contained in a *.json* file, one for each mobile phone.

Garmin .gpx data Considered as "Ground Truth".

uBLOX .ubx data Considered as "Ground Truth" when registering locations with GPS activated in the phone.

Vodafone Cell Tower location database

Logcat file containing towers connected to and neighboring towers. (two for each mobile phone)

5.2.1. Google Location History timeline data

Google Location History is an application that registers the location of the users' smartphones. In order to study the data collection, two phones Huawei G6-U10 with Vodafone SIMCARDs are arranged. Once the Google accounts are registered and logged in the application, Google automatically starts collecting data unless told otherwise. These data can be collected at any moment from the Google Maps website. The name of the used accounts were:

location.test2016@gmail.com First Google account.

location.test2016.2@gmail.com Second Google account.

For each account, there is *json* file which will provide the data for the Google accuracy study.

From the timeline webpage, it is possible to download the *Raw data* that Google stores. These raw data is stored in a *.json* file just like the shown in figure 4.8. The locations are expressed as WGS-84 coordinates.

JSON

19458x5 table

	1 TimeStamp	2 lat	3 lon	4 accu	5 date	6
1	7.3643e+05	52.0454	4.3583	53	08-Apr-2016 11:52:19	
2	7.3643e+05	52.0453	4.3583	71	08-Apr-2016 11:52:54	
3	7.3643e+05	52.0454	4.3582	81	08-Apr-2016 11:54:55	
4	7.3643e+05	52.0454	4.3581	87	08-Apr-2016 11:56:56	
5	7.3643e+05	52.0454	4.3582	79	08-Apr-2016 11:58:57	
6	7.3643e+05	52.0454	4.3582	80	08-Apr-2016 12:02:59	
7	7.3643e+05	52.0454	4.3583	75	08-Apr-2016 12:07:03	
8	7.3643e+05	52.0454	4.3583	76	08-Apr-2016 12:18:05	
9	7.3643e+05	52.0454	4.3584	68	08-Apr-2016 12:25:24	
10	7.3643e+05	52.0454	4.3583	76	08-Apr-2016 12:54:18	
11	7.3643e+05	52.0454	4.3584	70	08-Apr-2016 13:16:01	
12	7.3643e+05	52.0454	4.3583	78	08-Apr-2016 13:18:05	
13	7.3643e+05	52.0454	4.3583	76	08-Apr-2016 13:19:25	
14	7.3643e+05	52.0454	4.3582	80	08-Apr-2016 13:21:29	
15	7.3643e+05	52.0454	4.3582	81	08-Apr-2016 13:23:29	
16	7.3643e+05	52.0454	4.3582	81	08-Apr-2016 13:25:12	
17	7.3643e+05	52.0454	4.3583	75	08-Apr-2016 13:26:27	
18	7.3643e+05	52.0455	4.3582	85	08-Apr-2016 13:30:30	
19	7.3643e+05	52.0454	4.3584	73	08-Apr-2016 13:43:33	
20	7.3643e+05	52.0450	4.3664	800	08-Apr-2016 13:48:32	

Figure 5.1: Matlab JSON processed data into table.For each phone, we have a different table. The units are the same as JSON file, except for time, that is expressed in UTC time. The precision internally used is of 7 decimal digits for coordinates.

In Matlab, this is processed into a table just like the one shown in figure 5.1. For that, the timestamps registered in the file have to be converted into UTC (Universal Coordinated Time).

5.2.2. Handheld GPS Data

Within this thesis not only the accuracy radius os Google Timeline data is manipulated, but also it is of interest to know how off Google is with its location determination. In order to compute that, it is necessary to have some "Ground Truth" data. A GPS Garmin GPSmap 76Cx will serve as a comparable data for this purpose.

The data that Garmin provides is given in .gpx file like the one shown in figure 4.9.
For the experiments with mobile GPS capabilities activated, an UBlox device was used and the information is stored in a *.ubx* format.

In both formats, the locations are expressed in WGS-84 coordinates, and time in Zulu-time. The data is processed to Matlab in a table shown in figure 5.2.

JSON GARMIN

3495x5 table

	1 TimeStamp	2 lat	3 lon	4 elev	5 date
1	7.3640e+05	52.0444	4.3589	90.9820	09-Mar-2016 15:38:50
2	7.3640e+05	52.0444	4.3590	90.9820	09-Mar-2016 15:38:52
3	7.3640e+05	52.0444	4.3591	90.5010	09-Mar-2016 15:38:53
4	7.3640e+05	52.0445	4.3593	84.7330	09-Mar-2016 15:38:57
5	7.3640e+05	52.0445	4.3593	79.4460	09-Mar-2016 15:38:59
6	7.3640e+05	52.0445	4.3593	74.1590	09-Mar-2016 15:39:01
7	7.3640e+05	52.0445	4.3593	69.8330	09-Mar-2016 15:39:03
8	7.3640e+05	52.0445	4.3593	66.9490	09-Mar-2016 15:39:04
9	7.3640e+05	52.0445	4.3593	62.6230	09-Mar-2016 15:39:06
10	7.3640e+05	52.0445	4.3593	58.2970	09-Mar-2016 15:39:08
11	7.3640e+05	52.0445	4.3593	53.9710	09-Mar-2016 15:39:10
12	7.3640e+05	52.0445	4.3593	48.6840	09-Mar-2016 15:39:13
13	7.3640e+05	52.0445	4.3593	44.3580	09-Mar-2016 15:39:16
14	7.3640e+05	52.0445	4.3593	39.5510	09-Mar-2016 15:39:19
15	7.3640e+05	52.0445	4.3593	36.6670	09-Mar-2016 15:39:22
16	7.3640e+05	52.0445	4.3593	33.3030	09-Mar-2016 15:39:25
17	7.3640e+05	52.0445	4.3593	29.4580	09-Mar-2016 15:39:28
18	7.3640e+05	52.0445	4.3593	26.0930	09-Mar-2016 15:39:31
19	7.3640e+05	52.0445	4.3593	22.2480	09-Mar-2016 15:39:35
20	7.3640e+05	52.0445	4.3593	19.8440	09-Mar-2016 15:39:39

Figure 5.2: Matlab table for GPX data. The units are the same as the Garmin file. The precision stored and used for coordinates is of 7 decimal digits.

5.2.3. Set of experiments

To establish the experiments, a table has been written in excel. This table states in each Date, time and properties of the experiments, such as transportation mean or weather at the moment. An excerpt of this table is shown in figure 4.5.

5.2.4. Vodafone Cell Tower Database

NFI provided a Vodafone Cell Tower database with the locations of each of the towers in the Netherlands. With this table it will be possible to know the distance from the phone to the towers at any given location.

With this database, it is possible to know the localization of the Vodafone Cell Towers in the Netherlands. The coordinates of the cell towers in the database are in *Rijksdriehoekscoördinaten*, a specific set of coordinates in the Netherlands [61]. This is a cartesian system where the value of the x coordinate runs from west to east, and the y coordinate runs from south to north. The unit is the meter and the central reference point of the system is the spire of Onze Lieve Vrouwetoren ('Lange Jan') in Amersfoort. This central reference point is (155000, 463000) instead of (0, 0). See figure 5.3. With this origin all the points in the European Netherlands have a y coordinate greater than x coordinate and both positives [7].

But this system is not convenient to compare to json and GPS coordinates The system that GPS and Google (in its json files) use is WGS 84 [47] [11]. WGS 84 is an Earth-centered, Earth-fixed terrestrial reference system and geodetic datum (see figure 5.4). WGS 84 is based on a consistent set of constants and model parameters that describe the Earth's size, shape, and gravity and geomagnetic fields. WGS 84 is the standard U.S. Its origin is the mass center of the Earth, the z axis (90 latitude) corresponds to the direction of the BIH Conventional Terrestrial Pole (epoch 1984.0) and the x axis points to the IERS Reference Meridian (zero longitude). This meridian is about 102.5 m east of the Greenwich meridian at the latitude of the Royal Observatory [49].

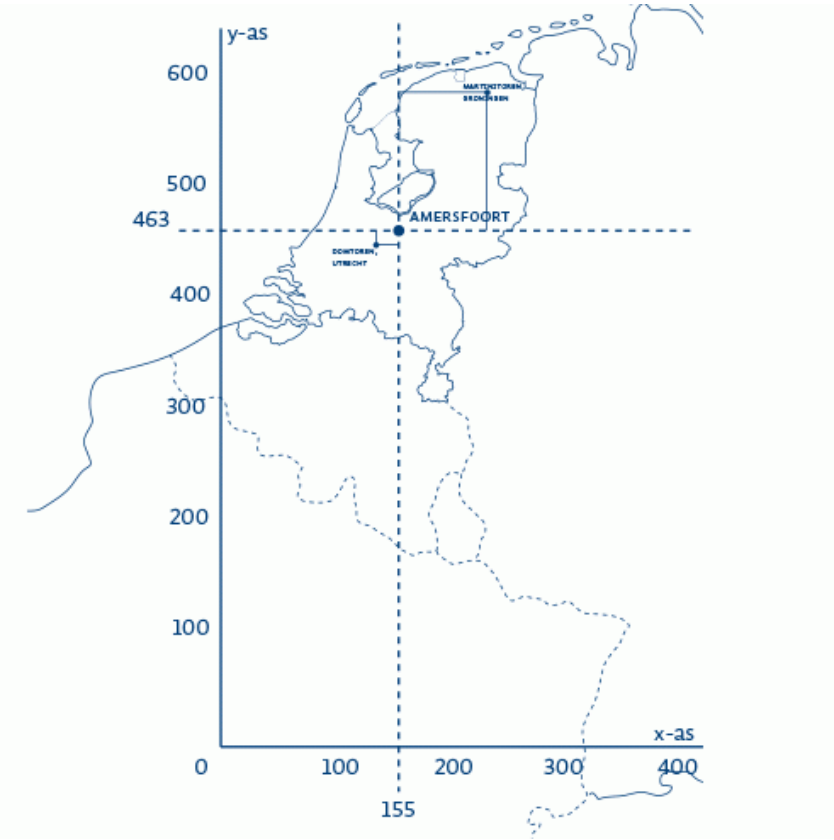


Figure 5.3: Rijksdriehoekscoördinaten: specific coordinate system used in the Netherlands. y coordinate is always greater than x coordinate, and both positives. The reference (155 km, 463 km) is in Amersfoort. Image taken from [7]

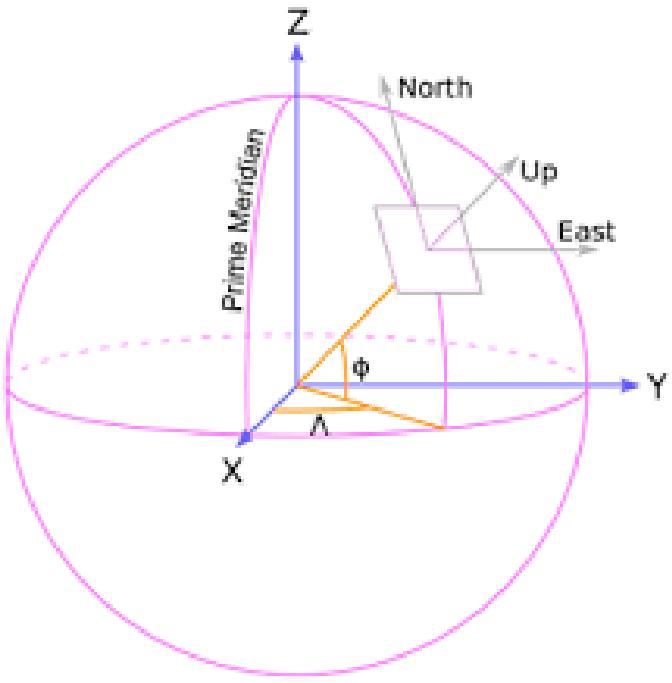


Figure 5.4: Geodesic WGS84 coordinate system. Λ represents longitude, Φ represents latitude. XY is the equator plane and XZ is the plane which contains the Reference Meridian. Image taken from [37]

So, NFI cell tower official database has to be translated from Rijksdriehoekscoördinaten to WGS 84. This

translation is done with a function downloaded from Mathworks®called rd2wgs [62]. It is not just a coordinate conversion but also a datum transformation. The accuracy from this transformation is better than half a meter [61]. This precision is more that enough because the positioning error in this thesis is always greater than 5 m.

5.2.5. Logcat File

To extract information of the phone, first rooted access is required. In this case, it was done with *Kingoroot*. With a terminal in the phone, it is easy to write all the data that is occurring in a text file (see figure 4.6) with the command:

`logcat -v time > LogatFileName.txt` For a general information about the events in the phone.

`logcat -v time - b radio > LogatFileNameRadio.txt` For information about Cell towers and Received Signal Strength of the phone.

The Logcat radio files contain information about the Cell ID towers which it is connected to and the signal power received from each tower. In 2G networks, the information available is for all Cell IDs and signal powers. In 3G, only signal power is available for all connected base stations, and only the main cell Id is fully identified.

These events are also marked with local date and time. It has to be converted to UTC in order to putting into accordance with the rest of the data. The format in which Android writes information when it is 2G or 3G is different. Because of that, there were two programs written to read this information from 2G and 3G which also read the two telephones independently. The time registered in the phone is local time. To convert this time into UTC, a function was written in Matlab. This function converts Amsterdam Time to UTC subtracting 1 hour. As DST (Daylight Saving Time) is active, 2 hours are subtracted in the summer period, that is between last Sunday of March, 2AM until last Sunday of October, 3AM.

5.2.6. Gathering all the data

With the data of all the former files, a table with all the synchronized data is generated. This table is shown in figure 5.6. As a base, *json* files are taken. The data from both phones are distinguished because the field *phone* takes the value 1 if it is *location.test2016@gmail.com* and value 2 if the account is *location.test2016.2@gmail.com*. After that, "Ground truth" information is added (Garmin or uBLOX). To add this information, we use the *timestamp* information as a basis. It is impossible that time stamp of *json* file and *gpx* file coincide, the nearest one is looked for.
For each date in *json* table, the nearest *gpx* date is searched. If it is inside a margin (7 seconds for our case), the *gpx* data are assigned to the synchronized table (see again figure 5.6).
At the same time we add the "Ground truth" latitude and longitude, the distance to the Google Timeline position is computed and saved in the table. It is called *Google Error*. It is also considered error in x (meters E-W) and error in y (meters N-S).

Google error calculation

As the distance between Google and actual locations are much smaller than the Earth radius, we can consider that these two points are in the diagonal of a rectangle on the Earth's surface. The length of the sides of this rectangle are the error in x-axis and y-axis. See figure 5.5.

$$\begin{aligned} e_x &= R\cos\frac{\theta_1+\theta_2}{2}(\phi_1-\phi_2) \\ e_y &= R(\theta_1-\theta_2) \\ e &= \sqrt{e_x^2+e_y^2} = R\sqrt{\cos^2\bar{\theta}\cdot(\Delta\phi)^2+(\Delta\theta)^2} \end{aligned} \tag{5.1}$$

Being:

R: Earth radius

θ_1, θ_2**:** latitudes of Google and actual locations (expressed in radians)

ϕ_1, ϕ_2**:** longitudes of Google and actual locations (expressed in radians)

e_x is measured over the parallel whose radius is $R\cos\overline{\theta}$
e_y is measured over the meridian whose radius is R

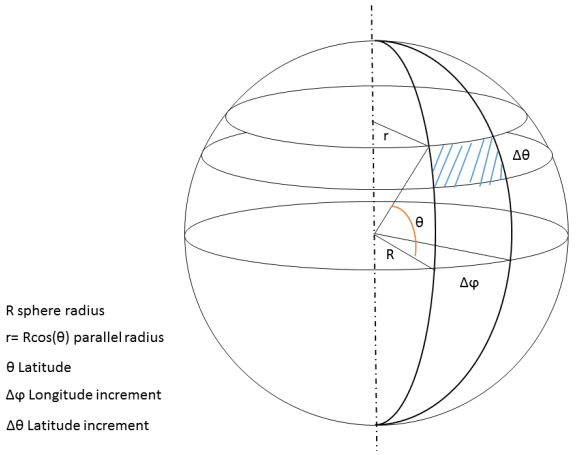


Figure 5.5: Distance on sphere. When the distances on a sphere are small compared to the radius, it can be calculated as the diagonal of a "rectangle" defined between two parallels and two meridians.

Merging Signal strengths and experiment conditions

With the *Logcat* files info, it is equally proceeded. the information added is the number of towers, the three strongest signal powers.
Finally, the information of the *Experiment table* is added. As each experiment has a starting and ending date, to all the registers of the synchronized table that are comprised between two timestamps are assigned the characteristics of the point experiment (source of signal, weather, traffic, environment, action).

1 TimeStamp	2 lat	3 lon	4 accu	5 Phone	6 Glat	7 Glon	8 err_xy	9 err_x	10 err_y	11 G2	12 G3	13 WIFI	14 GPS	15 Weather	16 Traffic	17 Environment	18 Action	19 SOURCE
7.3663e+05	52.0397	4.3139	778	2	52.0081	4.3527	NaN	NaN	NaN	0	1	1	0	Clear	Normal	Urban	Still	G3
7.3663e+05	52.0454	4.3580	79	1	52.0444	4.3598	160.3208	119.1322	107.2874	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3580	79	2	52.0444	4.3598	161.0472	120.2991	107.0731	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3581	19	1	52.0444	4.3598	161.9024	117.4327	111.4552	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3582	19	1	52.0444	4.3598	153.9548	108.8047	108.9213	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0452	4.3583	20	2	52.0444	4.3598	137.7492	99.8912	94.8513	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0449	4.3586	20	2	52.0444	4.3598	100.9037	82.9279	57.4857	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0450	4.3586	50	1	52.0444	4.3598	106.0738	82.2178	67.0221	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0467	4.3734	830	1	52.0448	4.3604	917.7567	894.7289	204.3778	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0449	4.3603	919.3845	898.3221	195.7473	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0449	4.3586	20	2	52.0449	4.3604	121.0499	120.9514	4.8847	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0467	4.3734	830	1	52.0335	4.3461	2.3776e+03	1.8774e+03	1.4592e+03	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0141	4.3513	3.9264e+03	1.5215e+03	3.6199e+03	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0024	4.3540	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	51.9932	4.3588	47	1	51.9932	4.3593	34.0335	33.9321	2.6259	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	51.9933	4.3589	4354	1	51.9919	4.3541	360.4386	327.8190	149.8462	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9917	4.3549	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9921	4.3549	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9922	4.3546	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	NaN	NaN	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	NaN	NaN	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	51.9920	4.3546	21	2	NaN	NaN	NaN	NaN	NaN	0	1	1	0	Clear	Normal	Urban	Still	WIFI

Figure 5.6: Crossed experiment table with all the data form the experiments, including Garmin-UBX, Logcat and Google. First column is the TimeStamp y Matlab format.
Columns 2 to 4 are Google provided coordinates (in degrees) and accuracy (in m).
Column 5 defines which mobile device the entire row refers to.
Columns 6 and 7 are the coordinates provided by ground truth device (in degrees).
Columns 8 to 10 are show the distance between ground truth and Google position, total and in x, y axes (expressed in m).
Columns 11 to 14 are booleans which indicate which network were active in the mobile device (2G, 3G, Wifi , GPS).
Columns 15 to 18 are the experiments circumstances.
Column 19: type of network used for classification.

5.3. Data Check

At this step, with all available data gathered, some simple sights and checks can be done.

5.3.1. Data Visualization

Available data has been organized on Signal Source, Environment and Action.

Tables 6.1 and 6.2 in chapter 6 give an idea of accuracies and error values, means and dispersion.

In tables 6.3, 6.4, 6.5 and 6.6 a more detailed study has been done on Accuracy variable. The data has been classified in the same way, but the data shown is the number of observations, minimum, maximum, 50, 68 and 95 percentiles, and root mean square.

5.3.2. 2G Location by power interpolation

A simple method of interpolation was programmed for doing a light test of data consistency. Taking data from 2G logcats, which include full tower identification and signal strength, and the official Vodafone tower database, the locations have been calculated by simple interpolation. For each available register, with three or more identified connected towers, the location is calculated with the tower coordinates weighted with the signal strengths.

$$\overline{x} = \frac{\sum (ss_i \cdot x_i)}{\sum ss_i} \tag{5.2}$$

Being

x_i latitude and longitude of i nearest cell tower.

ss_i signal strength received from i nearest cell tower, expressed in ASU (Arbitrary Strength Unit) [44].

$\overline{x}$ interpolated position (latitude and longitude)

Results are in table 6.7 and in figure 6.16

5.4. Defining the model

The data seen at this moment is only sorted, analyzed and classified. Next task is to look for a model (or several) to be able to predict the values of the variables of interest (Google provided Accuracy and Google error) as a function of other variables.

For this task (choosing the model) an interface has been prepared to ask the user which kind of data to take to analyze. The user can filter data, select the response variable and the regressor. Then one or several models are generated and can be improved. This software is explained with more detail in Chapter B.

5.4.1. Variables chosen

The chosen model to study the experiments is a multi linear regression model (least squares model) explained in chapter 3.

To apply this model, first it has to be decided which variables influence the response (google accuracy or google error). We will use the methods discussed in section 3.2.2.

We will follow two paths to study the subsets of variables that may influence the radius of accuracy of Google.

Subset A This subset will have the three strongest received signal powers registered in the Logcat radio file (2G and 3G networks) and also the categorical values of the excel table from the experiment (weather, traffic, etc.).

Subset B Based on the Vodafone Cell Tower data base, for each Google point the three nearest towers are searched. With those towers the distance and angles to the device are computed just as shown in figure 5.9. Experiment variables are also added (weather, traffic, etc.). The distances are calculated as explained in (5.1) and the angles using the dot product between vectors as described in equations (5.3) and figure 5.7.

$$\alpha_1 = \arccos \frac{\mathbf{r_2}^T \mathbf{r_3}}{|\mathbf{r_2}||\mathbf{r_3}|}$$

$$\alpha_2 = \arccos \frac{\mathbf{r_3}^T \mathbf{r_1}}{|\mathbf{r_3}||\mathbf{r_1}|}$$

$$\alpha_3 = \arccos \frac{\mathbf{r_1}^T \mathbf{r_2}}{|\mathbf{r_1}||\mathbf{r_2}|} \tag{5.3}$$

$$with$$

$$\mathbf{r_i} = \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \end{pmatrix} = \begin{pmatrix} r_{ix} \\ r_{iy} \end{pmatrix}$$

$$|\mathbf{r_i}| = \sqrt{r_{ix}^2 + r_{iy}^2}$$

Being:
(x_0, y_0) the location coordinates expressed in meters (not latitude and longitude)
(x_i, y_i) the i^{th} nearest tower location, expressed in meters.

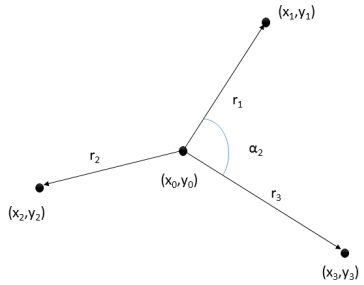


Figure 5.7: Angles between device and towers. The position of the phone is (x_0, y_0) and the three nearest towers (x_1, y_1), (x_2, y_2) and (x_3, y_3). r_1, r_2 and r_3 are the position vectors of the towers respect to the phone location. The angles are calculated using the equations (5.3)

5.4.2. Training models

We tried models with the two sets of variables and see if they provided good predicting results.

Subset A model

The subset of variables that was chosen is shown in figure 5.8.

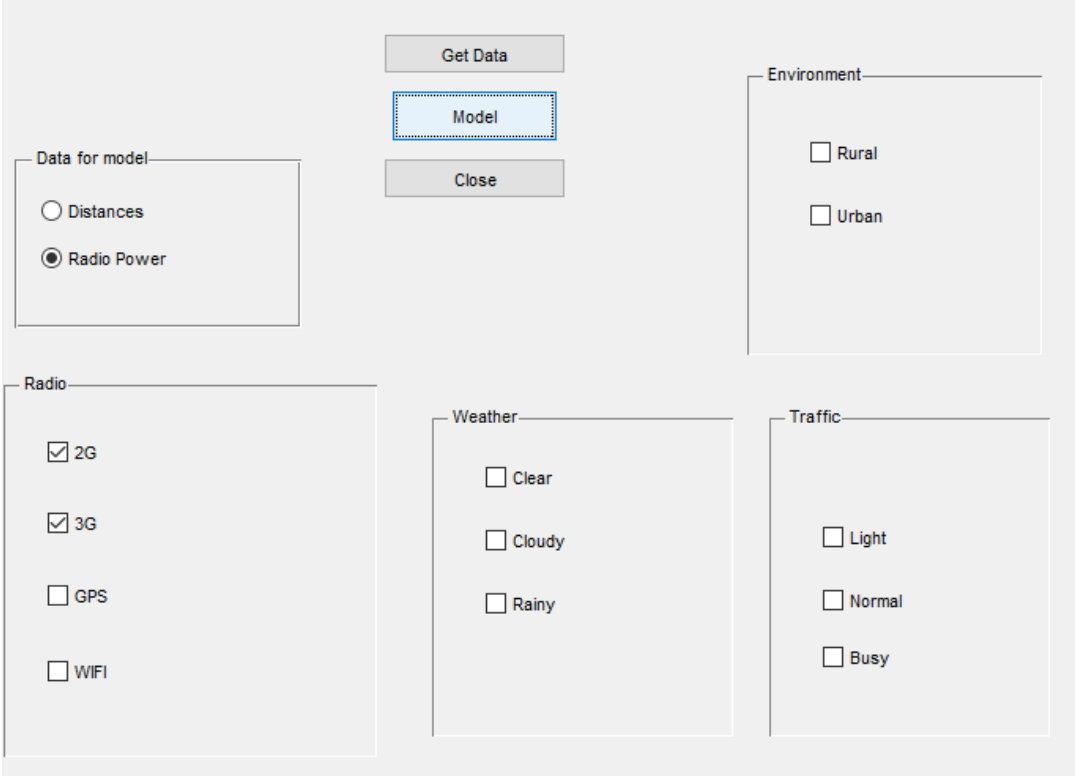


Figure 5.8: Data selection for trial Model A. Includes all the 2G and 3G events plus the Received signal strengths as regressors

For this model the Wilkinson notation is (refer to section 3.1.6):

$$Gaccuracy \sim Source * (dBm1 + dBm2 + dBm3) \tag{5.4}$$

Where dBmX are the received signal powers converted to decibel milliwatt. The results are shown in table 5.1.

The columns correspond to the values of [48]

Estimate The estimated value for the coefficient.

SE The estimated standard error for the coefficient. $SE = \sqrt{\hat{\sigma}^2 C_{jj}}$

tStat The ratio of the Estimated coefficient and its standard error. It is a statistic to evaluate the significance of the coefficient for the model. The t-value measures the size of the difference relative to the variation in your sample data. Put another way, T is simply the calculated difference represented in units of standard error. The greater the magnitude of T (it can be either positive or negative), the greater the evidence against the null hypothesis that there is no significant difference. The closer T is to 0, the more likely there isn't a significant difference.

pValue The P-value is the smallest level of significance that would lead to rejection of the null hypothesis H_0 with the given data. In this case H_0 is that the coefficient has not significance for the model. When the p-value is very low ($< \alpha = 0.05$), we reject the null hypothesis and conclude that there's a statistically significant difference.

	Estimate	SE	tStat	pValue
(Intercept)	2120.3911	650.0499	3.2619	1.5940e-03
dBm1	13.1938	14.2463	0.9261	3.5700e-01
dBm2	16.3894	19.4546	0.8424	4.0191e-01
dBm3	-12.7856	23.3478	-0.5476	5.8539e-01
SOURCE_G3	-1409.6345	906.6788	-1.5547	1.2373e-01
dBm1:SOURCE_G3	-11.3566	22.5347	-0.5040	6.1559e-01
dBm2:SOURCE_G3	-15.4737	27.7373	-0.5579	5.7840e-01
dBm3:SOURCE_G3	8.8148	31.1761	0.2827	7.7806e-01

Table 5.1: Subset A coefficients. To reject the null hypothesis of $\beta_i \neq 0$, p-Value should be smaller than 0.05. In this case, none of the coefficients meets this requirement This indicates the set of variables chosen is not adequate. The last two columns have dimensionless values. First and second column have the same units as the dependent variable ([m] for accuracy and error) divided by the units of the regressor. This way, Intercept and SOURCE are dimensionless, dBmx are expressed in [dBm] , the interactions (dBm1:SOURCE_G3, for instance) are expressed in the product of units, in this case [dBm]

As explained in 3.1.6 the meaning of equation (5.4) is that Google's accuracy is calculated as a linear combination of the three variables dBm1, dBm2 and dBm3, and all the products (interactions) with Source: Source*dBm1, Source*dBm2 and Source*dBm3. The linear combination may have a constant added (intercept). As *Source* is a categorical variable, in this case with two categories (2G and 3G), one category is considered as *base* and its influence is reflected in the intercept, and the other category appears as a possible factor (SOURCE_G3). For each categorical variable, the number of factors is equal to the number of categories minus 1.

The meaning of the columns in tables 5.1 and 5.2 are explained at the beginning of this section. *Estimate* is the value of the coefficient we are looking for and the other columns indicate how good or reliable this value is. For example, *pValue* indicates the probability that the estimated coefficient really doesn't influence the response variable *Accuracy*. So values below 0.05 are considered as good sign that the coefficient may be good, at least it is not zero.

Seeing the poor results of the Subset A, a stepwise method is tried in order to contrast them. The result is shown in table 5.2. Stepwise is a method in which combined variables and cross products between them are introduced and tested in an iterative manner. For this model, Stepwise method only retained 2G average (intercept) and 3G average (intercept + Source_G3).

	Estimate	SE	tStat	pValue
(Intercept)	1107.5623	37.0175	29.9200	1.5940e-145
SOURCE_G3	-298.9514	43.0473	-6.9447	6.3686e-12

Table 5.2: Stepwise model coefficients. Stepwise does not include any of the received signal strength powers, which means that these ones are not significant for the model. Signal strength may give information about the location, but not about the precision, because in the same place, the signal strength can vary depending on multiple factors, like moving objects and people, or network circumstances.

Subset B

To set a simple scenario, the usual way to locate a phone is to use the signal strength registered in it, and thus the distance to the Cell Tower connected. For this model, it is assumed that the closest distances to the nearest three cell towers influence the accuracy of Google, and also the angles that the device forms with the three cell towers, seen as figure 5.9.

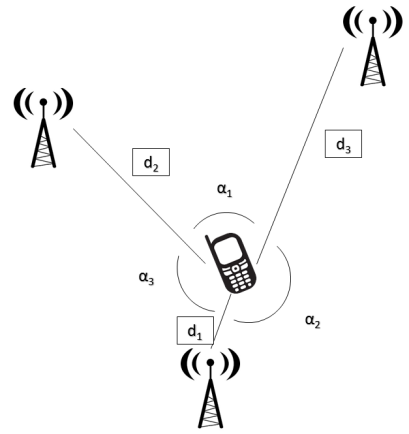


Figure 5.9: Cell Tower distance diagram. This is a diagram about how the variables in the model are going to be treated. d_1, d_2 and d_3 are the distances from the smartphone to the closest cell towers in ascending order. α_1, α_2 and α_3 are the angles opposing to their respective distances.

But Google doesn't only use cell tower location, but also uses both Wi-Fi and GPS location. When either Wi-Fi or GPS is set ON in the location device, it can be seen that the Google accuracy improves notably. To consider that in the model, two dummy variables are inserted, WiFi (with values "True", "False") and GPS (with values "True", "False"). Also, when the data is activated for the phone, there are two modes: 2G and 3G. To study the variability in the accuracy (and check if it influences) there is another dummy variable added that is called Source, that is a compilation of all the former variables. It classifies the accuracy of a unique observation and determines the source of it, "2G", "3G", "WiFi" and "GPS". With that, and adjusted model can be done for each of the four cases.

Subset B with no categoricals For the first try, we are going to consider a simple model in which Google only takes information of the Cell Towers the smartphone has been connected to. Then the parameters are the three distances to said Cell Towers and their respective angles, just as seen in figure 5.9. The model will be in Wilkinson's notation:

$$Gaccuracy \sim (d_1 + d_2 + d_3 + \alpha_1 + \alpha_2 + \alpha_3) \tag{5.5}$$

The resulting table contains seven coefficients (6 estimators plus the intercept).

Accuracy 2G	Estimate	SE	tStat	pValue
(Intercept)	386.8889	34.2313	11.3022	3.5437e-29
d1	0.6753	0.0461	14.6600	1.9616e-47
d2	-0.3949	0.0772	-5.1146	3.2916e-07
d3	0.1512	0.0510	2.9648	3.0472e-03
a1	119.2389	12.5533	9.4986	3.5592e-21
a2	-155.3732	12.4523	-12.4775	4.4497e-35
a3	-66.3875	7.1927	-9.2298	4.2916e-20

Table 5.3: Coefficient table of the first model. The first column contains the names of the estimators. The second contains its values, the third contains the total sum square, the fourth the t-Student statistic and the fifth the p-Value. p-Values below 0.05 indicate that the estimation coefficient can be significant for explaining the response variable *Accuracy*

Taking a look at table 5.3 every p-Value is below 0.05 for every case. For this significance level (95%) the t-Student value is $t_{0.025,1140-7} = 1.9621$. As every element in the fourth column (tStat) has absolute value greater than 196, we reject the null hypothesis that any estimator becomes not significant to the model.

On the contrary, the $R^2 = 0.1384$ and $\tilde{R}^2 = 0.1285$ which indicates this model doesn't approach reality very well. This is due to the lack of consideration in if the 2G, 3G Wi-Fi or GPS are activated, which has a great impact in accuracy.

Subset B Second model For a first analysis of the model, we only considered the variables mentioned in subsection 5.4.1. The first model considered is (see section 3.1.6 for interpretation):

$$Gaccuracy \sim (d_1 + d_2 + d_3 + \alpha_1 + \alpha_2 + \alpha_3) * SOURCE \tag{5.6}$$

Matlab calculus software generates a design matrix with the following columns:

1 Intercept Independent term.

6 Independent variables Six columns that include d_1, d_2, d_3 and $\alpha_1, \alpha_2, \alpha_3$.

2 Dummy variables As *Source* has three possible values (2G, 3G and WiFi), two dummy variables are added. These two variables are called *Source_3G* and *Source_WIFI*.

12 Cross products Variable *Source* has to be combined with all the other independent ones in order to take into account the intervention of WiFi, 2G and 3G in the slopes and not only the intercept.

The results obtained for this model in shown in table 5.4.

Accuracy	Estimate	SE	tStat	pValue
(Intercept)	121.9070	191.4825	0.6366	5.2448e-01
d1	0.1467	0.2222	0.6600	5.0938e-01
d2	0.4858	0.3008	1.6149	1.0662e-01
d3	-0.1022	0.1824	-0.5604	5.7535e-01
a1	260.7371	51.4531	5.0675	4.7178e-07
a2	33.6177	59.7007	0.5631	5.7348e-01
a3	32.6972	59.2695	0.5517	5.8129e-01
SOURCE_G3	838.3528	205.7028	4.0756	4.9169e-05
SOURCE_WIFI	0.0000	0.0000	-	-
d1:SOURCE_G3	-0.2654	0.2440	-1.0879	2.7687e-01
d1:SOURCE_WIFI	0.0000	0.0000	-	-
d2:SOURCE_G3	-0.1914	0.3424	-0.5590	5.7626e-01
d2:SOURCE_WIFI	0.0000	0.0000	-	-
d3:SOURCE_G3	0.0333	0.2087	0.1595	8.7329e-01
d3:SOURCE_WIFI	0.0000	0.0000	-	-
a1:SOURCE_G3	-266.0998	59.0556	-4.5059	7.3068e-06
a1:SOURCE_WIFI	0.0000	0.0000	-	-
a2:SOURCE_G3	-41.9665	64.9158	-0.6465	5.1810e-01
a2:SOURCE_WIFI	0.0000	0.0000	-	-
a3:SOURCE_G3	-163.4336	63.4370	-2.5763	1.0114e-02
a3:SOURCE_WIFI	0.0000	0.0000	-	-

Table 5.4: Second model coefficients. The Wilkinson notation for this model is $Source * (d_1 + d_2 + d_3 + a_1 + a_2 + a_3)$. The coefficients where *WIFI* appears do not have values because there were no observations in the data used to generate the model

In total, with this model we count with 21 estimators. This time, there are much more parameters that are worse. (For example just Wi-Fi). These parameters should be removed from the model if $\tilde{R}^2$ is increased. The good news is that $\tilde{R} = 0.6119$ has significantly improved, assuring now that the model adjusts much better considering *Source* as a categorical variable.

5.4.3. Other models

We are going to consider now more variables that can affect Google accuracy. We saw that Cell tower distance has a big deal when neither Wi-Fi or GPS is activated, so the first thing that comes to mind is to differentiate between rural areas, where the Cell tower density by km^2 is quite low, vs urban environment, where the density is greater.

The results are shown in table 5.5. The $R^2 = 0.6499$ and $\tilde{R}^2 = 0.6474$. This indicates that environment has a slightly increase in the adjusted R, which indicates that the new variable helps to make the model more precise.

Accuracy	Estimate	SE	tStat	pValue
(Intercept)	329.4402	274.6937	1.1993	2.3048e-01
d2	0.6041	0.2341	2.5804	9.9034e-03
d3	-0.0294	0.1501	-0.1956	8.4490e-01
a1	4.6355	35.2691	0.1314	8.9544e-01
a2	45.3278	44.4359	1.0201	3.0776e-01
a3	277.2828	112.7752	2.4587	1.3986e-02
Environment_Urban	-284.0369	205.0604	-1.3851	1.6609e-01
SOURCE_G3	1075.9354	268.4769	4.0076	6.2481e-05
SOURCE_WIFI	-420.3483	277.2663	-1.5160	1.2959e-01
d2:a1	-0.2444	0.0649	-3.7650	1.6899e-04
d2:a3	0.2703	0.0607	4.4497	8.8356e-06
d2:Environment_Urban	0.2362	0.0556	4.2459	2.2271e-05
d2:SOURCE_G3	-0.3542	0.1798	-1.9698	4.8936e-02
d2:SOURCE_WIFI	-0.6639	0.1716	-3.8697	1.1072e-04
d3:a1	0.1643	0.0504	3.2605	1.1217e-03
d3:a3	-0.3805	0.0518	-7.3437	2.5093e-13
d3:SOURCE_G3	-0.0575	0.1326	-0.4340	6.6428e-01
d3:SOURCE_WIFI	0.3082	0.1313	2.3469	1.8982e-02
a1:a3	183.0294	13.6269	13.4315	2.9778e-40
a1:SOURCE_G3	-236.2846	31.4198	-7.5202	6.7270e-14
a1:SOURCE_WIFI	-222.2322	31.2120	-7.1201	1.2752e-12
a1:SOURCE_GPS	0.0000	0.0000	-	-
a2:Environment_Urban	151.2227	37.9951	3.9801	7.0135e-05
a2:SOURCE_G3	-89.4121	33.9027	-2.6373	8.3893e-03
a2:SOURCE_WIFI	-13.9581	34.2109	-0.4080	6.8330e-01
a3:Environment_Urban	-378.9389	86.0963	-4.4013	1.1043e-05
a3:SOURCE_G3	-154.0993	32.3623	-4.7617	1.9885e-06
a3:SOURCE_WIFI	-76.2988	31.3156	-2.4364	1.4876e-02
Environment_Urban:SOURCE_G3	-158.2537	188.6300	-0.8390	4.0154e-01
Environment_Urban:SOURCE_WIFI	285.6247	202.4664	1.4107	1.5840e-01
Environment_Urban:SOURCE_GPS	0.0000	0.0000	-	-

Table 5.5: Table with 2G, 3G Wi-Fi and environment consideration.

Due to the model not adjusting well to the whole data collection, it is decided to separate the data by *Source* variable (2G, 3G, WiFi and GPS). So in total we developed 8 models: *Accuracy* (Google provided accuracy) for each of the 4 modes and *Error* (Google distance error) for the 4 modes.

5.4.4. Variable selection for simple linear model

To see which variables influence the model, a program is made in a way that all the possible linear models that use any parameter number are calculated. Being k the number of available regressors, there are $\binom{k}{1}$ that use 1 parameter, $\binom{k}{2}$ that uses 2 parameters and $\binom{k}{k}$ combinations that use k parameters. Remember that combinations are defined as:

$$\binom{k}{i} = \frac{k(k-1)\cdots(k-i+1)}{i(i-1)\cdots 1} = \frac{k!}{i!(k-i)!} \tag{5.7}$$

All the models are computed and the best values for R^2, $\tilde{R}^2$, C_p and $S = \sqrt{MSE}$ are saved in table. From this table the criterion for the best $\tilde{R}^2$ for each 2 best models from each group are searched. The results can be seen in table 5.6: Observing this table we can choose the best fit looking at the same time at R^2, $\tilde{R}^2$, C_p and S.

	R^2	$\tilde{R}^2$	C_p	S
1 + Environment	0.1236	0.1206	80.0130	541.4664
1 + d3	0.0837	0.0806	96.9928	553.6532
1 + d2 + a1	0.2460	0.2409	29.8996	503.0735
1 + Action + Environment	0.2398	0.2294	40.5305	506.8585
1 + d2 + a1 + Action	0.3083	0.2964	13.4062	484.3369
1 + d3 + a1 + Action	0.3034	0.2915	15.4667	486.0288
1 + d2 + a1 + a3 + Action	0.3143	0.3001	12.8308	483.0444
1 + d2 + a1 + Action + Environment	0.3138	0.2996	13.0461	483.2226
1 + d2 + a1 + a2 + a3 + Action	0.3205	0.3040	12.2200	481.7101
1 + d2 + a1 + a3 + Action + Environment	0.3191	0.3027	12.7782	482.1747
1 + d2 + a1 + a2 + a3 + Action + Environment	0.3264	0.3077	11.7057	480.4437
1 + d1 + d2 + a1 + a2 + a3 + Action	0.3230	0.3042	13.1328	481.6379
1 + d1 + d2 + a1 + a2 + a3 + Action + Environment	0.3280	0.3069	13.0159	480.7006
1 + d2 + d3 + a1 + a2 + a3 + Action + Environment	0.3264	0.3052	13.7050	481.2794

Table 5.6: Accuracy Variable selection

As the number of regressors shown in table 5.6 is not constant, and the value of $\tilde{R^2}$ is preferable to R^2 value. The value of C_p must be low and close to the regressor number. It has to be taken into account that categorical regressors, such as *Action* , count as several regressors. Each one counts as the number of categories (possible values) minus 1.
The value of $\tilde{R^2}$ quantifies how this models improves with respect to the model of constant Y (mean). The improvement is measured with the ratio of the standard deviation of the residuals in the study model with respect to the standard deviation of the residuals of the model with just intercept (constant model). The closer to one, the smaller the residuals are and the better the model is.

Once the variables are chosen the next step is to improve the model itself.

5.5. Refining the model

5.5.1. Global F test

As seen in section 3.2.2 F, we will use F so assess the fit of the model.
In general, an F-test in regression compares the fits of different linear models. Unlike t-tests that can assess only one regression coefficient at a time, the F-test can assess multiple coefficients simultaneously, all in one go.

The F-test of the overall significance is a specific form of the F-test. It compares a model with no predictors to the model that you specify. A regression model that contains no predictors is also known as an intercept-only model.

The hypotheses for the F-test of the overall significance are as follows:

Null hypothesis The fit of the intercept-only model and your model are equal.

Alternative hypothesis The fit of the intercept-only model is significantly reduced compared to your model.

The F for the 2G model is 17.4 > $f_{0.95,9,297-10}$ = 1.9126 so the null-hypothesis is rejected.

If the P value for the F-test of overall significance is less than your significance level, you can reject the null-hypothesis and conclude that your model provides a better fit than the intercept-only model. P-value is $3.69 * 10^{-21}$ so this can be assumed to be true.

5.5.2. Adjusted R^2

As seen in section 3.2.2 the adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the

model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it's usually not. It is always lower than the R-squared.

In the consideration of models seen in table 5.6, you can see that R^2 is sensitive to the number of parameters in the model, so it is better to consider $\tilde{R}^2$. The model chosen has a $\tilde{R}^2 = 0.3077$

5.5.3. Root mean square error (MSE)

This value indicates the mean of squared errors. As the errors has 0 mean, to quantify them the squared error is used. As the number of observations can vary, we use MSE that is the mean of all squared errors. The lower this value is, the better the model.(3.27)

5.5.4. Coefficient of variation (CV)

For any variable with non zero mean, this coefficient is the ratio between its standard deviation σ and its mean. This coefficient helps giving and idea of how big is the deviation with respect to the measure itself.

$$CV = \frac{\sigma_z}{\overline{z}} \tag{5.8}$$

5.6. Testing model assumptions

5.6.1. Three or more variables that are of metric scale

This condition is met because we use up to six of this kind of variables (quantitative). These variables are the three distances to the towers (m) and the three angles these towers define with the located point (rad).

5.6.2. Identify outliers

To identify outliers, we followed two methods. The first one, we removed those which Cook's distance is greater than 3 times the Cook's distance average. In our case, not many outliers of this kind are revealed. And they are not significant. Cook's distance of a residual measures how much that observation can modify the whole model coefficients. It can be calculated as the value of the residual times the leverage (distance of the observation to the centroid of the rest of observations). We applied Cook's distances to both Google Accuracy and Error to detect outliers in them. The figure 5.10 is and example of an examination of application of Cook's distance to Google accuracy in 2G. The observations are numbered in x axis, and Cook's distance is in y axis expressed in [m].

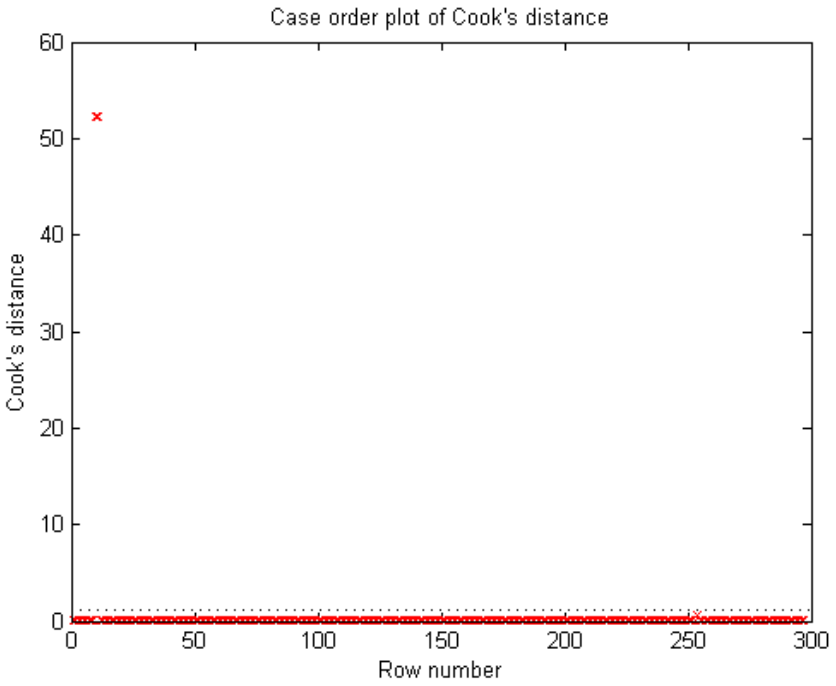


Figure 5.10: Model with no variable interactions Cook Distance accuracy.You can see one single outlier really far away that is really influential to the data.

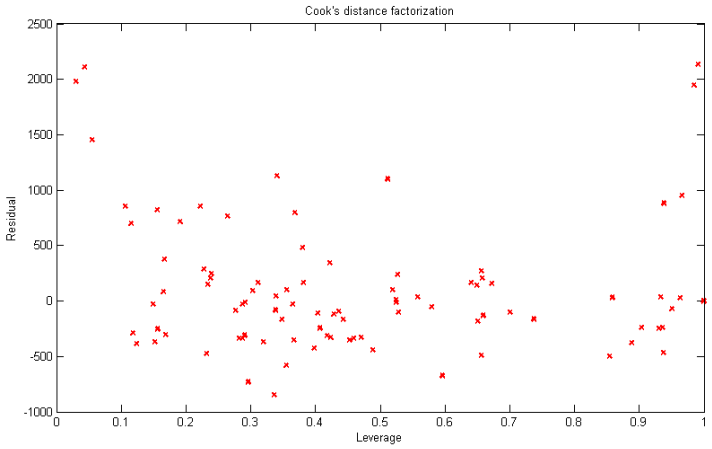


Figure 5.11: Factorized Cook's distance.

In figure 5.11 Cook's distance is represented as the two factors that form it. In x axis the leverage and y axis the residual. Values far away from the axes (Residual = 0 and distance = 0) are the ones with the biggest Cook's distance.

Another method to remove outliers is to check the observations with Pearson's residuals unusually big. As the Pearson residuals are already normalized, those with a value bigger than 3 in absolute value (equivalent to 3 times σ in Raw residuals) are neglected. See figure 5.12.

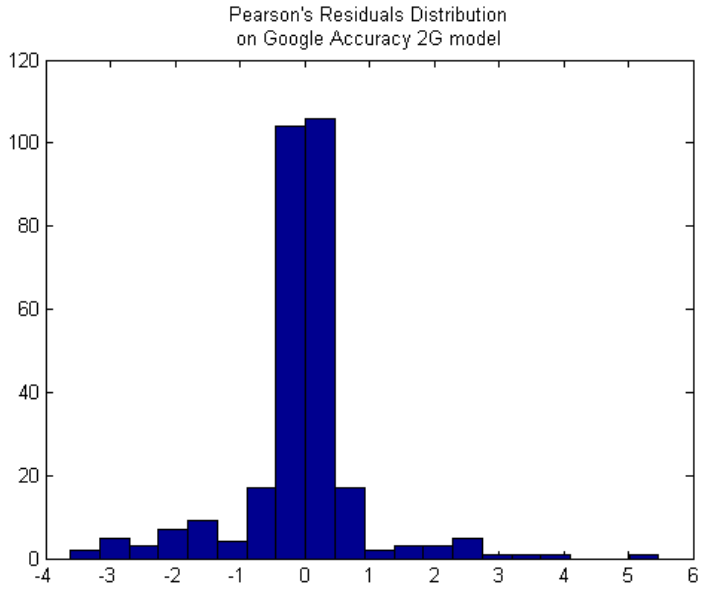


Figure 5.12: Histogram of Pearson residuals

5.6.3. Violations of linearity or additivity

Next step was to add interactions. In order to know which interaction is best to add, all the possible ones are tested one by one. As there are k parameters, there are $\binom{k}{2}$ possible interactions. The chosen model has been added one by one possible interactions and checked once again the values of R^2, $\tilde{R}^2$, C_p and S. See table 5.7.

		R^2	$\tilde{R}^2$	S
1	+ d2:a1	0.7000	0.6904	240.8681
2	+ d2:a2	0.6775	0.6672	249.7056
3	+ d2:a3	0.6396	0.6280	263.9926
4	+ d2:Action	0.6543	0.6407	259.4794
5	+ d2:Environment	0.6420	0.6305	263.1192
6	+ a1:a2	0.6404	0.6289	263.7070
7	+ a1:a3	0.6510	0.6398	259.7713
8	+ a1:Action	0.7342	0.7237	227.5174
9	+ a1:Environment	0.6864	0.6763	246.2708
10	+ a2:a3	0.6537	0.6426	258.7613
11	+ a2:Action	0.6559	0.6423	258.8915
12	+ a2:Environment	0.6706	0.6601	252.3808
13	+ a3:Action	0.6529	0.6392	260.0114
14	+ a3:Environment	0.6396	0.6280	263.9926
15	+ Action:Environment	0.6396	0.6294	263.5248

Table 5.7: Different possible interactions to add to the model. It can be seen that most of the $\tilde{R}^2$ have a significant improvement with respect to the simple model. The best model seems to be number 8. It has the biggest R adjusted $\tilde{R}^2$ and the smallest root mean square MSE S = 227.52.

This procedure can be applied several times, with several variables. Adding another interaction the $\tilde{R}^2$ will rise to 0.74.

5.6.4. Independence of observations

To test for non-time-series violations of independence, we have to look at plots of the residuals versus independent variables or plots of residuals versus row number in situations where the rows have been sorted or grouped in some way that depends (only) on the values of the independent variables [45]. The residuals

should be randomly and symmetrically distributed around zero under all conditions, and in particular there should be no correlation between consecutive errors no matter how the rows are sorted, as long as it is on some criterion that does not involve the dependent variable. If this is not true, it could be due to a violation of the linearity assumption or due to bias that is explainable by omitted variables (say, interaction terms or dummies for identifiable conditions). These conditions seems to meet quite good as seen in graph 5.13.

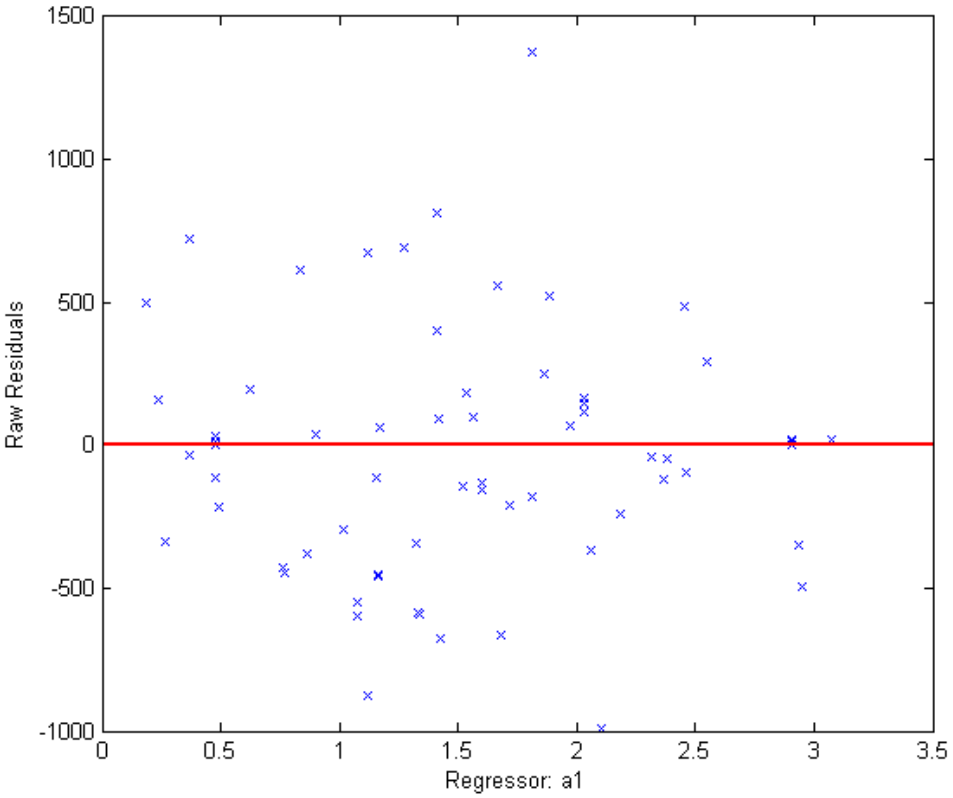


Figure 5.13: Residuals of the angle variable a1. The aspect is good because they don't seem to become larger as a1 increases. This visual inspection was done with every variable used in the model. The categorical variables appear as vertical lines.

5.6.5. Heteroscedasticity

We have to take a look at a plot of residuals versus predicted values and be alert for evidence of residuals that grow larger as a function of the predicted value. To be really thorough, it is advised to also generate plots of residuals versus independent variables to look for consistency there as well. Because of imprecision in the coefficient estimates, the errors may tend to be slightly larger for forecasts associated with predictions or values of independent variables that are extreme in both directions, although the effect should not be too dramatic. What you hope not to see are errors that systematically get larger in one direction by a significant amount.

There can be seen that this is not a problem in figure 5.14.

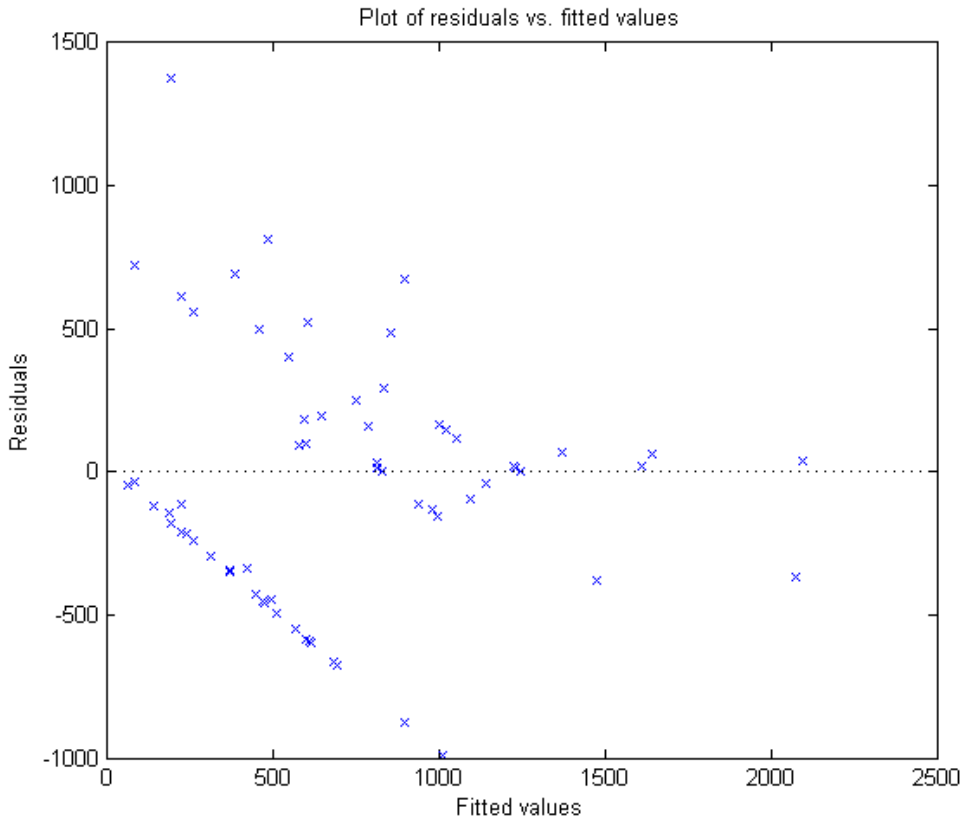


Figure 5.14: Residuals versus fitted values for Google Accuracy in 2G. There are not uniform around zero, so that means the variance of the noise is not constant with the residuals.

5.6.6. Multicollinearity

In order for the model to work, we have to check collinearity. Collinearity is a linear dependence among the variables that are supposed to be independent. The linear model, to have a unique and convergent solution, needs to be generated from a full **A** rank matrix. A test was done in the matrix **A** to see if this is an issue in our model.

It was tested with Belsley collinearity diagnostics and it didn't show strong signals of multi-collinearity [15] [14]. This was executed with Matlab function collintest. See figure 5.15.

```
>> collintest(v2,'plot','on')

Variance Decomposition

 sValue  condIdx    d2      a1      a2      a3    Action  Environment
-------------------------------------------------------------------------
 2.2143    1      0.0113  0.0033  0.0040  0.0057  0.0057     0.0030
 0.7392   2.9955  0.8229  0.0133  0.0037  0.0049  0.0040     0.0005
 0.5181   4.2737  0.0074  0.0338  0.1822  0.2030  0.0693     0.0033
 0.4050   5.4667  0.0027  0.0153  0.0282  0.2985  0.5726     0.0197
 0.2677   8.2706  0.1422  0.4524  0.0620  0.2405  0.0419     0.4871
 0.2155  10.2759  0.0135  0.4818  0.7199  0.2475  0.3065     0.4864
Warning: No critical rows to plot.
```

Figure 5.15: Belsley collinearity test. this example shows the collinearity test executed in A matrix corresponding to Google Accuracy 2G model. No rows have a condition index greater than 30 (default tolerance).

5.6.7. Normality of residuals

We have to check the residuals follow a normal distribution. Sometimes the error distribution is "skewed" by the presence of a few large outliers. Since parameter estimation is based on the minimization of squared

error, a few extreme observations can exert a disproportionate influence on parameter estimates. Calculation of confidence intervals and various significance tests for coefficients are all based on the assumptions of normally distributed errors. If the error distribution is significantly non-normal, confidence intervals may be too wide or too narrow.
The input are the residuals of both Google Accuracy and error.

In the image 5.16, Google Accuracy residuals for 2G normality check is done.

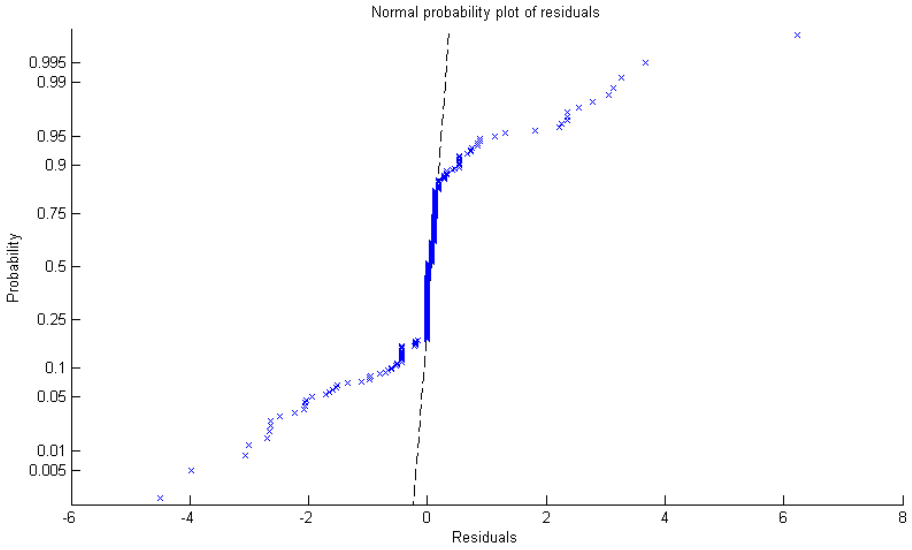


Figure 5.16: Accuracy 2G residuals plot. The farther away the residuals are from the line, the less resemblance to a normal distribution they will have.

5.7. Predict or simulate responses to new data

5.7.1. K-Fold validation

To evaluate model fitness to the data the K-fold cross validation was executed.

This test consists in divide the available data into several K bins, for example 12, in a random way, but with equal or very similar number of observations.
Then one per one, each bin of data is considered a test set, and the rest are considered the train data. This train data is used to generate a new model, applying the same criteria used that the original model (the one we want to test).
Using this new model, we predict the results for the test data, and compare to the observed values. This process is executed as many times as the number of bins created, so, every observation is used as test in one model an K-1 times as train data.
The results of this test is shown it table 5.8

Google Accuracy [m]			
$\overline{y}$	$\overline{y_{pred} - y_{obs}}$	D($y_{pred} - y_{obs}$)	$\sqrt{MS_E}$
947.2000	-6.9573	333.3653	326.7040
1031.4000	-16.2627	278.6512	273.5052
1094.3200	-14.2808	195.3043	191.8905
1067.1667	-52.7216	299.9072	298.2889
1348.4400	141.2391	747.4885	745.8806
1104.7917	-33.4312	244.6752	241.8454
1226.1600	152.8585	1016.3613	1007.4901
1205.5833	78.0555	676.6735	667.0090
1081.2000	-64.0756	408.0055	404.8647
977.0400	-71.6064	291.0883	294.0588
1038.1200	-51.5027	209.5393	211.6672
1171.5200	-57.2045	284.9433	284.9865

Table 5.8: K-Fold test for Accuracy 2G model

Each row of the table is the result of the model generated of each division of data.
The first column is the mean of the response variable (observations) in the test bin. The second column is the mean difference of the predicted value against the measured value. It is the bias of the train model. If the new partial model had the same coefficients than the whole model, the predicted results would be near the measures (first column), and this column (the difference) would approach zero
The third column is the standard deviation of the errors. It is important to remark that in this case, the mean error is different from zero, as the test data is not part of the model's coefficient calculation. And the fourth is the squared root of the mean squared error. The values are very near of the values in third column because they represent similar magnitude.

6

Google Accuracy and Error Assessment

Once all available data is collected from different sources and put together, some preliminary studies have been done to have a global impression of the characteristics of this information. In this chapter these results are shown numerically and graphically.

6.1. Hits and misses concept

The data Google provides is a location, and a measure of accuracy. Location is expressed as latitude and longitude. Accuracy is the radius expressed in meters of the circle around the given location, where the mobile device can be. The experiments consist in registering this information, and at the same time the real location provided by a *ground truth* device. With this information *Google error* can be derived as the distance between the location provided by Google and the actual location. A *hit* is defined when Google provides an accuracy greater than the real error (the mobile device is inside the circle), and a *miss* when the error is greater than the accuracy. Figure 6.1 defines this concept.

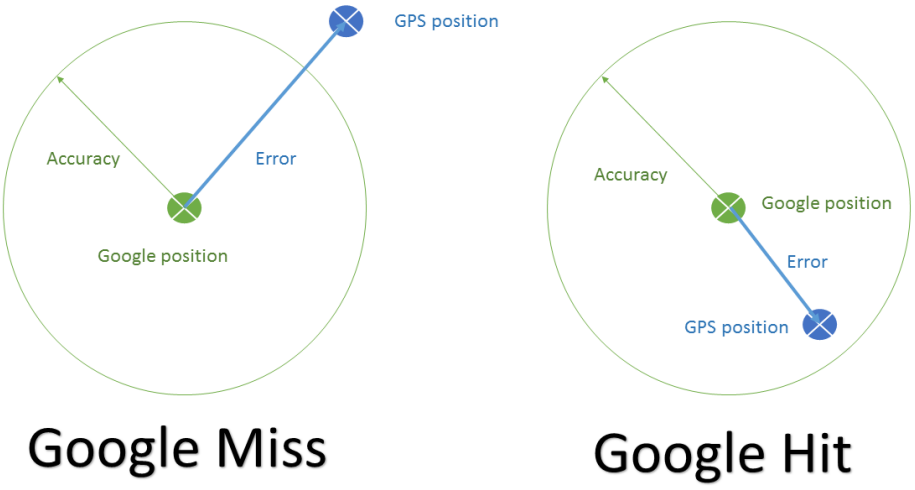
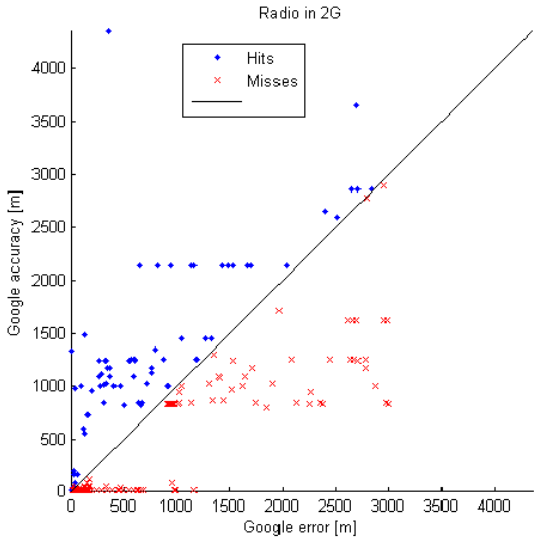


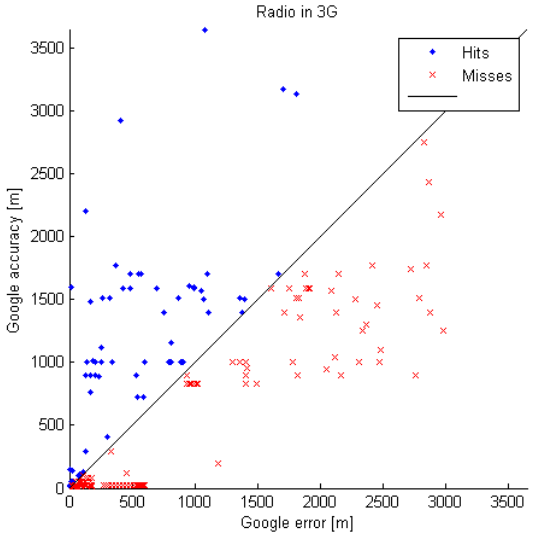
Figure 6.1: Description of what is considered a Google hit/miss. Google gives a radius of accuracy in meters where it is possible to find the device at a given time. Measuring the distance to the "Ground Truth" point (location provided by GPS device) we determine if the device was truly inside the circle. If it is, we call it a *Hit*, otherwise it is a *Miss*.

With this criteria Figures 6.2 show hits and misses statistics depending on the source of signal used.

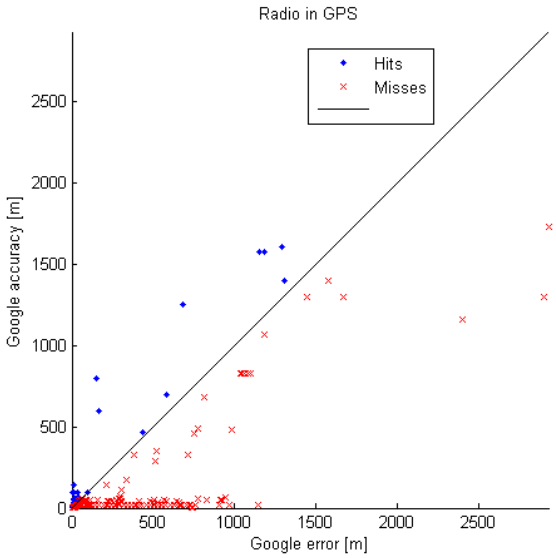
Figure 6.2: Google Error and Accuracy in the same plane. These figures represent each epoch in the experiment with the real Google error in X-axis and Google provided accuracy in Y-axis. The epochs above the unity slope line are hits and the epochs below are misses. In this graphs the magnitudes of error and accuracy and number of experiments can be visualized.



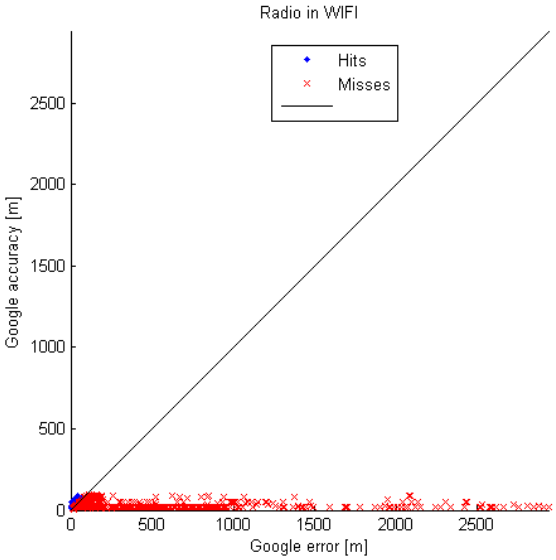
(a) When source of signal is 2G there is a cloud of misses with accuracy below 100 m and error between 100 and 700 m. There is a strip of epoch with accuracy around 1000 m, some of them are hits and other misses.



(b) When source of signal is 3G the distributions of epochs are similar to 2G. There is a group of misses with error below 150 m, other around 550 m with a low accuracy (less than 100 m) and there are some misses very close to the border (black diagonal) of hit/miss



(c) When source of signal is GPS there are many epochs near the origin, that is, low values of error and accuracy. There are few epochs with accuracy above 500 m. These are not normal values for a GPS location, they come from the first moments of each experiment. When a GPS device is switched on, it takes several minutes to acquire its own location because it has to receive the satellites' ephemerides. A detail on this figure is shown in figure 6.3a

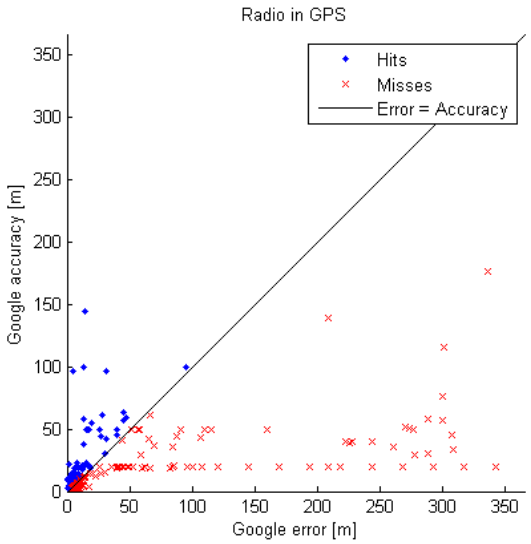


(d) When source of signal is WiFi, the accuracy is very low (below 100 m). There is a concentration of epochs with error below 150 m, but there are many errors above 2500 m. A detail on this figure is shown in figure 6.3b

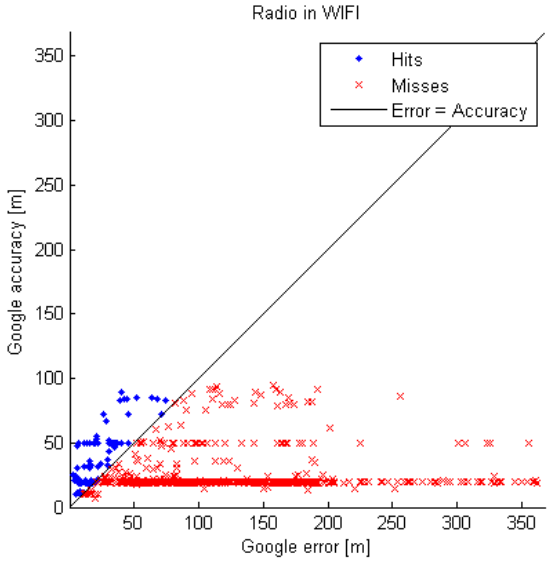
In figures 6.2a and 6.2b, the horizontal dot strips mean that Google provides accuracies in certain ranges, as prefixed or preferred values. In 6.2a points are around an accuracy of 1000 m and in 6.2b, about 1000 and 1500 m. In figure 6.2c the dots accumulate around the x-axis, which means that Google provides low radius

of accuracy even when the actual error may be any value, and around the unity slope line, which means that Google provides an accuracy radius near the actual error.
In figure 6.2d all the dots are near the x-axis, that means that Google provides low values for accuracy radius even when the actual error is more than 2000 m.

Figure 6.3: Detailed Google Error and Accuracy in the same plane for GPS and WiFi.



(a) When source of signal is GPS there are clouds of misses with accuracy about 20, 50 and 100 m.



(b) When source of signal is WIFI there are clouds of misses with accuracy about 20, 50 and 80 m.

If we enlarge the images 6.2c and 6.2d we obtain figures 6.3. In them we can observe that there are also dot horizontal stripes (preferred values for Google accuracy) around 20, 50 and 100 m for GPS signal, and about 20, 50 and 80 for WiFi.

A more simple and quantitative view of hits and misses is shown in figure 6.4. This figure shows that GPS has the highest hits ratio (52% of hits), followed by 3G, then 2G and the latest is WiFi with only 7% of hits. This low hit ratio in WiFi is because Google is too optimistic when calculating locations using WiFi networks. In further results we'll see that the errors are not huge when using this network.

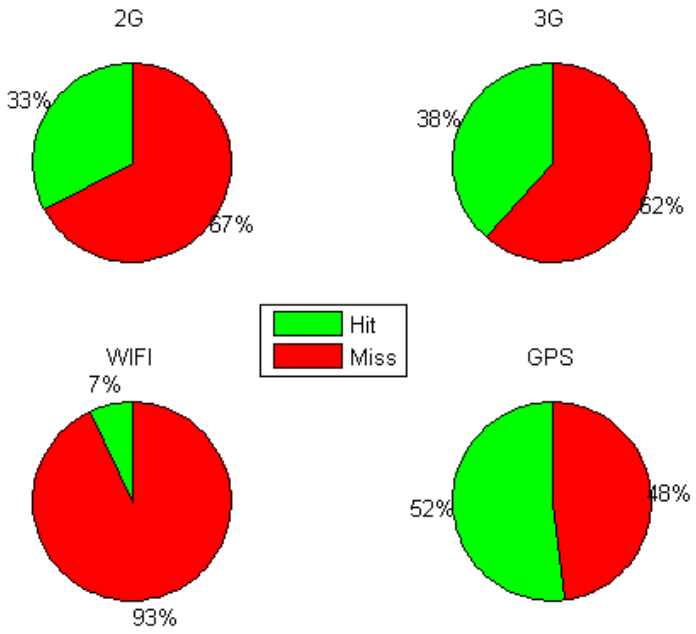


Figure 6.4: Google Hits classified by 2G/3G/WiFi and GPS. (See figure 6.1 to find a definition of Google Hit and Miss). Both 2G and 3G show similar results. GPS goes on a first position and Wi-Fi shows the the worst result. The reasons for this is that Google gives a large accuracy values for Cell ID positioning (2G and 3G) and a much smaller radius for WiFi and GPS, making it more prone to have a miss. Nevertheless, with GPS the location is better and the precision is inside the provided accuracy.

Having a look at figures 6.1 and 6.4 we could guess that Google, before assigning an accuracy radius to a calculated location, wonders which confidence interval can apply. If it takes a $\pm\sigma$ for its confidence interval, it would be natural (see figure 6.5) that it had a 68% of hits. Nevertheless, the hit rate seems to be lower in all the cases with respect to the expect 1 σ rule.

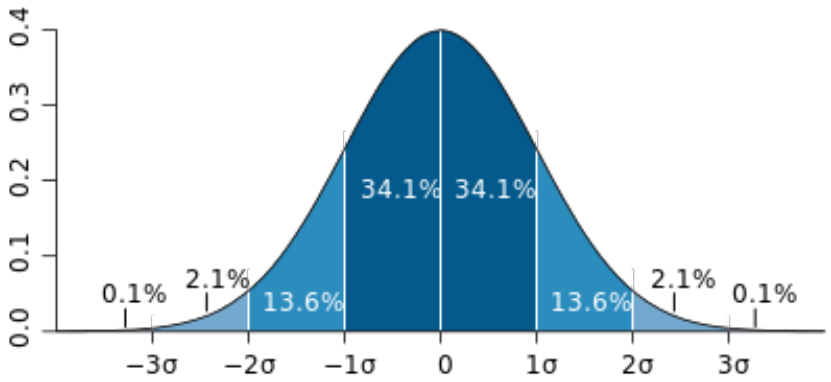


Figure 6.5: For the normal distribution, the values less than one standard deviation away from the mean account for 68.27% of the set; while two standard deviations from the mean account for 95.45%; and three standard deviations account for 99.73%.

6.2. Accuracy and Error based on phone configuration

To evaluate globally these two variables some histograms were obtained. *Accuracy* is the radius provided by Google to define the circle where the mobile device can be, and *Error* is the actual distance between the provided location and the actual location. in figure 6.6 the accuracy and error of 2G measurements are shown. But to have a better comprehension of these distributions, the cumulative distribution is used instead as shown in figure 6.7

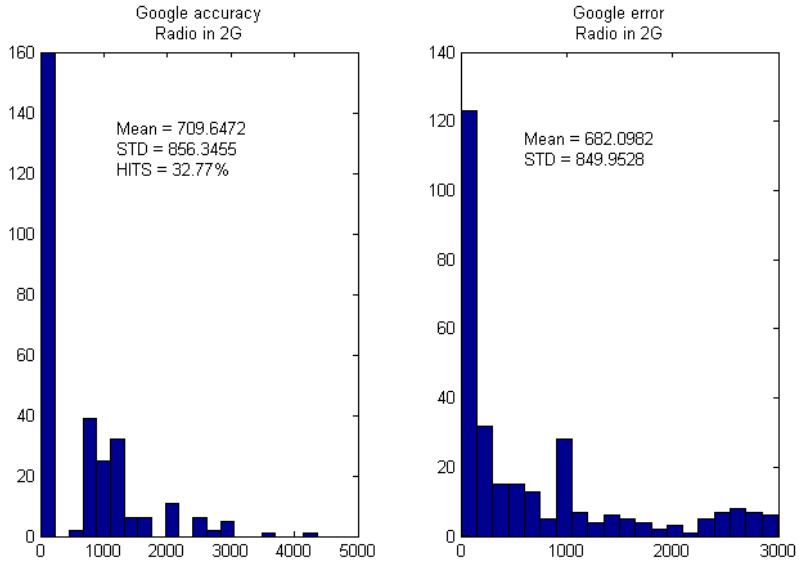
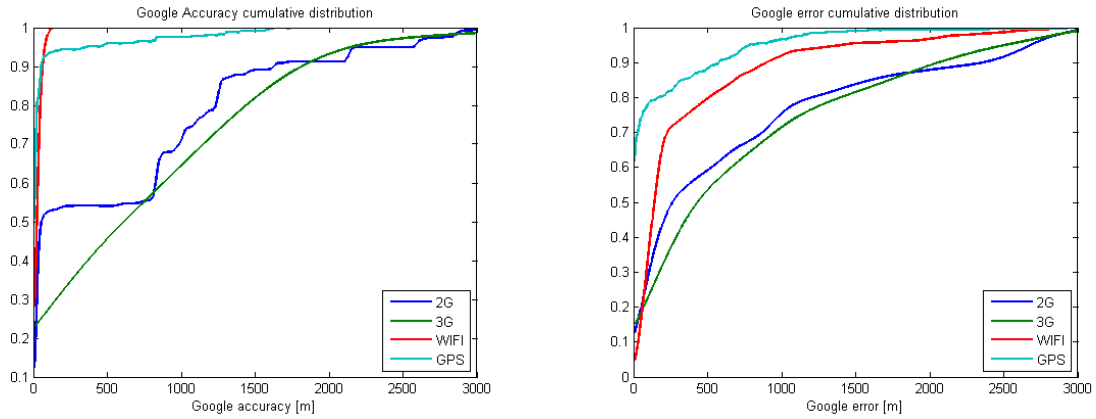


Figure 6.6: 2G Histograms. These histograms represent the Google accuracy and error when using 2G network. It can be observed that error has a wider distribution than accuracy. Google location is too optimistic providing small values of accuracy (high precision) when the location has in fact larger error



(a) Accuracy cumulative distribution function

(b) Error cumulative distribution function

Figure 6.7: Cumulative distribution functions for Accuracy and Error

In figure 6.7a The cumulative distribution function for Google accuracy is shown. It can be observed that the signal which approaches faster to unity is Wifi. This means that all the accuracies provided by Google when using this signal fall below a lower threshold, that is to say, all the accuracies are small, which means that Google is too optimistic. Next one is GPS, it gives 93% of its accuracies below 80 m and the rest of values are uniformly distributes up to 1700 m. 2G graph is a stepped one. 52% of its accuracies are below 170 m, then a new increase is about 800 m and then it grow irregularly up to 3000 m. The last signal, 3G has a uniform increase in accuracy, and is below 2G until 1800 m. When figure 6.7b is studied, one can see that the cumulative distributions or Google error are more uniform than accuracy. 2G and 3G cross each other when error equals 1800 m, and the order of functions, from best to worse is the same as accuracy: GPS, WIFI, 2G and 3G.

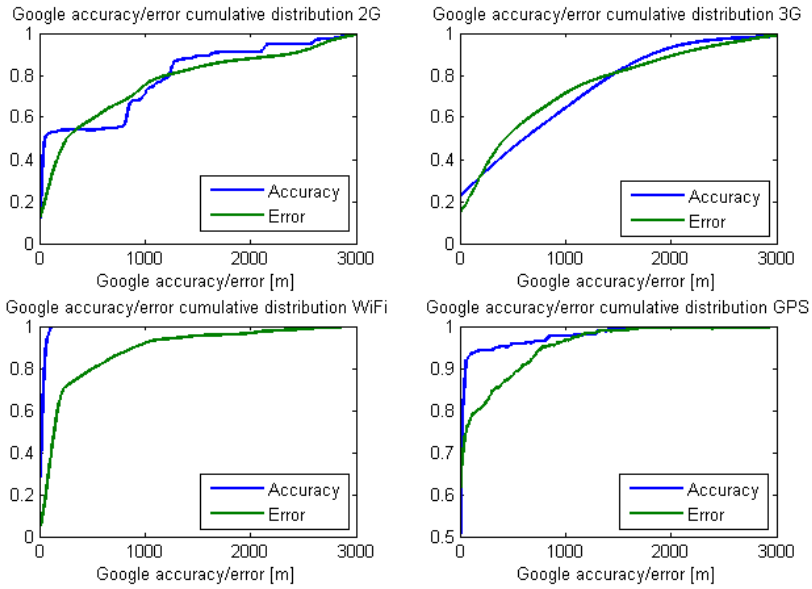


Figure 6.8: Comparative of Cumulative distributions for Accuracy and Error for each source of signal.

Observing figure 6.8 it can be noted that Google is optimistic when providing and accuracy between 300 and 1250 m when using 2G network because the number of observations in this range is larger that the number of observations with actual error in the same range. In the case of 3G, the results are similar. Google is optimistic between 200 and 1500 m.
With GPS and WiFi, Google is always optimistic giving any value for accuracy.

6.3. Accuracy and Error based on Environment

The first division of data was done with the source of signal. The second division is now studied with the Environment of the experiment. Two environments were defined for the experiments: Rural and Urban. The first one took place mainly in the zone of Gouda. The second in The Hague and Delft. The importance of this division is because the radio-electric circumstances are very different. In rural area there few obstacles to the radio signal, coming from Cell towers and Satellites. There are also less buildings where these signals can reflect and distort the measurements. But not everything are advantages for location in rural area. There are also less cell towers and WiFi networks.

In figure 6.9 it can be checked that the best hit ratio is in rural environment, using 2G signal with 68%. But this bar chart also shows that the number of events for 2G in rural environment is not high. The second in hit ratio is GPS in urban environment (62.6%) with a good significant number of events. The worst of all is WiFi in urban environment with a 7.9% if hit ratio.

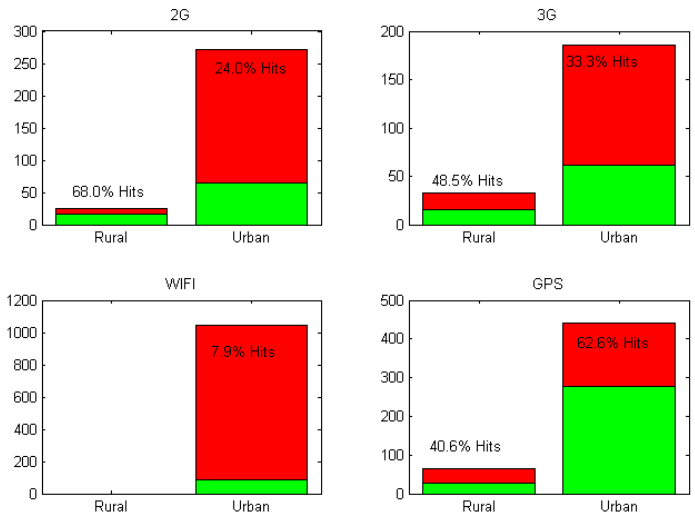
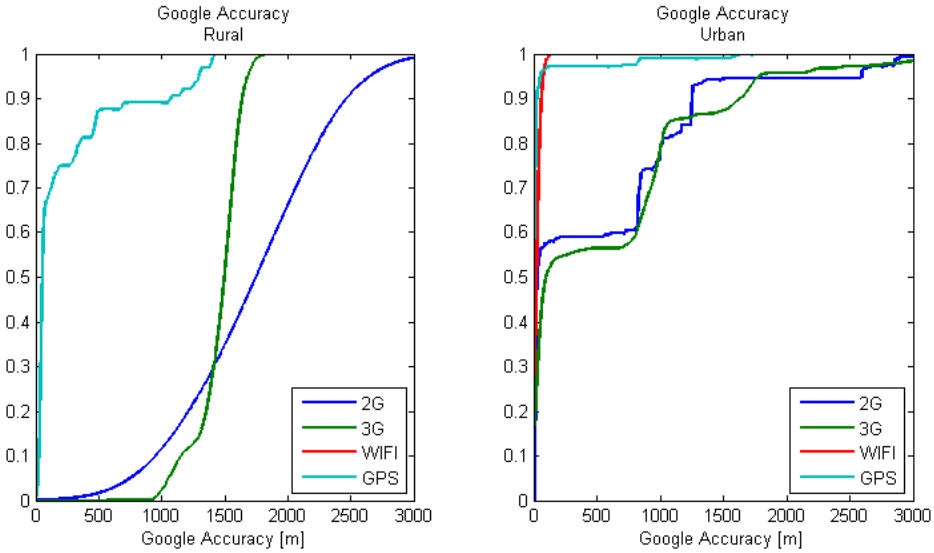
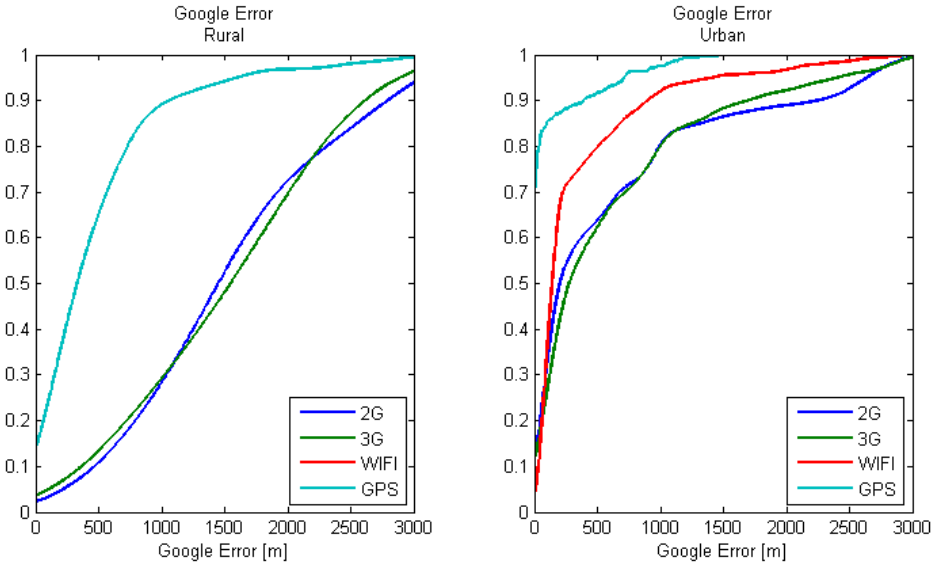


Figure 6.9: Hits related with Environment. In this chart it can be observed that there was none positioning using WiFi connection in rural environment. In rural environment 2G have more success than 3G, and in urban, 3G behaves better than 2G.In urban, the best is GPS and the worst is WiFi.

In figures 6.10 the Cumulative distribution functions have been calculated for both environments. The results for urban environment are similar to those explained for figure 6.7a, where both environments were not separated. The most important thing to remark is that both, 2G and 3G have very similar behavior. In rural environment it can be checked that the functions are very smooth. This is because there are not many measurements (experiments) and the results are homogeneously distributed. An important aspect is that there is no graph for WiFi in rural environment. In rural environment, 2G and 3G have similar error distribution, but the accuracy provided by Google is more uniform for 2G than for 3G.



(a) Accuracy CDF for different Environment. In rural environment (left) there is no graph for WiFi because the are no epochs in the experiments. GPS presents the best results, growing fast to 65% for accuracies up to 75 m, 75% for accuracies below 200 m 3G is the worst, but from 1500 it is better than 2G. 3G has no measurements below 900 m. in urban environment and 2G has a more uniform distribution.



(b) Error CDF for different Environment. In rural environment 2G and 3G have a similar behavior and GPS present a soft slope, compared to accuracy. It is strange that GPS doesn't have lower errors. This can be caused due to the elapsed time from the moment the phone is switched on to the time it starts to compute accurate locations, making some blunders using other location methods.
In assisted GPS this data is downloaded from server using other network. As 2G, 3G and WiFi were deactivated when using GPS, this can't be done, and the TTFF (Time To First Fix) is longer.
In urban environment 2G and 3G are very similar too, WiFi is available and gives better results, and the best is GPS, which has 85% of observations with error less than 100 m.

Figure 6.10: CDF for Accuracy and Error vs Environment

6.4. Accuracy and Error based on Action

The second division taken into account for studying the data is the speed and kind of movement. This information is registered under the name of *Action* and is represented by the means of transport. It is important to note that for GPS measurements, the ground truth used is a more precise device that is not able to record the data for further retrieval. So, for these experiments, a laptop is needed to get the information during its

execution. For this reason, only in *Car* experiments there are GPS measurements with actual error (and some in indoor).

In figure 6.11, a bar chart shows the hits and misses for this classification, for the four sources of signal. The height of the bars represent the number of experiments and the hits are represented in green, indicating the *hit* ratio.

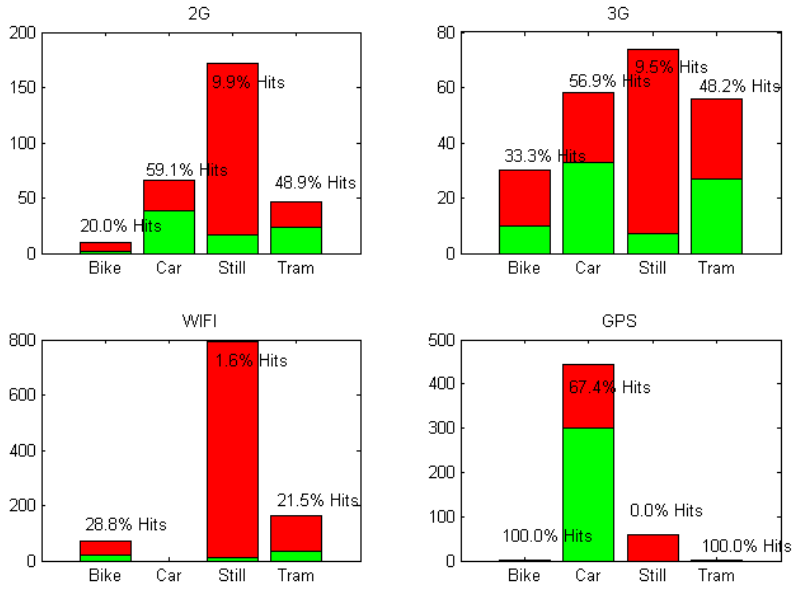


Figure 6.11: Hits for Source and Action. The best results are for GPS in *Car* with 67.4% of hits. Really, it is the only means of transport which admitted having a laptop connected, that's the reason there aren't experiments in other means of transport. The other GPS experiments can be discarded. The worst result is WIFI and Still, with 1.6% of hit ratio. This is caused because when connecting to a WiFi and indoor, the location provided by Google is the location of a unique point for the Service Set Identifier (SSID) of the network (or several networks in the same building). At the same time, the ground truth is a GPS device that doesn't provide good locations indoor, due to lack of satellite visibility. In these experiments, provided accuracy is low, and the actual error is not correctly calculated. That is why the calculated (but not real) hits are low.
In 2G and 3G, all actions have very similar *hit* ratio, and the best is *Car* (about 57%), followed by *Tram* (about 48%)

.

The same analysis has been done calculating the cumulative distribution functions (CDF) of Google accuracy (figure 6.12) and Google error (figure 6.13). The cumulative distribution function for *Bike* GPS and Wifi are very close. Both have very few samples and that's the reason they present a stepped shape. 2G has a sudden increase at 800 m, this means that most of the observations give this accuracy. In *Car* GPS has small values of accuracy, this means that in most of experiments, the satellites' ephemerides where available. If the ephemerides were not available at some moment, Google would not be able to use GPS to calculate the location, it would use any other method and probably would provide larger values of accuracy. The U-Blox device works connected to a computer, and this is able to have an updated database for ephemerides with its internet connection. This way the Time to Fix is very short. 3G has a irregular growing with two slopes that mean that many provided accuracies are about 300 m and 1800 m. In *Still*, WiFi accuracies have small radii, because the measures are taken indoor and the phone connects only to networks in the building, that have all very close locations. 3G gives small accuracies, below 150 m 92% of the measurements. This is very different from 2G that 40% of measurements are quite small and the rest are distributed in increasing distances up and above 3000 m. In *Tram* the accuracy provided by 2G and 3G are very similar, looking more precise 2G that reaches 98% of measures below 1400 m. WiFi network gives very low accuracies in every sample, that can be interpreted that WiFi networks are available at every moment in its urban trajectory.

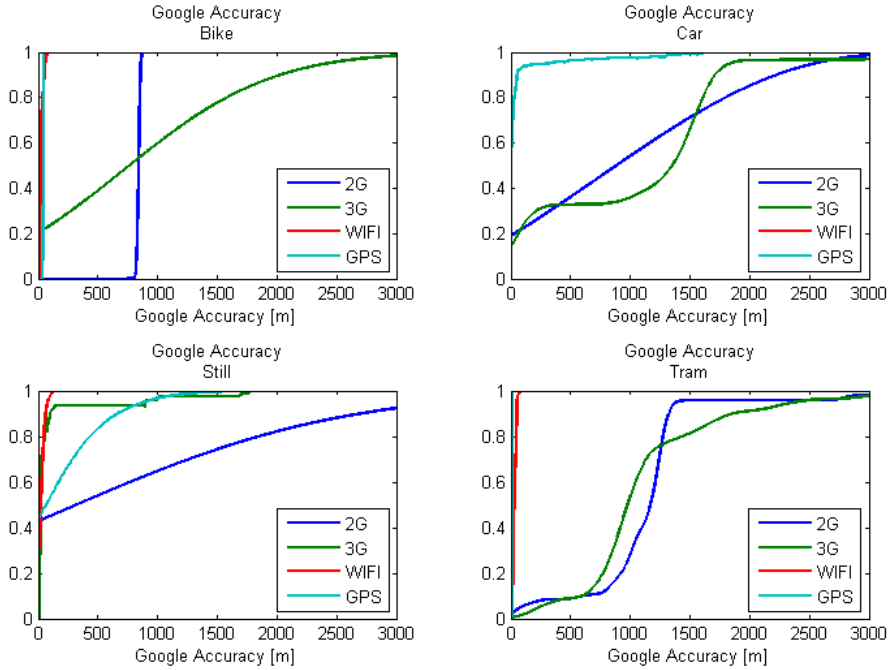


Figure 6.12: Cumulative Distribution Function for Google Accuracy and *Action*. In *Bike*, Wifi network give very small values for accuracy and 2G has a sudden increase at 800 m that means that most of the accuracies have this value. In *Car* GPS give very good values of accuracy. It was seen that in urban environment AGPS (Assisted Global Positioning System) was effective and allows the phone use this technology to determine the location. In *Still*, which is indoor WiFI give very small values of accuracy because it only connects to the same building networks. 3G has also a good rate of small accuracy values and 2G, only 40% of the measurements have very small accuracy, and the rest are distributed until very high values (more than 3000 m). In Tram, WiFi network gives very small accuracies, that means that this kind of network are available in all the trajectory.

The Google error cumulative distribution function shown in figure 6.13 describe a very different performance for the three networks (2G, 3G and WiFi) when riding a *bike*, in this order from worst to best. In *Car* the lower errors are given by GPS, as expected, and 2G and 3G have similar behavior, being better 2G. In Still, the three networks give similar errors up to 200 m in 70% of the experiments. For the remaining experiments the best is 2G, then WiFi and 3G the last.

In *Tram*, although WiFi gave low accuracies, the real error are quite big. Only 60% of measurements have an error below 150 m. 2G and 3G have similar performance, being 3G a bit better than 2G.

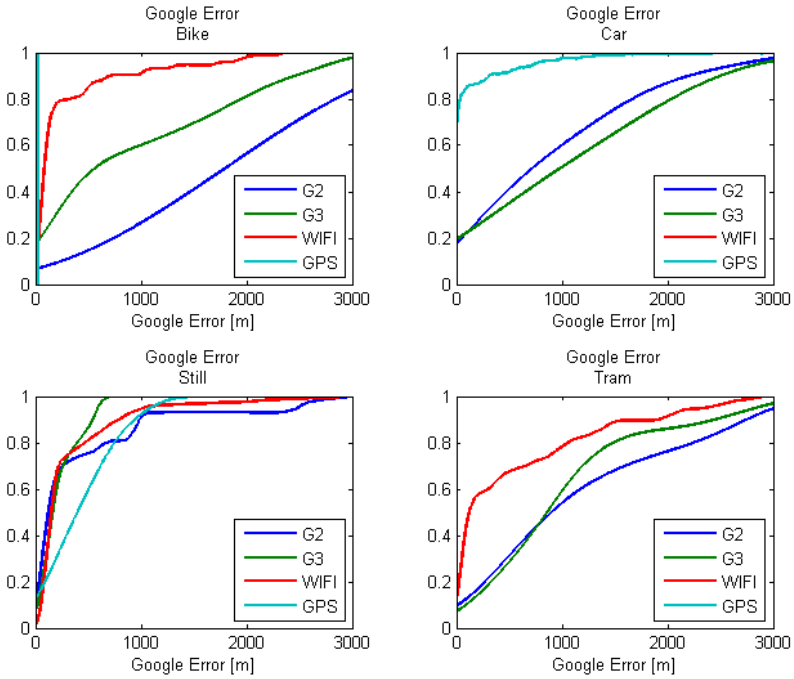


Figure 6.13: Cumulative Distribution Function for Google Error and Action. The best performance is shown in Car by GPS. In Still below 200 m of error, the three networks are similar. Above that the best is 3G. In bike the three networks are very different being from best to worst WiFi, 3G and 2G. In Tram, Wifi networks give greater errors than predicted. 2G and 3G behave similar.

6.5. Numerical Results

In order to show previous results in a numerical form, some tables have been built.

6.5.1. Google Accuracies and Errors vs Source-Environment and Source-Action

Tables 6.1 and 6.2 give an idea of accuracies reported by Google and Google error values, median and root mean square, dividing the data with two criteria: Source of signal and environment, source of signal and means of transport.

Accuracy [m]		2G		3G		WiFi		GPS	
		Median	RMS	Median	RMS	Median	RMS	Median	RMS
Environment	Rural	1627.0	1772.6	1513.0	1467.8	-	-	50.0	455.8
	Urban	23.0	922.4	82.0	932.9	20.0	31.3	9.0	198.0
Action	Bike	845.0	842.9	899.0	1057.2	20.0	31.9	44	44
	Car	964.0	1197.4	1399.0	1308.9	-	-	9.0	246.8
	Still	20.0	853.0	20.0	344.4	20.0	30.8	20.0	244.9
	Tram	1169.0	1255.7	1000.0	1276.7	23.0	34.7	13.0	13.0
	Walking	2857.0	2857.0	135.0	135.0	11.0	21.8	-	-

Table 6.1: Accuracy provided by Google expressed in meters. Table is divided into four columns which correspond to data acquired with 2G signal, 3G signal, WiFi signal and GPS. The classification is done with 2 criteria. First two rows are the Environment division and the other five are the Action. For each division two statistics are shown: median and root mean square (RMS). Urban data is better than Rural, and Tram has the worst results compared to the rest of means of transport, when the signal employed is 2G or 3G. WiFi and GPS has little variations.

Google Error [m]		2G		3G		WiFi		GPS	
		Median	RMS	Median	RMS	Median	RMS	Median	RMS
Environment	Rural	1425.2	1712.0	1712.6	1678.8	-	-	300.2	713.6
	Urban	164.6	1012.0	206.8	938.0	141.7	611.0	4.76	294.7
Action	Bike	1931.9	1978.9	416.5	1321.5	83.3	550.2	26.7	26.7
	Car	654.2	1207.8	973.1	1360.0	-	-	4.8	350.3
	Still	118.0	782.2	169.1	280.8	156.1	542.3	395.1	529.5
	Tram	878.1	1481.1	904.7	1273.6	96.6	915.1	3.1	3.1
	Walking	2656.5	2656.5	21.4	21.4	12.2	28.0	-	-

Table 6.2: Error measured in meters on Google location. Table is divided into four columns which correspond to data acquired with 2G signal, 3G signal, WiFi signal and GPS. The classification is done with 2 criteria. First two rows are the Environment division and the other five are the Action. For each division two statistics are shown: median and root mean square (RMS).

6.5.2. Google error related to Environment and Action

For showing the same data in more detail, Google error variable has been put into tables using the same division criteria. An independent table is built for each source of signal and they are tables 6.3, 6.4, 6.5 and 6.6

In tables 6.3, 6.4, 6.5 and 6.6 a more detailed study has been done on Google Error variable. The data has been classified in the same way, but the data shown is the number of observations, minimum, maximum, 50, 68 and 95 percentiles, mean, standard deviation and root mean square.

Summary 2G	Environment		Action				
Google error [m]	Rural	Urban	Bike	Car	Still	Tram	Walking
# obs	25	271	10	66	172	47	1
min	266.1	0.2	640.2	5.1	0.2	29.0	-
max	2985.1	2995.9	2995.9	2985.1	2948.5	2872.0	-
Perc 50	1425.2	164.6	1931.9	654.2	118.0	878.1	-
Perc 68	1682.1	604.7	2279.6	1284.3	175.9	1386.9	-
Perc 95	2955.7	2648.9	2994.8	2673.7	2515.6	2783.5	-
Mean	1532.8	603.6	1791.0	871.1	405.0	1152.6	-
Std	778.1	811.1	887.3	843.0	671.1	940.1	-
RMS	1712.0	1012.0	1978.9	1207.8	782.2	1481.1	-

Table 6.3: Google Error expressed in meters when using 2G signal. Two classifications are done: Environment and Action. For each of the the number of observations are given, minimum, maximum, 50,68 and 95 percentiles. There are few observations in rural environment and walking. The dispersion in car and still is very visible

Summary 3G	Environment		Action				
Google error [m]	Rural	Urban	Bike	Car	Still	Tram	Walking
# obs	33	186	30	58	74	56	1
min	265.0	3.1	21.1	3.8	11.6	3.1	-
max	2873.8	2978.3	2827.8	2873.8	602.5	2978.3	-
Perc 50	1712.6	206.8	416.5	973.1	169.1	904.7	-
Perc 68	1884.2	592.8	1396.1	1712.3.0	181.4	993.0	-
Perc 95	2742.6	2418.4	2760.3	2436.2	575.8	2857.8	-
Mean	1514.0	589.6	924.1	1025.8	217.2	1005.8	-
Std	729.7	731.5	960. 8	900.7	179.1	788.5	-
RMS	1675.8	938.0	1321.5	1359.9	280.8	1273.6	-

Table 6.4: Google Error expressed in meters when using 3G signal. Two classifications are done: Environment and Action. For each of the the number of observations are given, minimum, maximum, 50,68 and 95 percentiles. There are few observations on walking. The dispersion is high in all the rest of classifications

Summary WIFI Google error [m]	Environment		Action				
	Rural	Urban	Bike	Car	Still	Tram	Walking
# obs	-	1048	73	-	791	163	21
min	-	2.4	4.0	-	12.9	2.4	4.8
max	-	2944.1	2306.8	-	2944.1	2876.3	83.1
Perc 50	-	141.7	83.3	-	156.1	96.6	12.2
Perc 68	-	190.3	119.3	-	190.2	523.3	16.3
Perc 95	-	1403.6	1640.0	-	1009.7	2348.7	76.6
Mean	-	337.6	264.6	-	312.4	533.5	19.7
Std	-	509.5	485.7	-	443.5	745.8	20.5
RMS	-	611.0	550.2	-	542.3	915.1	28.0

Table 6.5: Google Error expressed in meters when using WiFi signal. Two classifications are done: Environment and Action. For each of the the number of observations are given, minimum, maximum, 50,68 and 95 percentiles. There are no observations on train. The values are low in walking. The rest of classifications are on similar ranges.

Summary GPS Google error [m]	Environment		Action		
	Rural	Urban	Car	Still	Tram
# obs	64	441	445	58	1
min	13.6	0.2	0.2	35.0	-
max	2898.7	2929.4	2929.4	1097.7	-
Perc 50	300.2	4.8	4.8	395.1	-
Perc 68	517.0	8.2	8.5	577.9	-
Perc 95	1605.7	713.5	743.8	1055.2	-
Mean	463.6	100.2	110.7	423.6	-
Std	546.7	277.4	332.8	320.5	-
RMS	713.6	294.7	350.3	529.5	-

Table 6.6: Google Error expressed in meters when using GPS signal. Two classifications are done: Environment and Action. For each of the the number of observations are given, minimum, maximum, 50,68 and 95 percentiles. There are no observations on bike, train and walking. Few experiments in rural area. Only car and Urban have a good number of experiments and the results ar similar because the are practically the same data.

In figure 6.6 histograms are plotted because normal distribution is not guaranteed.
Watching tables 6.3 to 6.6, specially, mean and standard deviation, one can see that it does not have the bell shape of Gauss distribution. They present long tails. For instance, in Urban GPS (table 6.6), the median and 68 percentile are 4.8 m and 8 m while the 95 percentile is 713.5 m.

6.6. Other Bar charts

The *hits* and *misses* can be subdivided depending on other factors. As figure 6.9 showed the *hits* and *misses* depending on the *Source* of signal and *Environment*, figures 6.14 and 6.15 use *Traffic* and *Weather* as second classification variable. In these charts the number of observations can be compared and show that in some cases, there are few events even none, for example, there are no measurements with WiFi connection in rural area. Some other interesting things can be seen, for example, weather conditions do affect in positioning using 2G and 3G, but not in GPS and WiFi.

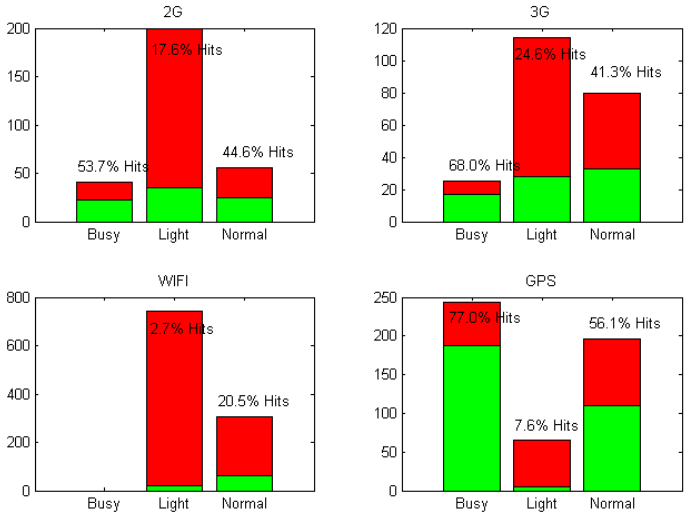


Figure 6.14: Hits related with traffic. Watching this chart, it looks that there are more *hits* when the traffic is heavier.The worst result is using WiFi with *Light* traffic, and the best is GPS with *Busy* traffic.

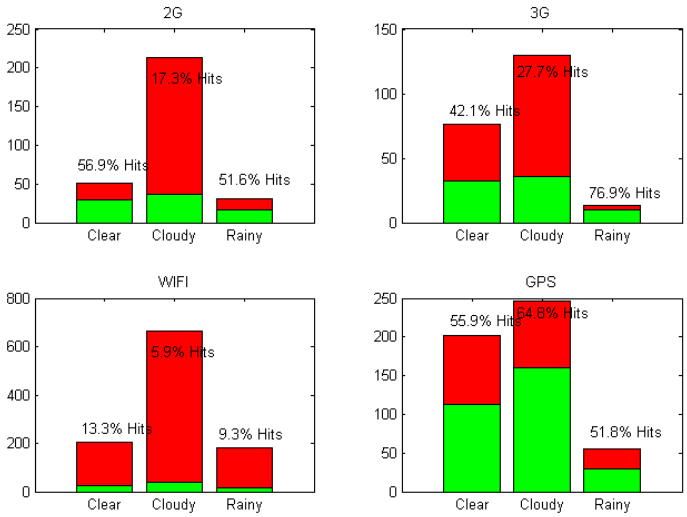


Figure 6.15: Hits related with weather. GPS has good results in the three conditions of weather, but the best result is with rainy weather using 3G. WiFi and GPS seem to be less influenced by weather (the hits ratio are similar) and 2G and 3G have higher hits rate with rainy weather. There is not a known specific reason for this behavior and it probably is a coincidence. In next chapter 7 it is checked that these conditions are not relevant.

6.7. 2G Location by Power Interpolation

A simple method of interpolation was programmed for doing a light test of data consistency. Taking data from 2G logcats, which include full tower identification and signal strength, and the official Vodafone tower database, the locations have been calculated by simple interpolation. For each available epoch, with three or more identified connected towers, the location is calculated with the tower coordinates weighted with the signal strengths as shown in section 5.3.2.

$$\bar{\mathbf{x}} = \frac{\sum(ss_i \cdot \mathbf{x_i})}{\sum ss_i} \tag{6.1}$$

Being

$\mathbf{x_i}$ latitude and longitude of i nearest cell tower.

ss_i signal strength received from i nearest cell tower.

$\bar{\mathbf{x}}$ interpolated position (latitude and longitude)

Results are in table 6.7 and in figure 6.16

Summary 2G power interpolation Error [m]	Environment		Action				
	Rural	Urban	Bike	Car	Still	Tram	Walking
# obs	-	469	15	35	352	60	7
min	-	32.7	105.8	32.7	36.0	45.8	130.5
max	-	4433.7	1218.2	1526.3	4433.7	1032.1	345.2
Median	-	345.2	289.0	419.8	338.6	402.8	156.6
Perc 68	-	600.9	380.6	735.9	614.7	506.6	174.9
Perc 95	-	1240.9	1176.1	1366.2	1360.9	978.6	345.2
Mean	-	536.3	403.2	546.5	565.5	432.9	184.1
Std	-	607.3	341.6	425.1	672.5	259.8	73.3
RMS	-	809.7	521.1	688.6	877.9	503.8	196.3

Table 6.7: Power interpolation error with 2G. This table can be compared with Google error using 2G (table 6.3). One can observe that the errors with our system are not much worse than Google's.

The distribution for error when using 2G power Interpolation in urban area (table 6.7, second column), looks a normal distribution taking into account the values of standard deviation, mean, median and 68 percentile.

In figures 6.16 one can see the "cloud" of error locations with Google location and with power interpolation. The error is represented in a two dimensional plane (North-South, West-East). There is not any special difference between the results of these two methods. With these results we can think that Google may use power interpolation when calculating accuracies when using 2G (and probably 3G) networks.

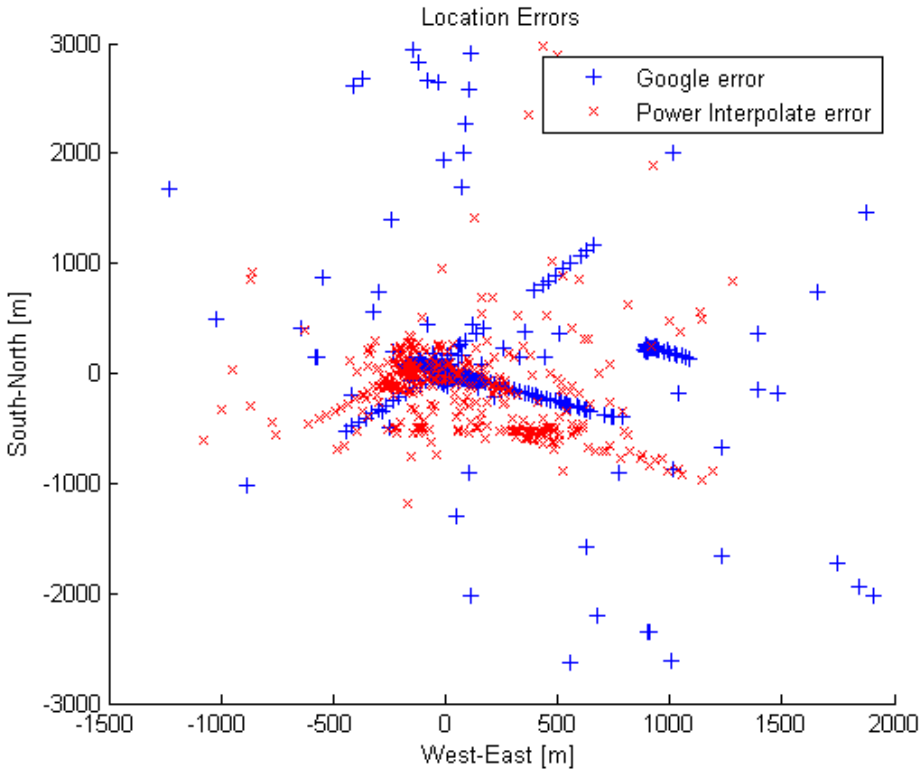


Figure 6.16: Location errors with 2G: This figure shows the errors that Google had when having 2G connection, and the errors with the simple interpolation method using power strength. The errors have similar magnitude order, and it looks that there are more Google errors far from the origin.

In figure 6.17, the location error cumulative distribution function is shown for both methods: Google location (2G) and Power interpolation. These curves cut each other when error distance is about 500 m. For error below this limit, Google gives better results (more epoch ratio) than power interpolation, but for errors higher this point, Power Interpolation has better behavior.

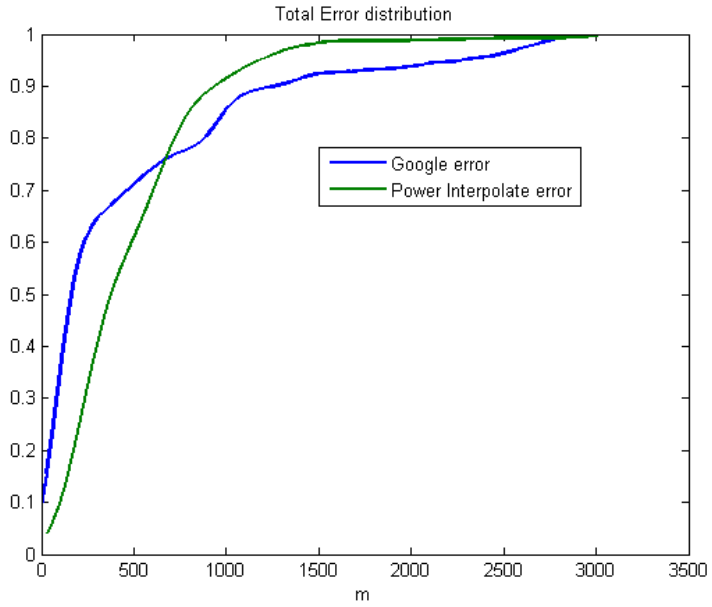


Figure 6.17: Cumulative distribution of errors of Google location and power Interpolation. It can be seen that for errors under 650 m (75% of the observations), Google gives better results, but for values higher, for example 1 km, Power interpolation has better results.

7

Prediction Model Results

In chapter 6 the data was classified depending on different variables, as the Signal Source, Environment, Traffic and Weather and represented graphically in pie and bar charts, and numerically in tables. Next part in the study is to define linear regression models for this data. As explained in chapter 5 a model was built for each response variable (Google accuracy and Google error) and each source of signal (2G, 3G, Wifi). These models used as regressors the three distances to nearest Cell-Id towers, the three angles, the environment and the Action as categorical regressors. Each model is generated in an iterative way, first looking for the best linear model with simple regressors, and then, adding interactions. At the end of this chapter we'll apply the models to new data and check validation.

The models obtained are the ones defined in next tables:

7.1. Linear models

7.1.1. Linear Models for 2G connection

The linear model obtained for Google provided accuracy when connecting to a 2G network is defined in table 7.1 and the model for Google Error when connecting to a 2G network is defined in table 7.2.

The columns for these tables correspond to the values of [48]

Estimate The estimated value for the coefficient.

SE The estimated standard error for the coefficient. $SE = \sqrt{\hat{\sigma}^2 C_{jj}}$

tStat The ratio of the Estimated coefficient and its standard error. It is a statistic to evaluate the significance of the coefficient for the model. The t-value measures the size of the difference relative to the variation in your sample data. Put another way, T is simply the calculated difference represented in units of standard error. The greater the magnitude of T (it can be either positive or negative), the greater the evidence against the null hypothesis that there is no significant difference. The closer T is to 0, the more likely there isn't a significant difference.

pValue The P-value is the smallest level of significance that would lead to rejection of the null hypothesis H_0 with the given data. In this case H_0 is that the coefficient has not significance for the model. When the p-value is very low ($< \alpha = 0.05$), we reject the null hypothesis and conclude that there's a statistically significant difference.

The values for the model details on the right are:

N#Obs Number of observations.

DOF Degrees of freedom.

R^2 **and** $\tilde{R^2}$ Correlation coefficient and adjusted corelation coefficient. See definitions in sections 3.2.2 and 3.2.2.

RMSE Root Mean Square of errors (or residuals). See definitions in section 3.1.2.

In table 7.1 some coefficients have high p-values which indicate that may not be relevant for the model. When introducing a categorical variable like *Action*, all its possible values are included, even if they do not have significance for the model. For example, the interaction *a1:Action* includes a significant coefficient (-431, comparable to the intercept 538) with a good p-value and others with smaller absolute value (have less effect on the model) with worse significance.

Accuracy 2G	Estimate	SE	tStat	pValue
(Intercept)	538.5266	159.0705	3.3855	8.1419e-04
d2	0.4770	0.0503	9.4774	1.2703e-18
a1	360.7430	28.6448	12.5937	5.0822e-29
a2	-66.2763	25.7469	-2.5742	1.0572e-02
a3	-65.7234	23.6976	-2.7734	5.9266e-03
Action_Car	257.8436	85.0358	3.0322	2.6597e-03
Action_Still	58.3721	74.9032	0.7793	4.3647e-01
Action_Train	0.0000	0.0000	-	-
Action_Tram	56.7262	215.0101	0.2638	7.9211e-01
Action_Walking	0.0000	0.0000	-	-
Environment_Urban	-335.5578	102.3598	-3.2782	1.1790e-03
a1:Action_Car	-431.2865	45.2140	-9.5388	8.1469e-19
a1:Action_Still	-9.5958	32.3309	-0.2968	7.6684e-01
a1:Action_Train	0.0000	0.0000	-	-
a1:Action_Tram	-21.4978	81.1150	-0.2650	7.9119e-01
a1:Action_Walking	0.0000	0.0000	-	-

N#Obs	291
DOF	279
R^2	0.7342
$\tilde{R^2}$	0.7237
RMSE	227.5

Table 7.1: 2G Accuracy model. Formula = 1 + d2 + a2 + a3 + Environment + a1*Action

Table 7.2 represents the model for Google Error when using 2G connection. This model has a negative intercept and there are many positive coefficients. The categorical variable *Action* as the base value on Bike, which doesn't appear in the table, and the others have positive coefficients which means that the model predicts lower errors when riding a bike than the rest of means of transport.

Error 2G	Estimate	SE	tStat	pValue
(Intercept)	-6412.3120	2947.5575	-2.1755	3.2663e-02
d1	1.5108	1.3142	1.1496	2.5386e-01
d2	1.0295	0.5291	1.9459	5.5314e-02
d3	2.6910	2.1604	1.2456	2.1669e-01
a1	1807.1640	1031.6793	1.7517	8.3812e-02
Action_Car	3196.8092	2327.9272	1.3732	1.7366e-01
Action_Still	3159.5153	2592.2855	1.2188	2.2664e-01
Action_Train	0.0000	0.0000	-	-
Action_Tram	3079.1627	2635.8276	1.1682	2.4633e-01
Action_Walking	0.0000	0.0000	-	-
Environment_Urban	2887.2094	1769.3680	1.6318	1.0681e-01
d1:a1	-0.6940	0.5741	-1.2089	2.3042e-01
d3:Action_Car	-3.1483	2.1690	-1.4515	1.5070e-01
d3:Action_Still	-3.2266	2.3665	-1.3634	1.7672e-01
d3:Action_Train	0.0000	0.0000	-	-
d3:Action_Tram	-1.8611	2.8383	-0.6557	5.1397e-01
d3:Action_Walking	0.0000	0.0000	-	-
a1:Environment_Urban	-1609.5390	698.7186	-2.3036	2.3947e-02

N#Obs	95
DOF	81
R^2	0.4980
$\tilde{R}^2$	0.4174
RMSE	658.3

Table 7.2: 2G Error model. Formula = 1 + d2 + d1*a1 + d3*Action + a1*Environment

7.1.2. Linear Models for 3G connection

The linear model obtained for Google provided accuracy when connecting to a 3G network is defined in table 7.3 and the model for Google Error when connecting to a 3G network is defined in table 7.4.

In table 7.3 one can see that the urban environment has a high negative coefficient, which means that provided accuracies in the city have lower values (more precise) than in the country. The Action variable interacting with d2 and d3 have low coefficients which means that they do not affect with high importance to the model predictions.

Accuracy 3G	Estimate	SE	tStat	pValue
(Intercept)	1457.6355	829.5868	1.7571	7.9316e-02
d2	-1.6735	0.7269	-2.3023	2.1589e-02
d3	1.3450	0.6355	2.1164	3.4640e-02
a3	1057.7028	250.5138	4.2221	2.7175e-05
Action_Car	-973.8782	299.8969	-3.2474	1.2166e-03
Action_Still	-109.9690	244.2393	-0.4503	6.5266e-01
Action_Train	0.0000	0.0000	-	-
Action_Tram	-119.9390	278.2004	-0.4311	6.6650e-01
Action_Walking	0.0000	0.0000	-	-
Environment_Urban	-1128.4664	767.2194	-1.4709	1.4175e-01
d2:d3	-0.0001	0.0001	-0.7970	4.2571e-01
d2:a3	1.3435	0.1863	7.2098	1.3721e-12
d3:a3	-1.5275	0.1637	-9.3289	1.1819e-19
d2:Action_Car	0.6521	0.6790	0.9604	3.3717e-01
d2:Action_Still	4.0701	0.6983	5.8283	8.3171e-09
d2:Action_Train	0.0000	0.0000	-	-
d2:Action_Tram	-0.2742	0.7708	-0.3558	7.2209e-01
d2:Action_Walking	0.0000	0.0000	-	-
d3:Action_Car	-0.0763	0.5314	-0.1437	8.8581e-01
d3:Action_Still	-4.1313	0.5604	-7.3727	4.4351e-13
d3:Action_Train	-0.4534	0.8318	-0.5451	5.8585e-01
d3:Action_Tram	0.0127	0.6578	0.0193	9.8458e-01
d3:Action_Walking	-0.5087	0.4550	-1.1180	2.6394e-01
a3:Action_Car	60.2524	94.5286	0.6374	5.2406e-01
a3:Action_Still	245.4190	58.5320	4.1929	3.0834e-05
a3:Action_Train	0.0000	0.0000	-	-
a3:Action_Tram	66.4652	72.8931	0.9118	3.6216e-01
a3:Action_Walking	0.0000	0.0000	-	-
d3:Environment_Urban	1.2041	0.3735	3.2238	1.3200e-03
a3:Environment_Urban	-823.5996	198.9028	-4.1407	3.8561e-05

Nobs	779
DOF	755
R^2	0.5154
$\tilde{R^2}$	0.5006
RMSE	373.9

Table 7.3: 3G Accuracy model. Formula = 1 + d2*d3 + d2*a3 + d3*a3 + d2*Action + d3*Action + a3*Action + d3*Environment + a3*Environment

Table 7.4 represents the model for Google Error when using 3G network. One can observe that *Action* variable with *car* and *tram* values give negative coefficients which reduce the predicted Google error in this model. *Urban* environment also affects in this direction. This model starts with a very high intercept and many negative coefficients.

Error 3G	Estimate	SE	tStat	pValue
(Intercept)	12104.4280	2497.5505	4.8465	3.4465e-06
d1	4.9330	1.5668	3.1484	2.0269e-03
d2	-0.4902	1.5352	-0.3193	7.5000e-01
d3	-10.2206	3.2717	-3.1239	2.1910e-03
a1	-1604.7647	620.8971	-2.5846	1.0828e-02
Action_Car	-8357.2590	2352.9617	-3.5518	5.2988e-04
Action_Still	0.0000	0.0000	-	-
Action_Train	0.0000	0.0000	-	-
Action_Tram	-6768.7982	2664.9852	-2.5399	1.2239e-02
Action_Walking	0.0000	0.0000	-	-
Environment_Urban	-4314.5144	951.2371	-4.5357	1.2683e-05
d1:a1	0.5951	0.1985	2.9975	3.2487e-03
d2:a1	-0.2983	0.2284	-1.3059	1.9384e-01
d1:Action_Car	-6.0149	1.5746	-3.8199	2.0400e-04
d1:Action_Still	-6.3668	3.9504	-1.6117	1.0940e-01
d1:Action_Train	0.0000	0.0000	-	-
d1:Action_Tram	-3.0674	1.6739	-1.8324	6.9125e-02
d1:Action_Walking	0.0000	0.0000	-	-
d2:Action_Car	1.8083	1.4375	1.2580	2.1060e-01
d2:Action_Still	11.9492	12.6380	0.9455	3.4612e-01
d2:Action_Train	0.0000	0.0000	-	-
d2:Action_Tram	2.1537	2.1077	1.0219	3.0870e-01
d2:Action_Walking	0.0000	0.0000	-	-
d3:Action_Car	9.1693	3.2493	2.8220	5.5059e-03
d3:Action_Still	-7.8004	9.4486	-0.8256	4.1053e-01
d3:Action_Train	0.0000	0.0000	-	-
d3:Action_Tram	5.6951	3.9205	1.4526	1.4869e-01
d3:Action_Walking	-1.1957	0.7445	-1.6062	1.1060e-01
a1:Action_Car	949.8238	413.6703	2.2961	2.3235e-02
a1:Action_Still	0.0000	0.0000	-	-
a1:Action_Train	0.0000	0.0000	-	-
a1:Action_Tram	1166.3151	456.9901	2.5522	1.1836e-02
a1:Action_Walking	0.0000	0.0000	-	-
d2:Environment_Urban	0.5471	0.6145	0.8904	3.7487e-01
d3:Environment_Urban	0.8939	0.4922	1.8162	7.1588e-02
a1:Environment_Urban	858.2211	421.1817	2.0377	4.3568e-02

Nobs	169
DOF	144
R^2	0.5351
$\tilde{R}^2$	0.4576
RMSE	587.4

Table 7.4: Error 3G model. Formula = 1 + d1*a1 + d2*a1 + d1*Action + d2*Action + d3*Action + a1*Action + d2*Environment + d3*Environment + a1*Environment

7.1.3. Linear Models for WiFi connection

The linear model obtained for Google provided accuracy when connecting to a WiFi network is defined in table 7.5 and the model for Google Error when connecting to a WiFi network is defined in table 7.6.

In table 7.5 we can see that this model has a very low value of R^2 which means that this model is not much better predicting values of accuracy than the constant model. At the same time we see that the coefficient values have low absolute value compared to the intercept. This means that the regressors adopted are not significant for the model. The Wifi google provides when using WiFi network doesn't depend on the distances to the telephony towers. The generated model looks to be a flat hyperplane 3.1, where all the slopes are very small.

WiFi Accuracy	Estimate	SE	tStat	pValue
(Intercept)	53.8240	12.9906	4.1433	3.5242e-05
d1	0.0033	0.0038	0.8819	3.7792e-01
d2	-0.0856	0.0166	-5.1505	2.7768e-07
d3	0.0525	0.0138	3.8119	1.4085e-04
a1	-0.4403	2.0616	-0.2136	8.3089e-01
a2	3.1686	0.8963	3.5350	4.1432e-04
a3	-5.5590	0.5855	-9.4952	4.5194e-21
Action_Car	-0.3023	11.2049	-0.0270	9.7848e-01
Action_Still	-9.2586	7.8049	-1.1863	2.3562e-01
Action_Train	-0.7817	11.0216	-0.0709	9.4346e-01
Action_Tram	15.1342	9.0515	1.6720	9.4634e-02
Action_Walking	8.7562	9.7480	0.8983	3.6912e-01
Environment_Urban	-8.8684	11.0490	-0.8026	4.2225e-01
d2:a1	0.0037	0.0018	2.0673	3.8800e-02
d2:Action_Car	0.0782	0.0173	4.5325	6.0717e-06
d2:Action_Still	-0.0288	0.0219	-1.3139	1.8899e-01
d2:Action_Train	0.0703	0.0459	1.5301	1.2609e-01
d2:Action_Tram	0.0185	0.0195	0.9459	3.4428e-01
d2:Action_Walking	-0.0139	0.0353	-0.3930	6.9436e-01
d3:Action_Car	-0.0517	0.0152	-3.3932	7.0035e-04
d3:Action_Still	0.0135	0.0165	0.8185	4.1315e-01
d3:Action_Train	-0.0364	0.0381	-0.9544	3.3996e-01
d3:Action_Tram	-0.0272	0.0171	-1.5928	1.1131e-01
d3:Action_Walking	0.0016	0.0283	0.0579	9.5387e-01
a1:Action_Car	-4.0601	4.6340	-0.8761	3.8103e-01
a1:Action_Still	6.0872	2.0880	2.9153	3.5818e-03
a1:Action_Train	2.5842	4.3298	0.5968	5.5066e-01
a1:Action_Tram	-3.3663	2.4810	-1.3568	1.7494e-01
a1:Action_Walking	-3.8819	4.3777	-0.8867	3.7529e-01

Nobs	2850
DOF	2821
R^2	0.1635
$\tilde{R^2}$	0.1552
RMSE	18.6

Table 7.5: WiFi Accuracy model. Formula accu ~ 1 + d1 + a2 + a3 + Environment + d2*a1 + d2*Action + d3*Action + a1*Action

Table 7.6 represents the model for Google Error when using WiHi network. this model presents a very low value of R^2 but in this case some coefficients have absolute value higher than the intercept. This means that although Google gives always the same accuracy (with low variations) really, the error committed depends on the speed of movement.

WiFi Error	Estimate	SE	tStat	pValue
(Intercept)	200.8308	430.5493	0.4665	6.4101e-01
d1	1.4821	0.5203	2.8488	4.4898e-03
d2	-0.1220	0.1416	-0.8611	3.8940e-01
a2	168.5409	140.1258	1.2028	2.2938e-01
a3	-18.6968	100.9664	-0.1852	8.5313e-01
Action_Car	331.7308	338.4075	0.9803	3.2722e-01
Action_Still	1066.7953	356.9618	2.9885	2.8801e-03
Action_Train	0.0000	0.0000	-	-
Action_Tram	811.5952	310.4085	2.6146	9.0840e-03
Action_Walking	-570.9371	2587.0432	-0.2207	8.2538e-01
Environment_Urban	-558.9229	328.7634	-1.7001	8.9466e-02
a2:a3	-61.4782	43.3133	-1.4194	1.5614e-01
d1:Action_Car	-1.4746	0.5341	-2.7609	5.8825e-03
d1:Action_Still	0.3088	0.9731	0.3173	7.5109e-01
d1:Action_Train	0.0000	0.0000	-	-
d1:Action_Tram	0.2196	0.5803	0.3785	7.0514e-01
d1:Action_Walking	-1.1171	3.2049	-0.3486	7.2750e-01
a2:Action_Car	-84.0505	139.6264	-0.6020	5.4735e-01
a2:Action_Still	-396.0496	160.1324	-2.4733	1.3574e-02
a2:Action_Train	0.0000	0.0000	-	-
a2:Action_Tram	-266.6296	134.7558	-1.9786	4.8167e-02
a2:Action_Walking	35.4009	691.0939	0.0512	9.5916e-01
a3:Action_Car	82.8561	97.9558	0.8459	3.9786e-01
a3:Action_Still	-122.4701	122.1130	-1.0029	3.1617e-01
a3:Action_Train	0.0000	0.0000	-	-
a3:Action_Tram	-97.0321	81.0108	-1.1978	2.3133e-01
a3:Action_Walking	246.4544	499.9163	0.4930	6.2214e-01
d2:Environment_Urban	0.3821	0.2329	1.6407	1.0121e-01

Nobs	919
DOF	895
R^2	0.1232
$\tilde{R^2}$	0.1007
RMSE	497.3

Table 7.6: WiFi Error model. Formula = 1 + a2*a3 + d1*Action + a2*Action + a3*Action + d2*Environment.

In both WiFi models, the values of R^2 and $\tilde{R^2}$ are very low, and the p-values for the coefficients are high. This takes to the conclusion that the models don't fit the observations, and it is because the methods Google uses to calculate locations and accuracies do not depend on the chosen predictors.

7.2. Urban vs Rural

As one of the important variables to take into account are the distances to the towers it is feasible that in urban environment these distances are shorter than in the rural environment.
For this exact reason, we want the model to have two working ranges.

In urban environment distances to towers are in the order of magnitude of 100 m. On the other hand, in the rural environment distance to towers are in the order of a few kilometers. This is the reason why we split the data using two categories. If we grouped urban and rural together, the coefficients that multiply the distances would be the same in both environments. Introducing the dummy variable *Environment* coefficients from urban and rural ranges are two different subsets.
For that purpose, two experiments were performed. One with rural environment (close to Gouda area, see figure 4.15) and other one in the Hague (see figure 4.16). The experiments were performed in a car and with 5 rounds, one only *2G* activated, 2nd *3G*, 3rd *WiFi*, 4th *GPS*, 5th all together. The results can be seen in tables

7.7 and 7.8:

7.2.1. Google Accuracy

Google	Rural				Urban			
Accuracy [m]	Estimate	Low Bound	High Bound	Measure	Estimate	Low Bound	High Bound	Measure
2G	2183.0	2058.1	2307.9	2134.0	698.1	573.5	822.6	950.0
3G	1469.1	1269.6	1668.7	1399.0	283.0	32.8	533.2	42.0
WIFI	52.7	38.1	67.3	42.0	34.8	33.0	36.6	7.0

Table 7.7: Results on Accuracy on 6 points sample. This table is divided into two sections. On the left it shows the predictions on three points in rural area and on the right it shows the predictions of three points in urban area. The points were selected randomly.
For the first point the mobile phone only had 2G connection, in the second only 3G, and in the third, WiFi was activated. The predictions for accuracy were calculated with the corresponding models. For each part of the table, the first column displays the estimated value of the accuracy, using the linear model. Second and third column show the lower and upper limits of the 95% confidence interval. The last column shows the real Accuracy provided by Google

In rural environment, accuracies are significantly less precise than in urban environment. We have 3 cases:

2G Predicted accuracies figures in 2G are greater (less precise) than in urban environment. The difference is 2 km in rural versus 700 m in urban. Taking a look at the Google error (distance between GPS point and Google registered point), we can see it is smaller in rural (650 m) environment than in urban (1025 m). This result can be a result of the few samples in urban environment though. The accuracy model provides reasonable confidence intervals (± 150 m for an accuracy of 2150 m) and the predicted value is inside the model provided range. On the other hand, in the urban environment 2G gives also a reduced confidence interval but it does not predict the correct value. The estimation is smaller than the measured value.

3G 3G experiment provides accuracies figures smaller (more precise) than 2G. Confidence intervals are reasonable and they are usually inside the limits too. Confidence intervals are around 150 m in an average of 1500 m meter accuracy. In urban environment, predicted accuracy figures are much smaller. It reduces from 1469 m in rural to 283 m in urban. Looking at the confidence interval, the model shows a too large range in urban. It goes from 30 m to 500 m, so it is not impressive that the model predicts the accuracy value inside. The model is accurate, but not very precise. [1] The low values of provided accuracy for 3G in urban environment (and the low errors too) may be because there are more 3G towers in the city, with better location services. for example, there are directional antennas which can assist to calculate locations based on Angle of Arrival.

WiFi Measured accuracy in Wi-Fi both in urban and rural areas are small. It is around 42 m in rural, 7 m in urban in this sample). The model provides reasonable intervals but it does not get the predicted value right in urban. The values are much higher. With these results is evident that for predicting WiFi other regressors are necessary (for example, power or distance to WiFi Access Points instead of telephony towers).

7.2.2. Google Error

Google error is calculated as the distance between each point provided by Google and the point registered by the GPS device at the same moment. As Google Timeline doesn't record the positions at every moment, it is impossible to have reads for both systems (Google and GPS device) at he same time. So a 7 seconds tolerance is taken to look for the best match. Having a look at table 7.8 we can make some annotations.

[1] Accuracy is the proximity of measurement results to the true value; precision, the repeatability, or reproducibility of the measurement

Google	Rural				Urban			
Error [m]	Estimate	Low Bound	High Bound	Measure	Estimate	Low Bound	High Bound	Measure
2G	1321.5	937.6	1705.5	650.4	1301.8	635.5	1968.1	1025.0
3G	1324.7	1013.2	1636.2	1309.8	**-295.1**	0.0	187.2	155.5
WIFI	211.1	0.0	588.3	64.7	36.2	0.0	283.8	19.6

Table 7.8: Results on Error on 6 points sample. This table has the same distribution as table 7.7 and corresponds to the same six random points. The difference is that the magnitude estimated by the model is the error Google Timeline had, and the measurements (fourth column for each section) are calculated with the information provided by the GPS device

2G As seen in the previous section, Google location timeline in 2G has less error in rural environment than urban. Our model predicts better accuracy in urban than in rural. But when it comes to error (distance from the Google point to the same point that the GPS gives for the same time) the model in the rural environment gives worse (greater) errors than in urban. The model gives a more correct estimation of this error for Urban environment.

3G Error is smaller in urban territory than in rural (around 8 times smaller). This is probably due that are more 3G cell towers in towns and city than in the country, and better capabilities, like directional transmission. The same tower has several identifications (CIDs) and each one transmits in certain angle. With this information calculations based on Angle of Arrival are possible and the location is more precise and accurate. See section 2.2.5.
To wrap up, our model is right in both cases (urban and rural). In rural it gives a reasonable interval. In urban environment, lower confidence interval bound is negative, which is an absurd result. (Zero is taken instead). Also a negative error prediction is obtained (absurd result) which has been marked in boldface in the table
Real error is smaller in rural environment than in urban. Our model gives a narrower confidence interval in urban model but it does not get it right. It is more precise but less accurate. In rural it is less precise but more accurate.

WiFi It happens the same phenomenon than in 2G, the prediction is inside the confidence interval but this interval is way too big. It is accurate but with terrible precision. Errors are small both in rural (60 m) and urban (20 m).

7.3. Transportation

Most of the experiments were performed in urban environment and what really affects Google's sensitivity is the speed we are moving. That's the reason another variable has been defined. Its name is *Action* and it defines the means of transportation. Google, in its files stores an action which is related to the velocity (Google measures the speed we move). Although this information is available in Google files, it has not taken into account The personal logbook has been used instead. In table 7.9 some results can be seen.
In this part accuracy provided by Google and its error are compared. For each way of positioning and means of transportation three random samples are taken.

7.3.1. Google Accuracy

Google Accuracy [m]	2G			3G			WIFI		
	Low	High	Measure	Low	High	Meas.	Low	High	Measure
Bike	699.5	847.3	845.0	556.3	746.2	21.0	31.5	40.6	20.0
	610.2	743.2	845.0	827.2	1161.2	20.0	26.7	34.4	29.0
	699.3	847.3	830.0	823.3	1115.8	899.0	26.7	34.4	36.0
Car	573.5	822.6	950.0	589.0	828.9	19.0	19.0	39.1	20.0
	242.1	482.4	21.0	968.2	1415.7	1571.0	31.0	53.3	23.0
	108.4	371.4	54.0	644.9	968.8	21.0	29.4	48.3	50.0
Still	1199.2	1300.4	1254.0	502.2	677.7	20.0	36.8	39.2	19.0
	567.6	808.9	20.0	502.2	677.7	20.0	36.8	39.2	19.0
	567.6	808.9	20.0	57.5	185.7	50.0	36.8	39.2	19.0
Tram	1125.4	1295.1	1247.0	704.6	944.2	765.0	36.1	48.7	19.0
	1051.6	1219.0	1000.0	791.8	992.3	1456.0	48.0	56.8	50.0
	1051.6	1219.0	1000.0	718.4	1070.8	721.0	25.8	32.4	20.0

Table 7.9: Results on accuracy on 36 points sample. This table is divided into three vertical sections and four horizontal sections. The first vertical section corresponds to predictions on accuracy calculated for experiments when only 2G was activated. The second vertical sections is for predictions when only 3G was activated, the third vertical section corresponds to predictions when WIFI was activated. The four horizontal sections correspond to predictions on accuracy for experiments in different means of transportation (Bike, Car, Still and Tram). For each vertical section there are three columns. The first and second column show the lower and upper limits of the 95% confidence interval given by our model for the accuracy. The third column is the Accuracy provided by Google

2G When positioning in 2G and we circulate by bicycle, accuracies provided by Google are quite wide (around 800 m). Our model is right with quality margins (margins narrow and right, both precise and accurate). In tram, provided accuracies have greater values (less precise), rounding the 1100, 1200 meters. Nevertheless, in still mode the samples are very dissimilar. It looks that the accuracy figure is small and our model predicts higher values than expected. It is precise but not accurate. When we travel by car the measures are very different (from 21 m to 950 m). The model predicts smaller values for smaller measures, but in the three cases the predicted intervals don't include the real measure.

3G Model predicts reasonable margins in every case but is not very accurate. The better prediction is tram because the interval contains the real measure in 2 out of 3 cases. In Car and Still the model fails in the three samples. Our model estimates accuracies bigger than what Google provides. In 3G, it doesn't look that the action taken has any significance. Intervals are similar. In Still the measured value is not in the confidence interval, but sometimes in the bicycle it is right.

WiFi Small accuracies figures in Wi-Fi make sense. They become bigger when velocity rises. The model gives narrow ranges and sometimes it is right, but it is too aleatory. Given that the accuracy variation intervals very small (between 20 m and 50 m), and the constant model (without regressors) is the one that approaches reality similar to this model.

7.3.2. Google Error Table

Results are shown in table 7.10:

2G The error lower limits of confidence intervals become negative. Negative error has no sense, so these values had to be replaced by zero. the cause for this can be a low number of samples and the existence of outliers. In tram the errors are 700 m and our model predicts them in two of the cases. Margins are acceptable. In Still and Car our models give ridiculous wide margins.

3G Still mode margin are very wide (from 0 to 3700 m). It is right in the three samples (it is accurate but not precise).

Wi-Fi In Still the model gives the same confidence interval for the three samples, but the real measures are out of the intervals: two measures below and the other above. In Bike and Car the intervals had negative lower bounds (and were replaced by zeros). The intervals are very wide and include all the real measures. In tram one of the samples looks like an outlier: 2098 m, much higher than the other two, 564 m and 61 m, but it is a case of Google Stuck. See figure 7.2.

Google	2G			3G			WIFI		
Error [m]	Low	High	Measure	Low	High	Meas	Low	High	Measure
Bike	1358.7	2428.1	2353.1	659.0	1287.4	91.3	0.0	191.4	109.8
	0.0	1578.9	640.2	0.0	526.6	121.6	0.0	351.8	32.4
	1370.6	2456.9	2995.9	0.0	1111.0	2760.3	0.0	351.8	52.3
Car	635.5	1968.1	2257.7	647.9	1591.5	24.2	0.0	547.1	47.5
	0.0	730.7	48.1	582.8	1321.3	2086.5	0.0	521.6	15.3
	0.0	688.9	338.7	587.1	1441.7	13.5	0.0	441.8	276.9
Still	0.0	1817.2	595.1	0.0	3719.2	179.3	314.1	395.3	64.5
	0.0	1011.8	538.7	0.0	3719.2	573.9	314.1	395.3	609.3
	0.0	1011.8	21.4	0.0	2287.1	122.4	314.1	395.3	179.3
Tram	367.4	1122.7	605.3	0.0	484.5	169.8	561.5	1005.4	563.7
	654.4	1485.5	922.0	1450.9	2324.0	2455.4	358.4	681.8	2098.7
	654.4	1485.5	2872.0	122.4	974.0	594.5	112.6	399.8	60.9

Table 7.10: Results on error on 36 points sample. this table is divided into three vertical sections and four horizontal sections. The meaning of these divisions are the same as in table 7.9 .
For each vertical section there are three columns. The first and second columns show the lower and upper bounds of the 95% confidence interval given by our model for Google error. The third column is the real error between the location provided by Google compared to the real position provided by GPS device.

7.4. Prediction models discussion

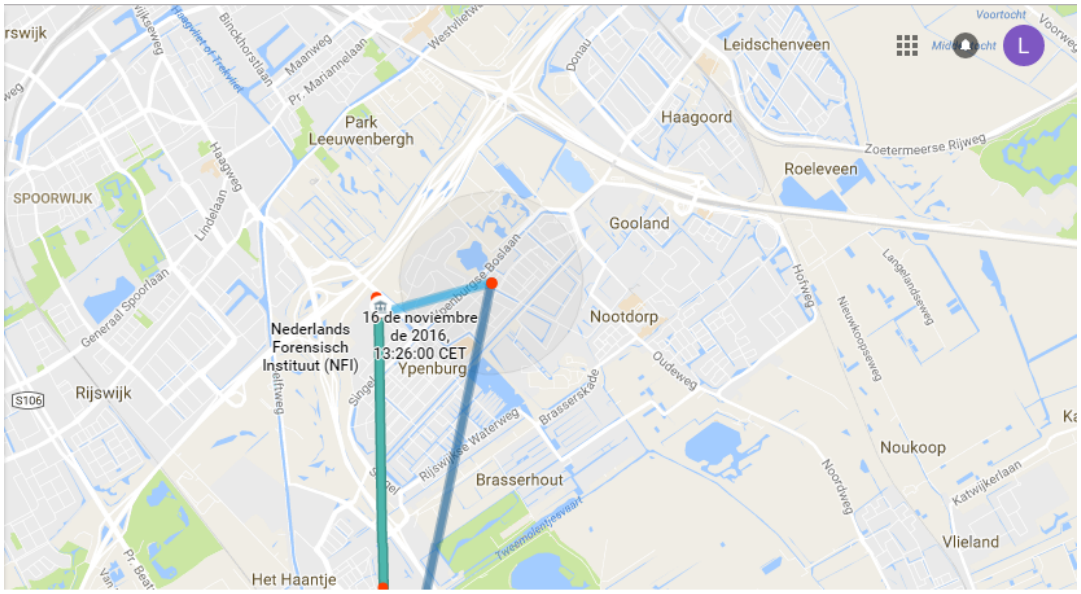


Figure 7.1: Same point, multiple "Timestamps". This figure is what Google Timeline shows for a particular event. Figure 7.2 represents the same event taking the information from the *.json* file, provided by Google. In that figure we can see many timestamps for the same point, and in this figure they do not appear. Google has its methods to remove outliers for not showing in the web.

As observed in tables, and figures of Google Stuck (see figures 7.1 and 7.2), another was extracted where this phenomenon does not occur. A new set of 36 samples has been selected to study the model's performance. With these data all the predictions and measurements are registered in tables 7.11 and 7.12.

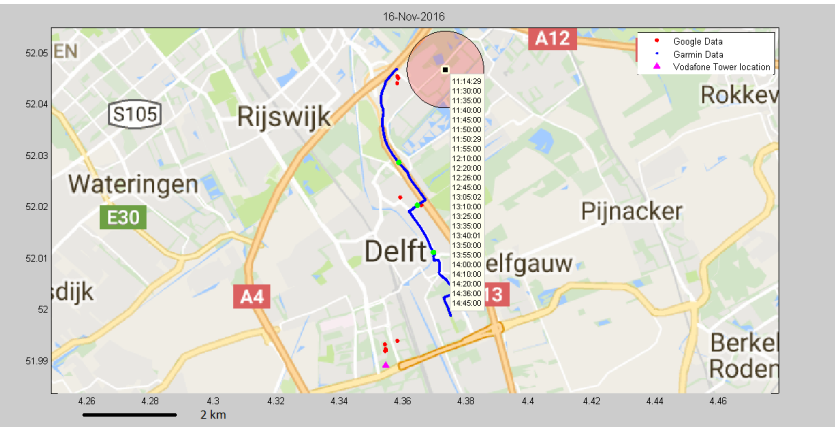


Figure 7.2: Same point, Multiple "Timestamps". The green dots (real positions) are far from the center of the circle (Google position). this point has a lot of Timestamps, this means that Google says that the phone was in that position many times or during a long period of time. But this point is far from the trajectory. It look like Google got stuck in that position.

7.4.1. New data for Google Accuracy

Google Accuracy [m]	2G			3G			WIFI		
	Low	High	Meas	Low	High	Meas	Low	High	Meas
Bike	610.2	743.2	845.0	817.8	1183.5	405.0	30.1	38.2	72.0
	610.2	743.2	845.0	809.3	1052.8	1016.0	18.6	30.7	32.0
	931.5	1076.5	44.0	556.3	746.2	21.0	26.7	34.4	36.0
Car	0.0	271.5	63.0	1112.2	1662.2	20.0	30.9	50.1	45.0
	318.8	610.2	59.0	0.0	357.8	1571.0	30.3	49.4	20.0
	621.0	883.8	1347.0	0.0	302.8	50.0	22.7	53.1	20.0
Still	1199.2	1300.4	1254.0	523.8	629.6	100.0	27.6	32.5	50.0
	730.1	923.7	829.0	485.6	666.8	131.0	27.6	32.5	21.0
	569.6	810.6	171.0	0.0	107.4	28.0	37.7	40.7	83.0
Tram	1125.4	1295.1	1247.0	788.3	1013.4	1000.0	10.7	21.2	24.0
	1051.6	1219.0	1000.0	672.1	1028.1	292.0	46.3	56.0	50.0
	988.6	1210.5	205.0	718.4	1070.8	721.0	19.9	29.7	19.0

Table 7.11: Results on Accuracy. This table is similar to table 7.9 but the predictions are calculated on other 36 samples

As explained before, another sample of 36 point has been retrieved in order to study accuracies and errors. The predictions for these samples, and the real measurements are shown in tables 7.11 and 7.12.
It can be observed that the accuracy estimations when circulating by bicycle have acceptable ranges with 2G. The model is wrong in 3 out of the 3 cases. The case 3 is show in figure 7.3. This case is really strange, because Google gave a very good acuracy (44 m) and the value was right (real location at 26.7 m shown in table 7.12).

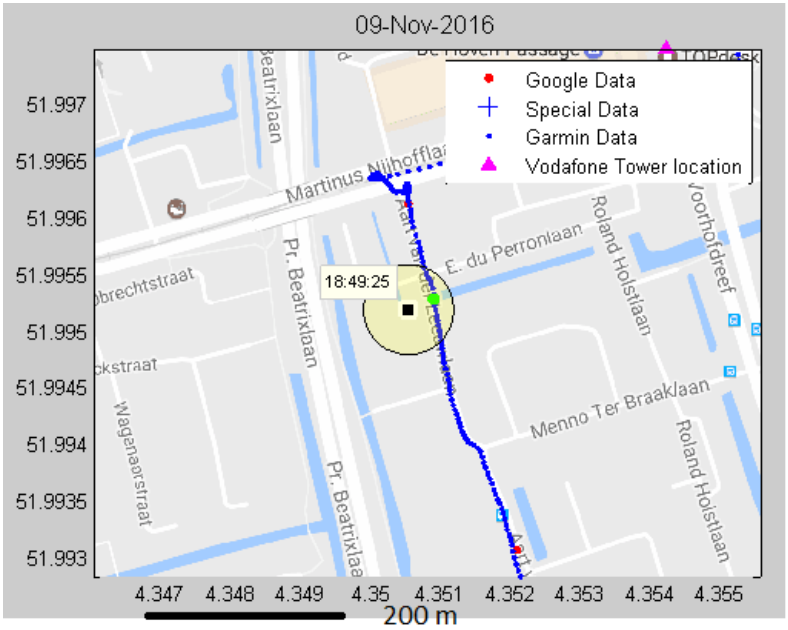


Figure 7.3: Location of sample 3 on Bike, for model 2G

Figure 7.3 shows the third point for Bike, with 2G connection in tables 7.11 and 7.12. For this point Accuracy is 42 m and Google Error 26.7 m. The green point represents the real location, which is inside the circle, which represents the position provided by Google and its accuracy. For this point our models predicted an accuracy between 931 and 1076 m, and an error less than 1154 m. The confidence interval for accuracy is reasonable and for error is very high. The measures were much better than expected.

On the other hand, when circulating by car the model 2G is right only when the range had negative lower bound (first sample, 63 m of accuracy, when predicted was 271 m or less).
The second sample when traveling by car gives very good accuracy (59 m), better than the predicted by the model (between 318 and 610 m) and the error agrees the accuracy (46 m). The model for error has a very wide margin (less than 772 m). The position and accuracy of this sample is shown in figure 7.4. For this point Accuracy is 59 m, and Google Error is 47 m (see tables 7.11 and 7.12). For this point our models predicted an accuracy between 319 and 610 m, and an error less than 771 m.
The confidence interval for accuracy is reasonable, but the model didn't do a good prediction and for error is very high. The measures were much better than expected.

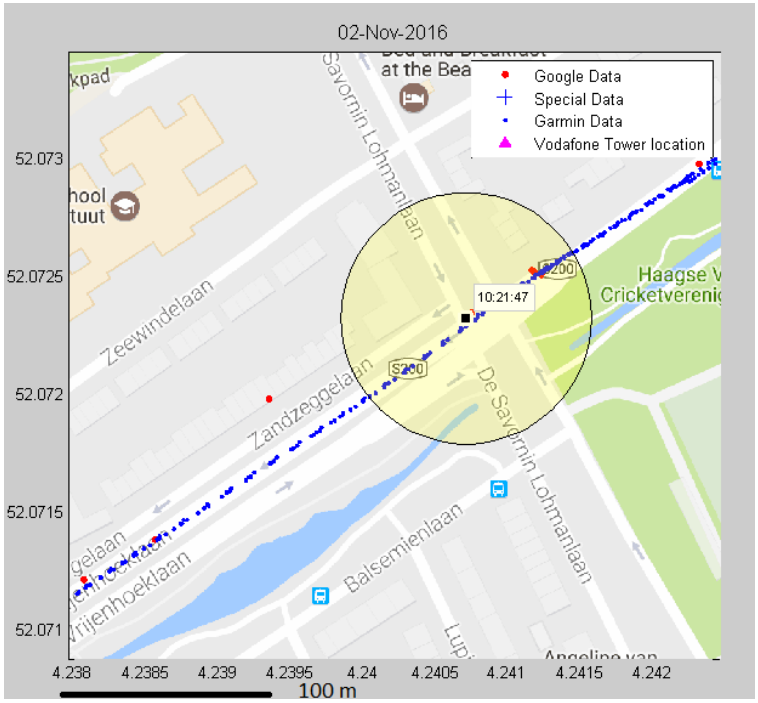


Figure 7.4: Location of sample 2 on Car, for model 2G

In Still mode, the model is right for high accuracy values (more than 500 m) and margins are reasonable, but fails in the third sample (It has an accuracy of 171 m and predicted was between 597 m and 811 m).

When the transportation is tram the model always gives high values of accuracy with reasonable margins. Both models (Accuracy and Error) agree with the measured values in two out of the three cases.
The exception, third case, is shown in figure 7.5. In this case, both models gave predicted values higher than real ones: Google Accuracy between 989 and 1211 m, and an Google Error between 562 and 2562 m. The predicted error was greater than the rest of the cases (between 562 and 2562 m) and the real value was very accurate (only 29 m of distance to the real position). The accuracy was 205 m, very small compared to the rest of the Tram samples (1000 m

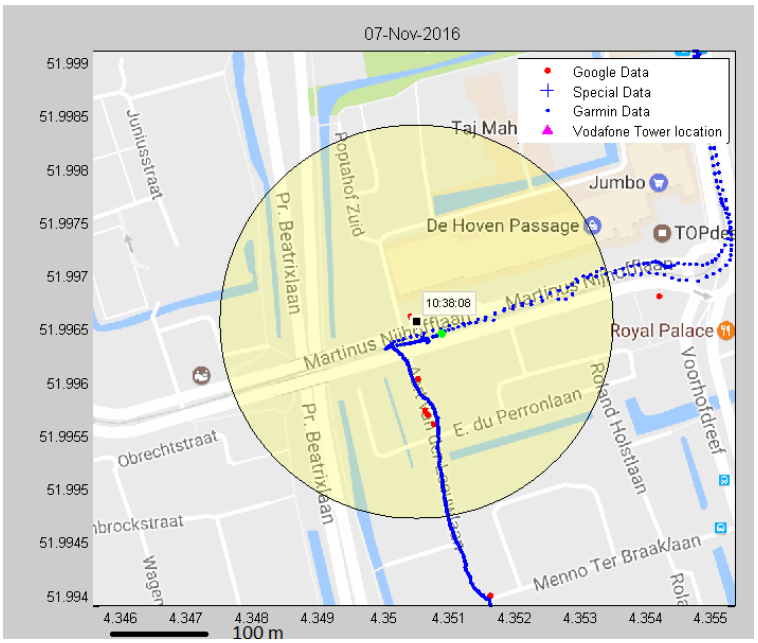


Figure 7.5: Location of sample 3 on Tram, for model 2G. The green dot inside the circle, indicates the real position at that moment.

and 1247 m).
The confidence interval for accuracy is reasonable, but the model didn't do a good prediction. For Google Error, the model gave a very wide interval, and the real measure was much lower. The measures were much better than expected, like the other cases, and our models were not able to predict them.

For Still and Tram, Google and our model seem to behave similar.

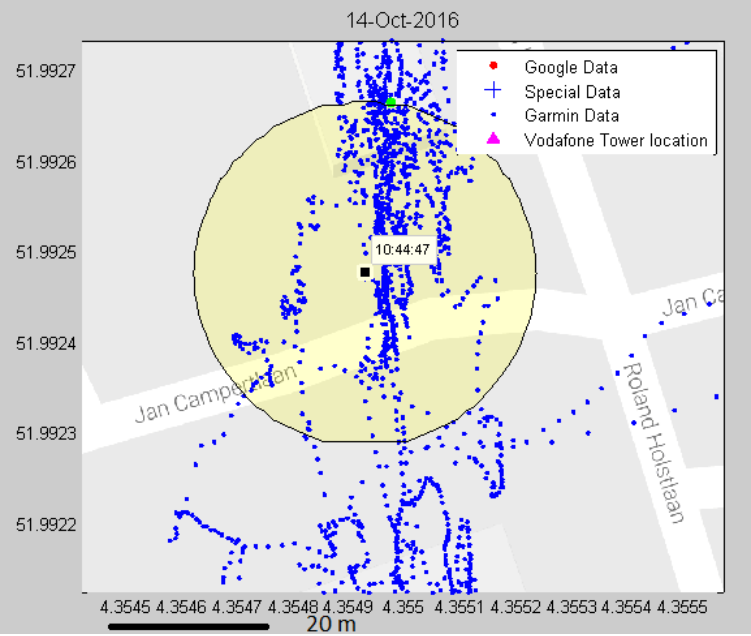


Figure 7.6: Location of sample 3 on Bike, for model 3G,

In figure 7.6 the location for the third sample of Bike, for models 3G (accuracy and error) appears. The accuracy provided by Google and the real error were 21 m. Our models predicted less accurate measures. Looking at the great quantity of GPS locations and this is the only one which has a Google location confirms that this case is an exception. For this point our models predicted an accuracy between 556 and 756 m, and an error between 659 and 1287 m. The confidence interval for accuracy is reasonable, but the model didn't do a good prediction. For Google Error, the model gave a wide interval, and the real measure was much lower. The measures were much better than expected, like the other cases, and our models were not able to predict them. The green dot in the border of the circle, indicates the real position at that moment. The figures for Accuracy and Error indicate that Google knew the real position at that time, but, as seen in the figure a lot of blue dots appear and no other Google record (red point) is in the scene. This indicates that this "good point" is only an exception or that when there is not movement, Google doesn't insert locations in its database.

In figure 7.7 one can see a very big accuracy radius for a normal location inside the trajectory. This kind of measures can not be predicted by our models. It is 1571 m for Accuracy and 1054 m for Google Error in this sample and 20 m and 50 m (for Accuracy) in the other two samples. Apparently it should be similar to the rest of the Google locations registered in that trip, but it isn't.
For this point our models predicted a Google Accuracy less than 358 m, and a Google Error between 154 and 569 m. The confidence interval for Accuracy had a negative low limit, which has no sense, and zero is adopted as lower bound for the interval.
The model didn't do a good prediction for Google Accuracy. For Google Error, the model gave a reasonable interval, but the real measure was much higher. The measures were much worse than expected, and our models were not able to predict them.
The rest of red dots (Google locations) do not have such a big radius. This location is very near to the trajectory so the big radius is caused by unknown circumstances in Google's calculation algorithms. Maybe that point is in a radio shadow zone (an area where there is signal obstruction) and location is estimated by the previous ones (and velocity) but in that moment there was no connection.

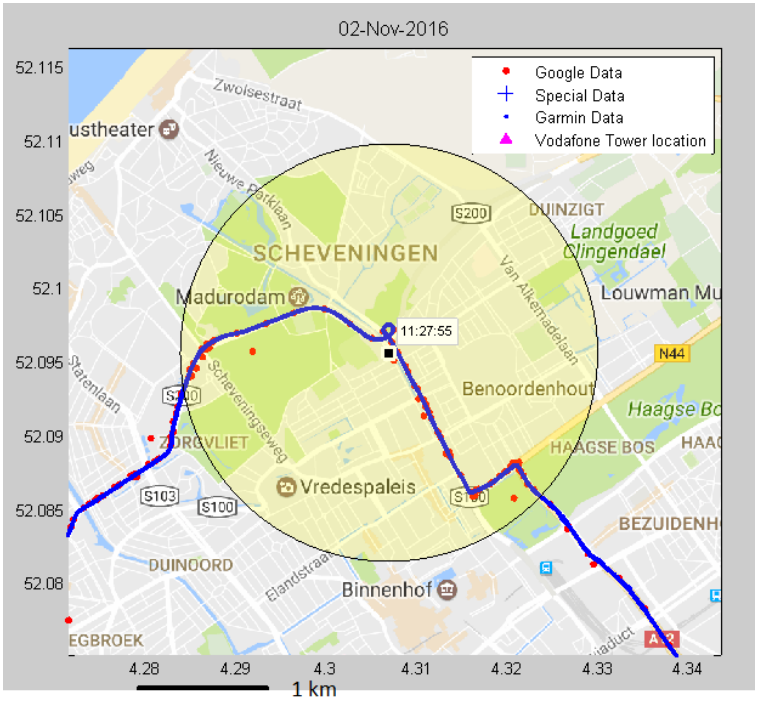


Figure 7.7: Location of sample 2 on Car, for model 3G, in tables 7.11 and 7.12.

In figure 7.8 the location of second sample on Tram, for the 3G model. In this figure it can be noted that the position given by Google is near the tram stop, but at that moment the trip had already started. Maybe the delays on calculation or lack of synchronism between GPS device and Google server can cause this shift in position. These delays can not be predicted by our model.

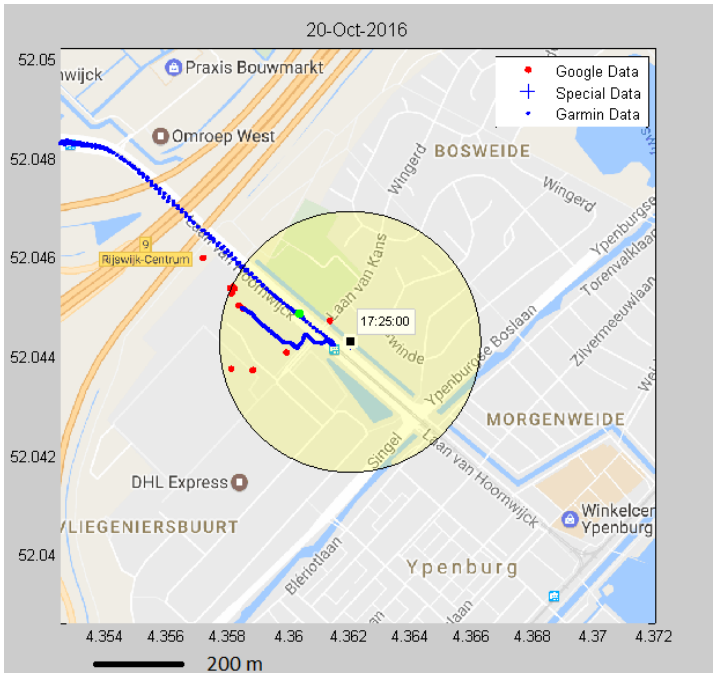


Figure 7.8: Location of sample 2 on Tram, for model 3G. Accuracy = 292 m, Error = 131 m.

For this point our models predicted a Google Accuracy between 672 m and 1028 m (and Google provided 292 m), and a Google Error less than 875 m (and the measured error was 131 m). The confidence interval for accuracy is almost reasonable (a bit wide), but the model didn't do a good prediction.

For Google Error, the model gave a wide interval, (with negative lower limit), and the real measure was inside the range. The measures seem to have been taken near the tram station, but at that time, the travel had already begun. It seems that Google's delays in calculations affects to the position provided.

Our models were not able to predict these delays. The green dot inside the circle, indicates the actual position

at that moment.

When Google Accuracy measures are done with 3G, the model gives reasonable margins except in extreme cases with very high accuracies (more than 1500 m by car) or very low (28 m when Still or 21 m in Bike).

In these experiments the inferior margin that the model gives is negative and it is used zero instead. The model is right in the majority of the cases, or it gives accuracy values above the average.

The model which estimates Google Accuracy from the WiFi data gives some acceptable margins (none of them negative) and they are close to the registered value given by Google. This value is not correlated with the regressors used, but there is so little variance that any constant model close to the variable response Y can seem valid even when it is not.

7.4.2. New data for Google Error

For the positioning error committed by Google some models have been elaborated in order to estimate it. Observing the values obtained with the same sample as the former section it can be observed that many confidence intervals have negative limits, specially when 2G data is taken.

Google Error [m]	2G			3G			WIFI		
	Low	High	Meas	Low	High	Meas	Low	High	Meas
Bike	0.0	1578.9	665.3	46.2	1544.4	300.9	220.4	929.1	45.4
	0.0	1578.9	640.2	0.0	1039.6	184.0	132.5	493.0	14.5
	0.0	1154.1	26.7	659.0	1287.4	21.1	0.0	351.8	14.9
Car	0.0	825.3	45.2	784.5	2030.1	11.7	29.8	688.4	39.8
	0.0	771.6	46.6	154.4	569.0	1054.1	64.5	704.9	17.5
	306.1	1164.9	799.9	0.0	696.9	15.1	56.7	619.8	26.1
Still	0.0	1817.2	591.5	0.0	911.4	73.1	0.0	207.4	12.9
	640.8	1931.6	909.9	0.0	3747.9	108.9	0.0	207.4	16.5
	0.0	1020.1	24.5	0.0	2476.1	20.4	136.9	349.7	39.2
Tram	367.4	1122.7	539.7	711.7	1175.5	908.4	322.6	586.6	31.0
	654.4	1485.5	922.0	0.0	875.3	130.8	269.8	537.1	36.3
	562.1	2561.9	29.0	122.4	974.0	538.1	90.3	571.4	17.7

Table 7.12: Results on error on selected 36 points sample This table is similar to table 7.10 but the predictions are calculated on the same 36 samples as table 7.11.

With 2G data, the only limits that maintain themselves positive in all three samples is when traveling by tram. The margins are quite wide and it is right only in two out of the three cases.
In the rest of transportation means, the model is right, but the confidence interval are too wide.
In 3G model, the margins of the confidence intervals are also wide, specially in Still. In the tram the model is right in all three cases. The confidence intervals look more uniform in bike than in car. Positioning errors in Google present great dispersion.

The model to predict the error committed by Google when Wi-Fi is connected differs from the former one that computes accuracy.
To compute the error the confidence interval are wider and even with that, the real measures are not within them. Only in Still it is right two out of the three cases,again with wide margins.

7.4.3. Models results summary

In order to evaluate each one of the six developed models in a global way, all the data entries (regressors and observations) hav been classified attending to the corresponding model, and for each entry the predicted value has been calculated, as well as the 95% confidence interval.
Thereafter, each observation is looked whether it falls inside its confidence interval or not. If it is inside it is considered as a *Correct Prediction*. Table 7.13 shows for each model, the number of observations, the Correct Prediction rate, prediction mean and standard deviation, and the confidence interval width (mean and stan-

dard deviation).
Models which have many Correct Prediction with loose confidence intervals are accurate but not precise.
If confidence intervals are narrow, the model is precise, i.e. model for Accuracy in WiFi, with 5.3 m, but it has few correct predictions (13%) so it is not accurate.
It is clearly perceived that models which predict Google Error are less precise than models which predict Google Accuracy, as well they also have less observations inside the predicted confidence intervals (less accurate).

Linear Model		Num Obs.	Correct Prediction [%]	Predicted value		95% Confidence interval width	
				Mean [m]	Std [m]	Mean [m]	Std [m]
Accuracy	2G	297	76	1111	351	374	167
	3G	779	23	772	388	231	154
	WiFi	2850	13	31	8.2	5.3	6.1
Error	2G	297	18	460	1056	2666	2088
	3G	842	11	956	1640	3404	2707
	WiFi	3115	6	297	211	453	622

Table 7.13: Predicted values and 95% confidence intervals

8

Conclusions

8.1. Thesis review

This thesis has been done under supervision of the Netherlands Forensic Institute (NFI) and the Delft University of Technology. The objective is to evaluate the information that Google Location Timeline provides for its possible use as court evidence.

8.2. Research questions

8.2.1. What is the actual accuracy of the location data that Google Location History provides?

Google Timeline provides a dataset with location records. Each register contains among other data, position (latitude and longitude) and a radius (accuracy). In this thesis we worked with them: position and accuracy.

Experiments have been performed with 4 different phone configurations and other parameters. The phone configurations are 2G, 3G, WiFi and GPS connections. The condition parameters are:

Environment Rural, Urban.

Mean of Transport Walking, Bike, Car, Tram, Still.

Weather Sunny, Cloudy, Rainy.

Traffic Light, Normal, Busy.

How do we quantify Google Geolocation Accuracy?

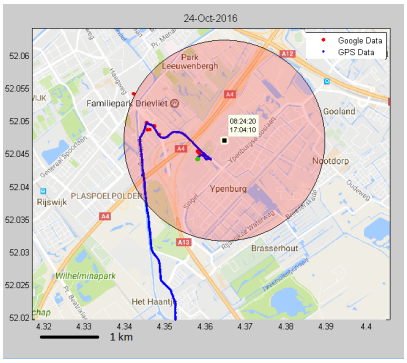
Two variables have been evaluated in this research:

Radius of the circle Stated in meters. Google provides a position estimate together with an indication of accuracy namely the radius of a circle. In this thesis we assess to what extent this radius reflects the actual position accuracy.

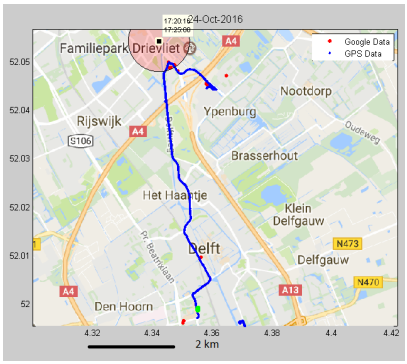
This radius is called through the thesis **Google Accuracy** and its study is deployed in chapter 6. To quantify this variable a set of experiments have been executed as explained in chapter 4.

Google true error Error that Google makes when providing a position. It has to be considered from two points of view:

1. First as a numerical value, i.e. The distance measured in meters between the position that Google provides and the true position of the device. During the thesis actual position is taken as the one a GPS device with a smaller positioning error gives.
2. Secondly we have to compare the position error with the accuracy radius given by Google, what is likely a statistical bound (eg 95% confidence) on the position error.
   A *Hit* is defined, when the actual position is within the circle given by Google, and a *Miss* if it is outside. See figures 8.1 for a graphic explanation.



(a) This is a *Hit* example, but with such a large radius (1750 m), the information of this location is useless.



(b) In this case the radius is small but the location has a big error (4800 m). It is convenient to know both that it is a miss and the position given by Google is tremendously far away from the real one.(The real position is the green dot)

Figure 8.1: Small and lage accuracy radii

Does accuracy stated by Google correspond to actual accuracy?

A summary table (table 8.1) is shown below with the experiment measures explained in chapter 4 and result tables from chapter 6.

[m - m - %]		2G			3G			WiFi			GPS		
		Accu	Error	Hits	Accu	Error	Hits	Accu	Error	Hits	Accu	Error	Hits
Environ.	Rural	1627	1430	68	1513	1710	48.5	-	-	-	50	300	40.6
	Urban	23	164	24	82	206	33.3	20	142	7.9	9	5	62.6
Action	Bike	845	1930	20	899	415	33.3	20.0	83	28.8	44	27	100
	Car	964	655	59.1	1399	975	56.9	-	-	-	9	5	67.4
	Still	20.0	120	9.9	20	170	9.5	20	156	1.6	20	395	0
	Tram	1169	880	48.9	1000	905	48.2	23	97	21.5	13	3	100

Table 8.1: Median Values provided by Google for Google Accuracy and Google Error expressed in meters and *Hit* percentage. Table is divided into four columns which correspond to data acquired with 2G signal, 3G signal, WiFi signal and GPS. The classification is done with 2 criteria. First two rows are the Environment division and the other four are the Action. For each division three statistics are shown: Google Accuracy and Google Error medians and Hit rate as a percentage.

From this table, we draw the following conclusions:

First, we observe Google's behavior regarding Environment.
We can see that for both 2G and 3G the Hit rate in rural is more realistic (68-48%) than for the urban environment (24-33%).
However, looking at the order of magnitude, we can see the interval for both, error and accuracy for the device is narrower in urban (around 20-80 m) than for the rural case, where the accuracy and error are in the order of magnitude of 1500 m. This make measures and positions taken in urban environment more useful to pinpoint where the device was at the given time.

Regarding **Wi-Fi** connection, we can see in rural environment there is no WiFi connection data and in Urban environment Google is too optimistic, giving smaller radii than the actual errors. This makes Google location by Wi-Fi not reliable.

It is very strange that the localization by **GPS** in rural environment gives worse result than in urban surroundings. With the phone model used in the experiment, it is impossible to deactivate the radio completely leaving the GPS active. So it could be that the location was actually using Cell tower and not GPS. The large error figures are due to non synchronism (in time) since the locations appear on the same path, but at a different time.

Secondly, we regard the means of transport:

In **2G and 3G** the best results are given by *Car* and they are better in 2G than in 3G. However, with *Still*, the accuracies are small, and although *Hit* rate is low, Errors are also small (about 100 m), so Google is not optimistic on its own predictions under these circumstances.
Nonetheless, using **Wifi** the worst results are at *Still*. This may be because Google likely uses the fingerprinting method with visible networks at any given time. Within a building only the networks of the building itself are visible, and all coincide in position. So Google gives a fixed position when it is inside the building, and the size of the building determines the error. When the device is out of the buildings in the open air, the accuracies that Google gives are very optimistic, and therefore it has low *Hit* rate. However, the errors are not large (less than 100 m).
When **GPS** is used, the location is good except for *Still*, which coincides with being inside a building, which is a very logical result because there are no visible satellites. Under these circumstances, Google probably calculates the position based on telephony networks

Examining these results, Google is not totally reliable. Viewing the percentages of hits, none exceeds 70% except when using the GPS that is usually deactivated in mobile phones. But observing the numerical values, although Google does not succeed, the errors are of the same order of magnitude as the accuracies. So even if the phone is not in the circle that Google provides in its Timeline, normally the error is not that big to deny that at least it has been in the whereabouts.

Maybe Google computes its own position with a certain σ. To make sure that 100% of the results would be inside the accuracy circle, then the radius should be ∞ , but this information would have little use.
So, maybe what Google does is computing a σ from its own location algorithm and base its radius on that. Using one σ would take 68% of hits. 2 σ 95% and 3 σ 99.7%.
In the case of 2G and 3G, one σ seems to be quite close to this line of thinking. On the other hand, both WiFi and GPS hits ratio is far too off of the one σ value.

Is there the possibility of doing reverse engineering to determine how Google computes the accuracy?

Google is a black box to us and we do not know how it does its calculations. However using the results of the experiments performed in the chapter 4 we are able to determine some of the parameters that affect the results.

These parameters are:

Network connection 2G, 3G, WiFi, GPS

Environment Rural, Urban.

Means of Transport Walking, Bike, Car, Tram, Still.

At the same time we discovered that *Weather* and *Traffic* do not affect the performance.

In case the phone is connected to the 2G network, we have both the information of the Cell Tower Tower that the phone is connected to and the neighboring towers. Thanks to this information, it is possible to implement a localization calculation based on power strength ourselves. Comparing the errors we made computing this method with the data available and those of Google, it is discovered that both are of the same order of magnitude. This suggests that Google could in fact use a similar method to determine the location and accuracy radius.

These results are developed and shown in the chapter 6.

Where does Google take the information from?

As answered in the previous questions, the accuracy and error of Google are very dependent on the four possible phone configurations (2G, 3G, WiFi, GPS). When the connection is 2G, we can suppose that Google determines the device location with RSS, basing this supposition on our own calculations explained in chapter 6

When 3G is activated, we can only access to the information of the signal strength of the neighboring towers, but not to their Cell IDs. We can only have access to the complete information about the tower the

phone is connected to, which is not enough to perform position computations on our own. However, given that the order of magnitude of both accuracy and error is similar to the one found in 2G, we could also assume that Google has somehow access to the complete information on the neighboring towers and perform similar computations as 2G configuration.

When using WiFi networks it probably uses fingerprinting methods. While Google vehicles take the Streetview information (war-cars), they take at the same time a fingerprint of available WiFi networks and strengths. This information is compared to the one registered by a mobile device and location is based on best matches. It is possible that these fingerprinting methods are applied by Google with telephony networks too.
It was detected in our experiments that WiFi configuration gives better results outdoors than indoors (still). This is because when the phone is outside it has several Wi-Fi networks in sight and fingerprinting can be used [17]. When the phone is inside only the networks of the building are in available and therefore always gives a single location.

It is obvious that when GPS signal is active in the mobile device, Google uses it and with the best results.

How and when does Google store/compute the locations and send them to the server?

To know exactly what triggers Google to either store or upload one or several locations, a Logfile is looked into. This file is called Logcat and can be obtained from a rooted phone.

This file has logs of two kind of relevant events:

Insert When one or several locations are registered "Successfully inserted *x* locations"

Upload When the information is uploaded to the server "Upload task finished".

It is assumed that the insertions are done when an application requires location, like Google Maps, or when a WiFi connection is available. Perhaps only when Internet access is required (and available) is the moment the mobile device uploads the locations. This line of investigation is left for future work.

8.2.2. Is it possible to perform a prediction of the accuracy radius and error that Google will provide in case there is new experiment are performed?

In this research we developed several linear models to estimate and predict Google provided accuracy and Google error, based on the experiments executed and data recorded by Google and our own location devices. In the case there is a new phone with evidence in it, we would like to know the error really committed by Google, and this way we would be able to do an estimation of the real location of the mobile using the data stored and provided by Google. In this new case there is obviously no *Ground truth* data available, we would calculate the location and error with a statistical method.

What information can be extracted from the phone?

From the phone two logcats were extracted:

Normal logcat From the normal logcat we could extract actions that were not evaluated in this research. The texts with higher number of repetitions surrounding "location inserted" were scanned. This information could give a clue to determine which actions trigger the phone to insert its location information in the Google application. No conclusive results were obtained.

Radio logcat From the Radio logcat, cell towers ID and signal strengths can be retrieved. This information was used to determine that Google uses a RSS (Radio Signal Strength) method to determine mobile position.

In order to obtain these files the phone has to be configured beforehand. If a phone is received as evidence for a new case, these files are not available. That is why the models developed in this research do not need these files. They use:

Google Timeline location data The *json* file downloaded from Google Timeline application.

Cell Tower position database The Vodafone Tower location database, provided by NFI. It contains all the Vodaphone telephony towers in the Netherlands including their locations and identifications.

With this new information we would be able to calculate the new input for the linear models.

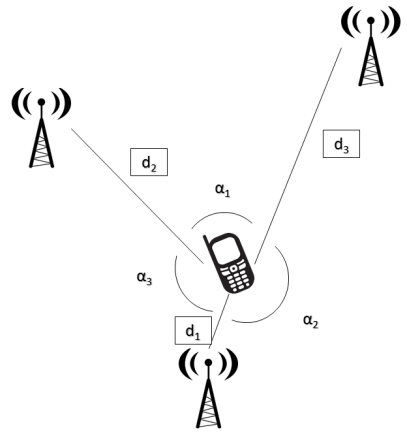


Figure 8.2: Input for linear model: distances d_1, d_2, d_3 and angles α_1, α_2 and α_3.

Models	Google Accuracy	Google Error
2G	model 1	model 4
3G	model 2	model 5
WiFi	model 3	model 6

Distances Distances expressed in meters from Google provided locations to the three nearest base stations.

Angles Angles expressed in radians from Google provided locations to the three nearest cell towers.

For an illustration of the input for the linear model, see figure 8.2.

So, if a phone in a new case has to be investigated as evidence, and we want to apply the linear models developed in this research, only the Google Timeline *json* file and the Vodafone Cell Tower database provided by NFI are necessary.
Another source of information could be obtained from the Telecom Company. The cell towers the phone has been connected to are registered by the Telecom Company and can be used for investigation purposes [35]. This is the CDR (Call Detailed Record). .

With this information, can an algorithm (or several ones) be used to deduce its previous locations?

The linear models developed use as input the information described in previous paragraph. The training data for these models were the experiments performed for the first part of this thesis. These experiments are described in chapter 4. These experiments include the following data:

Network connection 2G, 3G, WiFi, GPS

Environment Rural, Urban.

Means of Transport Walking, Bike, Car, Tram, Still.

Six linear models were developed: three for predicting Google Accuracy and three for predicting Google Error. No models were developed for GPS connection because it was confirmed by the experiments that the variables under this study are independent from available regressors.

Once the model has been developed, is it good enough to be considered accurate?

These models were checked with k-fold method. This method consists in dividing available data in k folds. One fold is used as test data and the other $k-1$ as training data. The process is repeated k times so that each observation is used once as test-data.
It is evaluated whether the actual measurement of the test data is within the predicted 95% confidence interval. With the *Hits* percentage, the predicted values and the amplitudes of the confidence intervals, table 7.13 is elaborated.

With these results we can assure that the best model is Google Accuracy for 2G, with a hit rate of 76%. This model presents a large accuracy mean (1111 m) and a confidence margin of 374 m. The model for Google Accuracy for 3G has a success rate of 23%. The predicted Accuracies are smaller and the confidence intervals too.
The other models have few successes and yield only low confidence.
These results indicate that the developed models for WiFi are not adequate for predicting accuracy and error. However this is a prototype model to put the idea of prediction into practice. With a better tuning and parameter election it could develop into models which provide better predictions and narrower confidence intervals.

8.3. Future research

8.3.1. Recommendations about collecting data

For future research in this field, I would suggest to obtain more data in rural area and on long distance travels. Also taking more data in public transport (train and bus) could give an idea of the influence of network has on Google performance.
Another point of interest is the behavior of mobile phones and Google when the phone is not in the SIM card operator's country. When a phone is in its home country, only connects to its own network, but when it is roaming, it may have connection access to several networks. The more cell stations available, the more accurate the position may be.
This research was done using only Vodafone telephones. Future investigations can be done with other operators (KPN) and virtual operators.
4G networks are now available everywhere. Future studies should also include this kind connection to evaluate Google's performance.

8.3.2. Recommendations about methodology

First approach that could be investigated is to change the regressors in the linear model. Signal strengths recorded on the phone logs have shown that by themselves are inefficient for a linear model. A new starting point to study could be to use less angles regressors. At this moment the three angles with the three nearest towers are used. Perhaps a new regressor based on angles could be use to substitute them.

For example:

- The difference between the greatest angle and the minimum of the three.
- The geometric mean of the three angles.
- The minimal angle.
- The greatest difference between any angle and 120 degrees.

The quantity of data provided by Google is not uniform. In some periods of time it stores a high number of locations and in others only a few ones. The linear model developed in this research takes all the data as a set. A new idea would be to reduce observations where these have a high density in time, avoiding that these time intervals have a heavy weight in the model.
Google location gives small accuracies when the phone is connected to Wifi and GPS. The models developed in this research could not predict these results using cell towers. For WiFi connection a similar work can be done using a database of WiFi networks (SSID). For sure Google has its own database and there are public databases on the web, and they have daily updates.
In this research the information extracted from the phones has been only about radio connection, cell towers and signal strengths. But the phone registers a huge quantity of information about its activity. A new interesting research to be done is to decode the phone activity using these logcats, and try to discover the actions the phone or Google perform to calculate its location.

8.4. Final conclusion

The Final Conclusion can be summarized in two ideas.

Based on the performed experiments, Google locations and their accuracies should not be used in a definite way to determine the location of a mobile device, however, although Google does not succeed, the errors are of the same order of magnitude as the accuracies. So even if the phone is not in the circle that Google provides in its Timeline, normally the error is not that big to deny that at least it has been in the whereabouts.

The linear models developed in this thesis were improved adding interactions to achieve better predictions and narrower confidence intervals. Even that, the results were not satisfactory enough yet. Further research in the parameters involved and a major collection of data is required.

The linear model is the first step to begin a Big Data Analysis system, and it will surely need more input than the gathered in this research.

A

Experiment data collection

A.1. Collect data from Google Location Timeline

The tools used for this section can be found in the zip file. They are:

Program names *JSON_reader1.m* and *JSON_reader2.m*. During the experiments two mobile phones were used,and each of them has a different Google Timeline, so the program has been duplicated. *JSON_reader1.m* reads and translates the timeline history of phone number one to Matlab format, and *JSON_reader2.m* reads and translates the timeline history of phone 2 to Matlab format also. They are in different folders, and each of them only reads the JSON file contained in its folder.

Input *json* file downloaded by Google Location Timeline. To download you have to click on the button *Timeline* on Google maps webpage. See figures A.1 and A.2.

Output *JSON1.mat* and *JSON2.mat*. Matlab files that contain the information retrieved from the *json* file. For a visualization of the entire table, see figure A.3. Inside, the information is organized in columns:

TimeStamp The time stamp of the event provided by Google. It is a number which represents in Matlab notation the time in UTC (Coordinated Universal Time). The *json* file from Google gives this information with other units, but no Timezone translation is needed.

Lat Latitude provided by Google, it indicates where the device was located. Expressed in degrees.

Lon Longitude provided by Google, it indicates where the device was located. Expressed in degrees.

Accu Accuracy provided by Google. It is a radius of a circle which center corresponds to the latitude and longitude given. It represents Google's own uncertainty and indicates that the device could be anywhere in this circle at the indicated Timestamp. It is expressed in meters.

Date The information is the same as TimeStamp, but in a human readable format (day-month-year hour:minute:second)

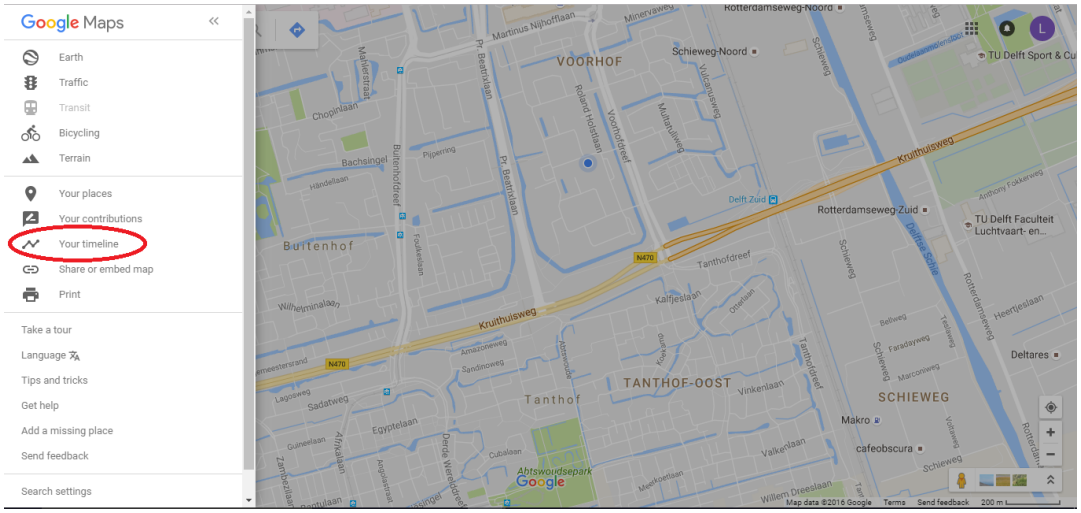


Figure A.1: Google maps webpage where you can download you timeline data if it is activated on your Google account.


```
  "accuracy" : 22
}, {
  "timestampMs" : "1470476420820",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476387607",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476026837",
  "latitudeE7" : 519923318,
  "longitudeE7" : 43543980,
  "accuracy" : 22
}, {
  "timestampMs" : "1470476005798",
```


Figure A.2: Excerpt from the JSON file downloaded form the Google test account. In it, it can be seen timestamp (ms from 1-1-1970), latitude longitude (in degrees x 10^7) and accuracy (in meters).

1 TimeStamp	2 lat	3 lon	4 accu	5 Phone	6 Glat	7 Glon	8 err_xy	9 err_x	10 err_y	11 G2	12 G3	13 WIFI	14 GPS	15 Weather	16 Traffic	17 Environment	18 Action	19 SOURCE
7.3663e+05	52.0397	4.3139	778	2	52.0081	4.3527	NaN	NaN	NaN	0	1	1	0	Clear	Normal	Urban	Still	G3
7.3663e+05	52.0454	4.3580	79	1	52.0444	4.3598	160.3208	119.1322	107.2874	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3580	79	2	52.0444	4.3598	161.0472	120.2991	107.0731	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3581	19	1	52.0444	4.3598	161.9024	117.4327	111.4552	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0454	4.3582	19	1	52.0444	4.3598	153.9548	108.8047	108.9213	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0452	4.3583	20	2	52.0444	4.3598	137.7492	99.8912	94.8513	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0449	4.3586	20	2	52.0444	4.3598	100.9037	82.9279	57.4857	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0450	4.3586	50	1	52.0444	4.3598	106.0738	82.2178	67.0221	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0467	4.3734	830	1	52.0448	4.3604	917.7567	894.7289	204.3778	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0449	4.3603	919.3845	898.3221	195.7473	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0449	4.3586	20	2	52.0449	4.3604	121.0499	120.9514	4.8847	0	1	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	52.0467	4.3734	830	1	52.0335	4.3461	2.3776e+03	1.8774e+03	1.4592e+03	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0141	4.3513	3.9264e+03	1.5215e+03	3.6199e+03	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	52.0024	4.3540	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	51.9932	4.3588	47	1	51.9932	4.3593	34.0335	33.9321	2.6259	1	0	1	0	Clear	Normal	Urban	Still	WIFI
7.3663e+05	51.9933	4.3589	4354	1	51.9919	4.3541	360.4386	327.8190	149.8462	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9917	4.3549	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9921	4.3549	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	51.9922	4.3546	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	NaN	NaN	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	52.0467	4.3734	830	1	NaN	NaN	NaN	NaN	NaN	1	0	1	0	Clear	Normal	Urban	Still	G2
7.3663e+05	51.9920	4.3546	21	2	NaN	NaN	NaN	NaN	NaN	0	1	1	0	Clear	Normal	Urban	Still	WIFI

Figure A.3: Matlab unified table. All the information is unified in this table. The entries are conformed by the Google Timeline points that are registered in the json file. First column is Timestamp, time and data in UTC when the event was registered. Second and third columns are latitude and longitude registered by GPS device respectively (in deg), considered as "Ground truth". Fourth column is accuracy, radius of the circle centered in the Google point where Google indicates the device is located at the given time. Fifth column is Phone, indicating if the point was taken from phone 1 or phone 2. Sixth and seventh columns are latitude and longitude registered by Google respectively (in deg). Columns eighth, ninth and tenth are the distances in meters from the point given by Google to the point given by GPS device, or in other words, the error Google is making. Columns eleventh, twelfth, thirteenth and fourteenth indicate what configuration was activated at the moment of the registration of the point (2G, 3G, WiFi and GPS respectively). Fifteenth column is the weather (clear, cloudy, rainy). Sixteenth column is traffic (light, normal, busy). Seventeenth column is environment, that could be Urban or Rural. Eighteenth column is source, a filter of the configuration columns (2G,3G,WiFi or GPS) that determines which was the true source of the point if two or more were activated at the same time.

A.2. Collect data from GPS device

A.2.1. Collect data from Garmin 76Csx

The tools used for this section can be found in the zip file. They are:

Program name *GPS_reader.m*. The program reads all the files with *.GPX* extension in the same folder and generates a Matlab table whose name is *GARMIN*.

Input Any file with .GPX extension provided by the GARMIN device.

Output *GARMIN.mat* file. Matlab file that contains the information retrieved from the *GPX* file. The fields of the table are:

TimeStamp The time stamp of the even provided by GPS device. It is number which represents in Matlab notation the time in UTC. The GPS device gives this information with ms from Jan 1st 1970. No Timezone translation is needed.

Lat latitude provided by GARMIN device, in degrees.

Lon longitude provided by GARMIN device, in degrees.

Elev Height of the position point above ellipsoid WGS 84.

Date The information is the same as TimeStamp, but in a human readable format (day-month-year hour:minute:second).

A.2.2. Collect data from u-BLOX

As two GPS devices were used, and the output format was different, another program was needed to read it.

Program name *UBX_reader*. The program reads all the files with *.UBX* extension in the same folder. This program has to be run after *GPS_reader.m*, because it takes the data saved by *GPS_reader.m* in *GARMIN.mat* file, and appends the new information in the same format. After that the program saves the information again in the same file *GARMIN.mat*.

Input Any file with .UBX extention provided by the u-blox device.

Output Same output as Garmin device, but with an extra column:

Accu The accuracy provided by the u-blox device, in meters. It is not considered in the thesis as GPS devices are always considered "Ground truth".

A.3. Collect Data from Mobile Device

Two kind of (logcat) files were retrieved from both mobile phones. One of them (Logcat radio file) has radio information and the other (Logcat file) is a global log of the phone activity. Besides, the format for the log created when connected to 3G is different from the log when it is connected with 2G. In this thesis we focused in extracting the information contained in the Logcat radio, leaving the Logcat general file as a support file to consult what kind of activity the phone was performing when a point in Google Maps appeared.

A.3.1. Logcat radio from 3G connection

Program name *READ_radio.m*

Input Text files in folder *./RADIO* whose name is finished in *_radio.txt*. As we have the log of 2 phones, the files whose name ends with *b_radio.txt* are treated as logs for the second phone, and the rest are treated as logs of the first phone. An example of the file I used is *nov1a_radio.txt* for a file recorded on November 1st for the phone 1.

Output *CID.mat*, *PSC.mat* and *CID_PSC.mat*. Matlab files described below.

CID.mat It contains the table CID. This table has the columns:

Time Number that represents in Matlab format the date and time of the event. It is converted into UTC. The contents of the logcat file don't have any information about the year it was recorded, so it is taken from the properties of .txt file. So if the file is modified and saved again, this information may be lost. The date and time stored in the file are in local timezone. function *date2UTC()* converts Central European Time (+Daylight saving time) into UTC. If the phone is going to work in other Timezone, this function should be modified.

Phone Stores 'Phone1' or 'Phone2' depending the phone this log belongs to.

mMcc Identification of cell-tower . This identifies the country of the mobile company. For the Netherlands, 204.

mMnc Identification of cell-tower. This identifies the company of the mobile operator. For Vodafone, the network used in this thesis, it is 04.

nLac Identification of cell-tower, stands for Location Area Code. This identifies the local area.

mCid Identification of cell-tower. This identifies the tower Cell Tower ID.

mPsc Identification of cell-tower. It is scramble code. Only in 3G, it changes over time and the number repeat themselves in different areas of the country, so it can't be used for tower identification in this thesis. It could be done doing a fingerprinting of the area though.

ss Signal strength in Arbitrary Strength Unit (ASU).

ber Signal to noise ratio.

dates The information is the same as Time, but in a human readable format (day-month-year hour:minute:second).

PSC.mat It contains the table PSC. This table has the columns:

Time Number that represents in Matlab the date and time of the event. It is converted into UTC. This field is the same that field Time in table CID table.

Phone Stores 'Phone1' or 'Phone2' depending the phone this log belongs to.

PSC_1 to PSC_10 Identification PSC of up to ten cell towers. When connecting to 3G and several towers are in sight, the phone registers the PSC of each tower (neighboring towers), but not the rest of the ID (mMcc, mMnc, mLac, mCid). So it is impossible to identify to exactly which tower it refers to.

SS_1 to SS_10 Signal Strength of up to ten cell towers (towers that correspond to PSC_1 to PSC_-10).

BER_1 to BER_10 Signal to Noise Ratio (SNR) of up to ten cell towers (which correspond to PSC_1 to PSC_10).

mMcc Identification of first cell-tower . This identifies the country of the mobile company. For the Netherlands, 204.

mMnc Identification of first cell-tower. This identifies the company of the mobile company. For Vodafone, the network used in this thesis, it is 04.

nLac Identification of first cell-tower, stands for Location Area Code. This identifies the local area.

mCid Identification of first cell-tower. This identifies the tower Cell Tower ID.

dates The information is the same as Time, but in a human readable format (day-month-year hour:minute:second).

The last five fields are the same as in *CID.mat* table. The values contained in them correspond only to the first connected tower. For the neighboring towers, we only have the PSC information, as indicated above.

CID_PSC.mat It contains table CID_PSC. This table connects the PSC ID with CID. This connection is not in the official database proviced by NFI and it is created based on the log files (so it is not complete and may be variable with time). The relation is not one to one, so, the same PSC can correspond to several CIDs. The table contains the columns: mMcc, mMnc, MLac mCid and mPsc whose description is the same as the two previous Matlab files.

A.3.2. Logcat radio from 2G connection

In 2G, a lot more of information can be extracted from the Logcat. While in 3G only the scramble code PSC could be obtained in the neighboring towers, in 2G the complete ID (MCC, MNC,LAC and ID) is achievable.

Program name *READ_radio_2G.m*

Input Text files in folder "*./RADIO*" whose name is finished in "*_radio.txt*". As we have the log of 2 phones, the files whose name ends with "b_radio.txt" are treated as logs for the second phone, and the rest are treated as logs of the first phone.

CID_2G.mat It contains the table CID_2G. This table has the columns:

Time Number that represents in Matlab the date and time of the event. It is converted into UTC. This field is the same that field Time in table CID table.

nTower Number of neighboring towers.

Phone Stores 'Phone1' or 'Phone2' depending the phone this log belongs to.

mMcc_1 to mMcc_8 Identification of up to eight cell towers. This identifies the country of the mobile company. For Netherlands, it is 204.

mMnc_1 to mMnc_8 Identification of up to eight cell towers. This identifies the company of the mobile phone. For Vodafone it is 04.

mLac_1 to mLac_8 Identification of up to eight cell towers. This identifies the local area.

mCid_1 to mCid_8 Identification of up to eight cell towers. This identifies the tower uniquely by its Cell ID.

ss_1 to ss_8 Signal Strength of up to eight cell towers (the one the device is connected plus the neighboring ones).

ber_1 to ber_8 Signal to Noise Ratio of up to eight cell towers.

A.4. Collect data from Excel Logbook

For every experiment, the conditions of connection, weather, traffic, means of transport, date, starting time, finishing time (both in UTC) and phone used were noted down in a logbook. This logbook was the registered on an Excel file, and then the program experiments.m translates this information from excel to Matlab.

Program name *Import_experiments.m*

Input *./DATA/Experiments.xlsx* (Excel file where all the parameters were written down).

Output *Experiments.mat*, Matlab table which contains the table experiments. This table has the columns:

Tstart Number that represents in Matlab the date and time of experiment start (UTC). It is the number of days since 0 of January of year zero.

Tend Number that represents in Matlab the date and time of experiment end (UTC).

Weather Categorical variable that indicates the weather condition in the experiment (clear, cloudy, rainy).

Environment Variable which indicates what the experiment was carried out in rural or urban area.

Traffic Variable which indicates the traffic conditions (busy, normal, light).

Action Variable which indicates the mean of transport (still, walking, bike, car, tram, train)

G2 Logical variable which indicates the 2G connection was active in the phone.

G3 Logical variable which indicates the 3G connection was active in the phone.

WIFI Logical variable which indicates the WIFI connection was active in the phone.

GPS Logical variable which indicates the GPS connection was active in the phone.

Phone1 Logical variable that indicates that preceding columns refer to phone number 1.

Phone2 Logical variable that indicates that preceding columns refer to phone number 2.

A.5. Putting all together

Once all the data is collected in different files, the program cruzar collects them all and builds table XTABLE. XTABLE is built using *JSON1* and *JSON2* tables as the base data.
The fields taken from JSON files are:

- TimStamp (UTC time)
- Lat (Latitude in degrees)
- Lon (Longitude in degrees)
- Accu (Radius of Google accuracy in meters)
- Phone (phone 1 or 2)

Then, data from GARMIN is appended to the registers where *TimeStamp* from both Garmin and JASON files matches (within a margin of 7 seconds).
The columns appended are:

Glat Garmin point latitude

Glon Garmin point longitude

Err_xy Distance between Google JSON point and position given by Garmin in meters.

Err_x Distance measured in East-West direction between Google and Garmin position in meters.

Err_y Distance measured in North-South direction between Google and Garmin position in meters.

Then, the data from Logcat radio file are appended to the registers where TimeStamp matches (within a margin of 5 seconds). The columns appended are:

P3G Signal Power of the 3 nearest towers in ASU (as taken from logcat) when 3G connection is active.

P2G Signal Power of the 3 nearest towers in ASU (as taken from logcat) when 2G connection is active.

nTower Number of neighboring cell towers

dBm1, dBm2, dBm3 Power of the 3 nearest towers in dBm (decibel milliwatt)

Last data to append is the data taken from the experiment conditions:

G2, G3, WIFI, GPS Possible connections were active at the moment.

Weather, Traffic, Environment, Action Parameters variables characteristic of the experiment.

Now all the data taken from Google, GPS device, mobile devices (phones) and logbook, are in the same table XTABLE. Distances and angles are not added to this table, they will be calculated separately.

B

Matlab interface user guide

This Annex is to describe the collection of programs written in MATLAB®to elaborate this thesis.

B.1. Interface to show 1 day period of experiment data

The purpose of this interface is to check that data introduction from annex A is done correctly. We can also see details of sources of data (Garmin and Google). The name of the program to be run is *datasalection2*.

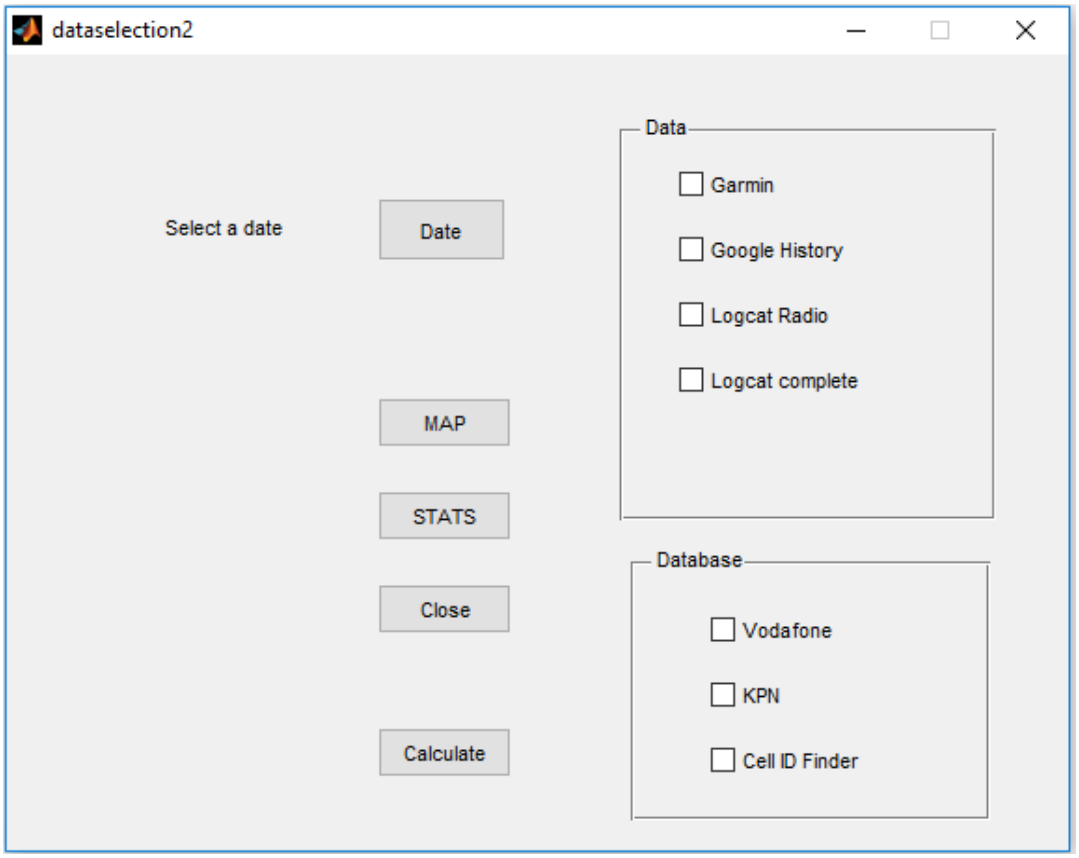


Figure B.1: dataselection2 interface. In the box of data you can select the data source you want to load (Garmin, Google History, Logcat radio and Logcat). In the box of Database you choose the Cell tower database where you want to get the Cell Tower position from (Vodafone, KPN or Cell ID finder). Vodafone and KPN databases belong to NFI, and Cell ID finder gets the tower locations from a public database. With the buttons, you can select a date to study, generate a map and general statistics about the chosen data and calculate a model for the next step.

B.1.1. Instructions

Running the program *dateselection2* the window shown in figure B.1 appears. Click on *Date* button and a calendar appears (see figure B.2). Then select a date which is shown in red. The dates in red are the ones we have GARMIN Data explained in section A.2. Note that although it is called Garmin data, it includes u-blox data if the other GPS device was used that day. Then click the button *OK*.

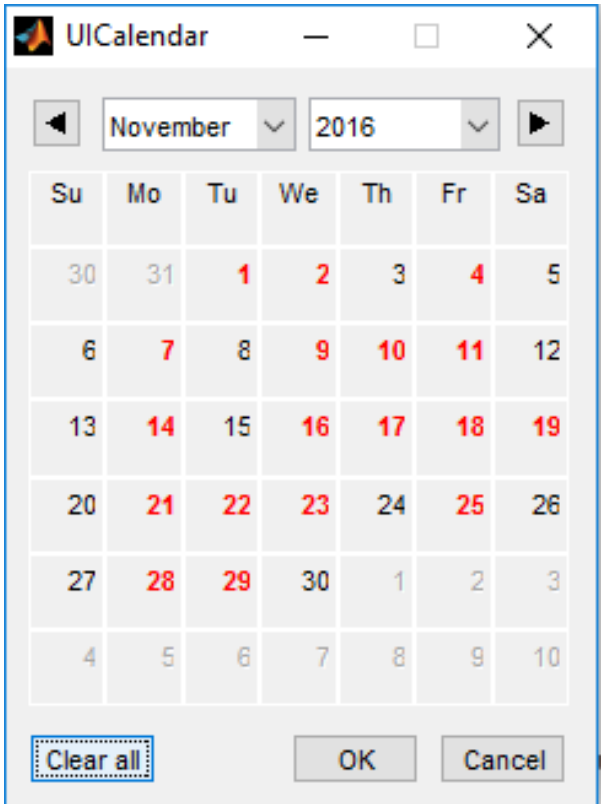


Figure B.2: Matlab calendar. It allows you to choose a date to examine the data collected on that date. Days in red are the ones that actually contain experiment data, days in gray are empty.

After the day is selected, select the check-boxes GARMIN, GOOGLE, Logcat RADIO and Vodafone (if working with the thesis data, KPN network data could also be used and then KPN should be checked instead). The numbers which appear near to the check-boxes are the number or registers of the kind chosen for that day. See figure B.3.

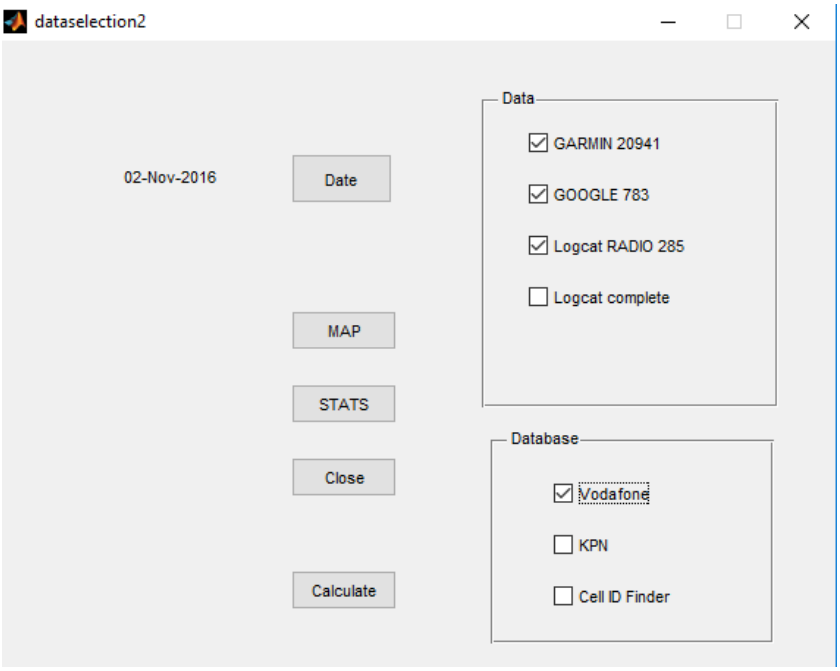


Figure B.3: Same interface as figure B.1 but with the appropriate data selected. We excluded logcat complete, which shows the complete Logcat file gathered in that day (if available) but is difficult to manage and read.

Clicking on button *Map* and the map of figure for that day appears. See figure B.4.

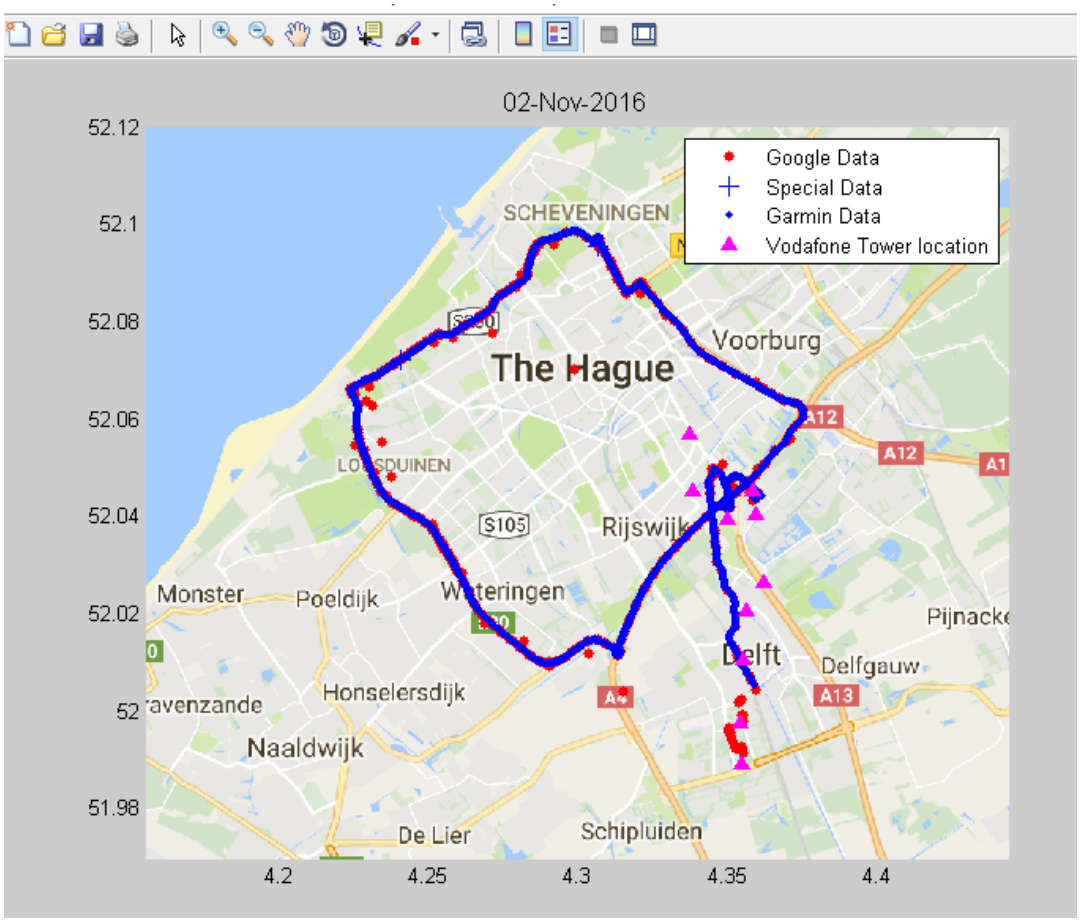


Figure B.4: Map of experiments performed in the 2nd of November 2016. It pops up when pressing the button MAP of figure B.3. In red, the Google point obtained from JSON file. In blue, the GPS points and pink are the Cell Tower location. Special data are the points taking expressly for the chapter results. In pink, the chosen Cell Tower location, in this case Vodafone.

The map can be zoomed in and out, panned clicking on the buttons shown in figure B.5.

Figure B.5: Cursor zoom in/out. With the hand, the image can be moved.

Clicking on the cursor button and selecting a point in the map, information will appear.

- If it is a GPS point (the blue ones), the time registered for that point will appear. See figure B.6.

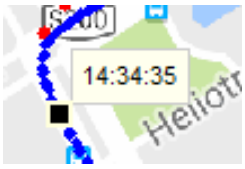


Figure B.6: Pressing on the Garmin point, the device time registered shows up.

- If it is a GOOGLE TimeLine point, the time (or times) that Google says the device during experiment was there will appear. A circle with accuracy provided by Google will be shown, and the corresponding GPS point (or points) will become green color.

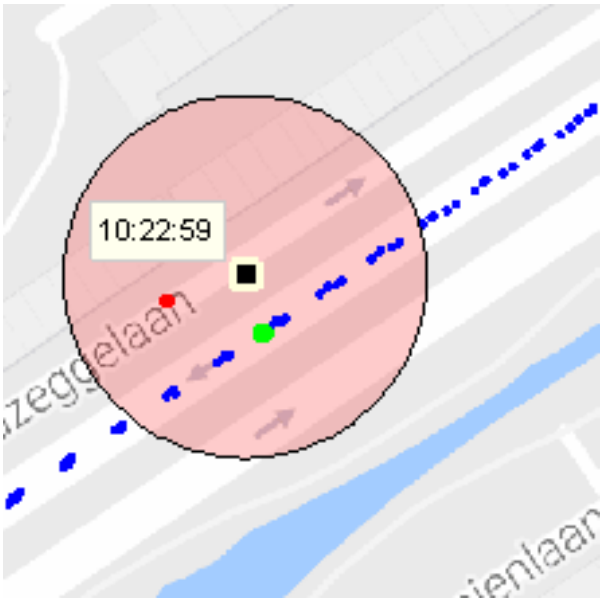


Figure B.7: Google point shown in interface. It shows the time in UTC. Red circle is drawn with the accuracy radius in meters given by JSON file. The device could be anywhere inside that circle at that time. The blue points are all the GPS points but the one turned green is exactly device's true position at the given time.

- If Logcat radio was selected and the data is available, then the towers the phone was connected at that time will change from pink to green too.

- If you click on a tower Vodafone, its info will appear, and the corresponding Cell Finder location will appear as a six-pointed star. With this option we discovered that Cell finder location (public database) is not a good database. Not all connected towers where found, many were missing in the database and the locations do not coincide with the official database. Many of them are really far away. If you click on a Cell Finder tower, the corresponding Vodafone tower will appear as a star.

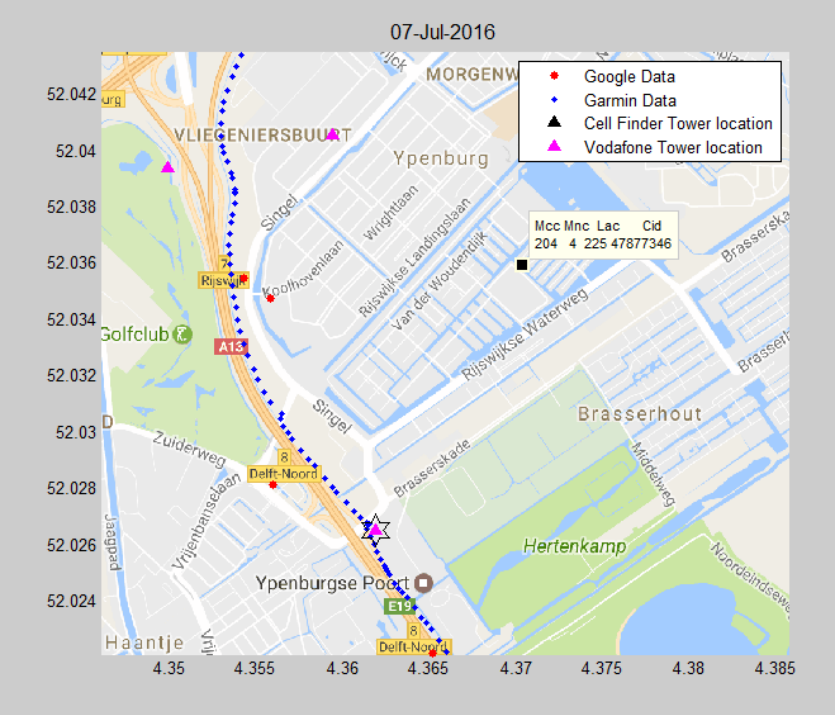


Figure B.8: Cell finder selected. When two or more Cell Tower databases are selected inside the *Database* box (see figure B.1) we can compare the public database obtained from the Internet to the official database provided by NFI. In the figure, we pressed both Vodafone and Cell ID finder and then pressed a Vodafone tower, the tower indicated by a six point star is the one that in Cell ID finder has the same Cell ID in the public database

B.2. Model generation

To select data and generate linear model the interface *Graphics* was programmed.

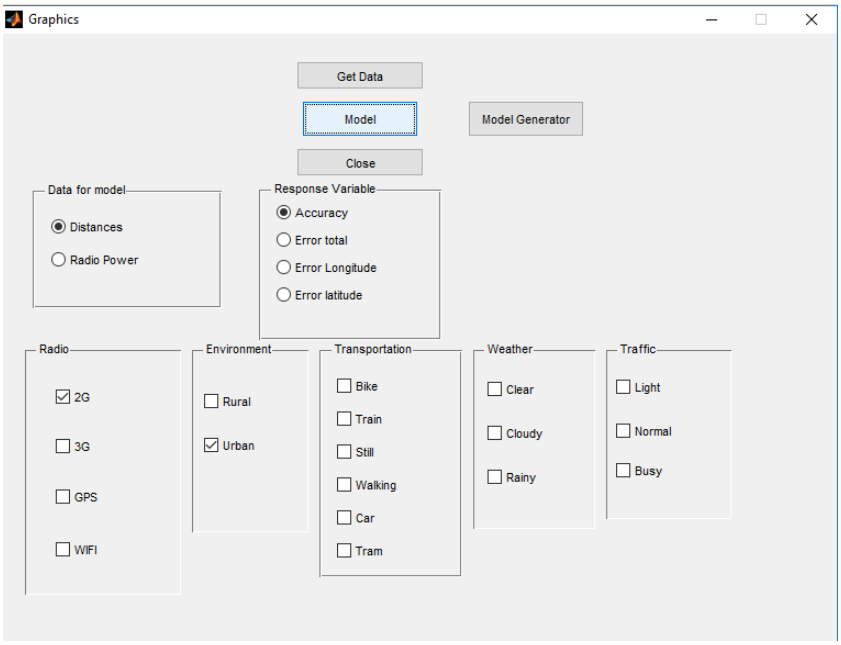


Figure B.9: *Graphics.m* interface. This interface allows to select the data. In it, you have the box data for model (were you can choose to take as matrix A the distances to towers or Signal Power). Then the response variable box, with accuracy, Error, error in x and error in y. The rest of the boxes you mark which regressors you want to have into account to perform the model).

This interface (see figure B.9) includes:

Data for model With this option we can choose which regressors will be used to generate the model. We have 2 options:

Distances Includes the distances and angles to the 3 nearest towers connected at that given time.

Radio Power The Signal Strength registered on the phone of the 3 nearest towers. Information acquired from Logcat radio.

Response Variable With this option we select which will be the variable to model: Accuracy provided by Google, or error in Google measurements (distance in meters from Google location point compared to Garmin location point at the same time).

Filters:Radio, Environment, Transportation, Weather, Traffic With this options we can select a filter to generate the model. If one or more options are selected in a group, the corresponding registers in XTABLE are selected to generate the model. Besides a categorical variable will be used in the model so that coefficients for the rest of regressors (distances, angles, power strength) will be different depending of the values of this variable. For example, if in the group **Weather**, *Clear* and *Cloudy* options are selected, experiments with *Rainy* weather won't be taken into account, and the coefficient for *distance1 clear*, is a different coefficient for *distance1 cloudy*. In the other hand, if no option is selected in a group, no filters are applied regarding that variable, and no distinctions are done with respect the values it can have. For example, if no option is selected in group **Weather**, all the experiments are part of the model generation, but the value of Weather variable is not taken into account, the coefficients have not *weather* distinction.

Once the *Data*, *Response Variable* and *Filters* are selected, the button "*Get Data*" has to be pressed. At this moment, the filters are applied, and distances and angles are calculated (they are not stored in the database). The function which calculates these distances and angles is *dist_regressor(T)*. It also calculates the Google hits and misses, and generates some statistical graphs with selected data. No model is yet generated, just data analysis. See figures B.10 and B.11.

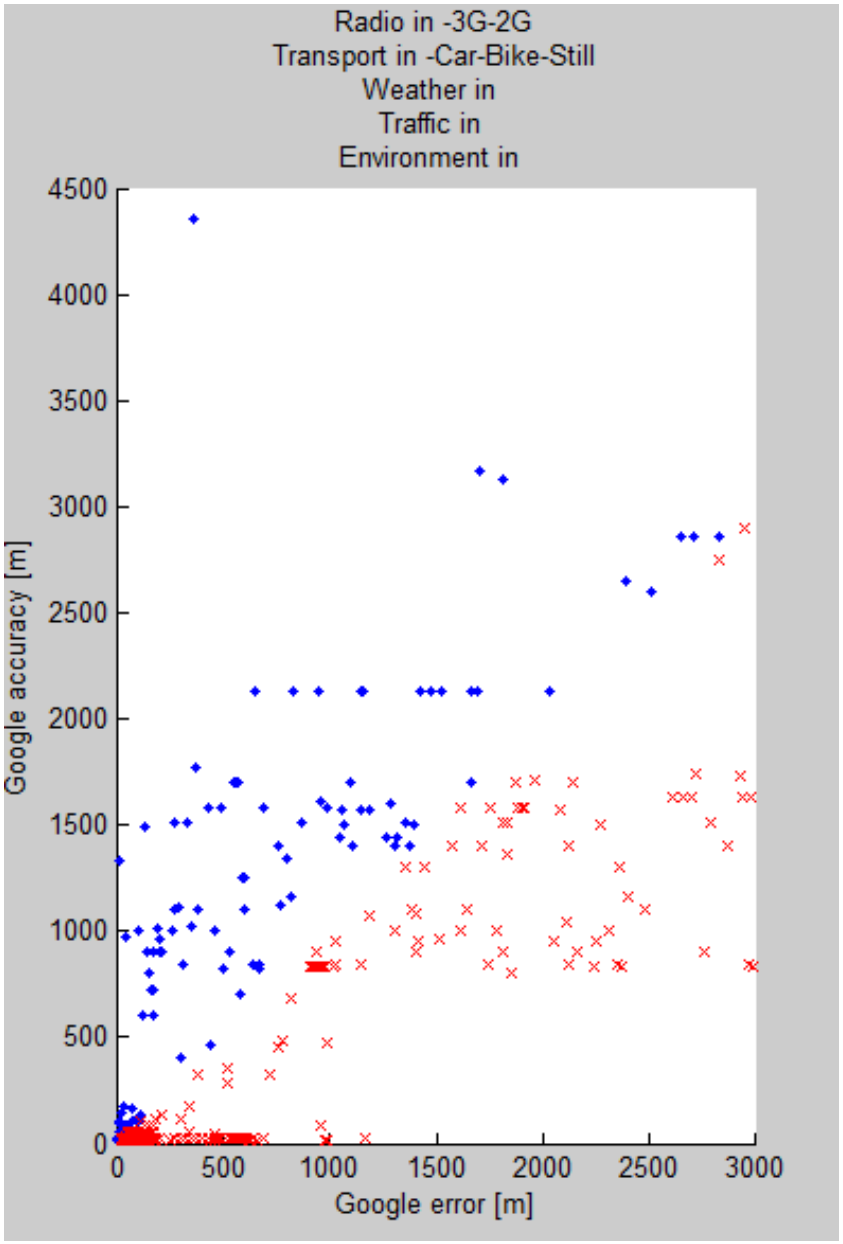


Figure B.10: Google hits (blue) and misses (red). Vertical axis represents Google accuracy in meters (radius that Google provides around the JSON point). In horizontal axis the Google error is represented, the distance between the Google pint and GPS point in meters. So if error is smaller than accuracy, Google did a good prediction because the point is inside the accuracy circle, and we call it a hit. Contrary case if it is a miss.

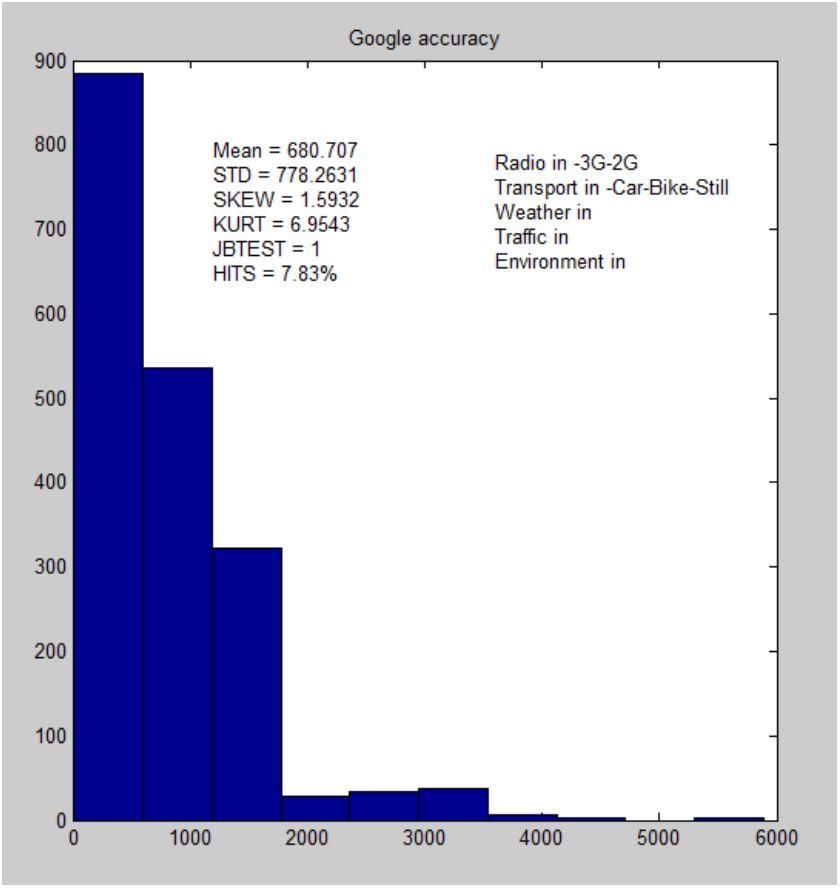


Figure B.11: Accuracy statistics, window that appears after pressing "Get Data" Button. After selection the regressors, data box and response variable. in this case, the parameters were 2G and 3G, transport Still-Car-Bike and No weather, traffic or environment classification taken. The information we can see about the data is the mean, standard deviation, skewness, kurtosis, Jaque-Bera test, and percentage of hits. It is an histogram representing the Google accuracy in meters.

Once the data is selected buttons "*Model*" and "*Model generator*" become active. If we click on "*Model*" 2 models are generated. One of them is the normal linear model with all the regressors selected and other is a model generated with stepwise. *Stepwise* is an intelligent technique that generates a linear model adding and removing regressors and interactions (products between regressors) depending of their significance to the model. ANOVA (ANalysis Of VAriance) tables are shown, and some graphics or the two models are generated. See figures B.12 and B.13.

	SumSq	DF	MeanSq	F	pValue
d2	2.19188e+06	1	2.19188e+06	4.4607	0.0348
d3	1.5311e+06	1	1.5311e+06	3.1160	0.0777
a1	7.8464e+06	1	7.8464e+06	15.9689	6.6932e-05
a2	1.1142e+06	1	1.1142e+06	2.2676	0.1323
a3	3.6325e+07	1	3.6325e+07	73.9280	1.7036e-17
SOURCE	6.4223e+03	1	6.4223e+03	0.0131	0.9090
d1:SOURCE	5.1494e+05	1	5.1494e+05	1.0480	0.3061
d2:SOURCE	3.7463e+06	1	3.7463e+06	7.6243	0.0058
d3:SOURCE	3.2874e+06	1	3.2874e+06	6.6906	0.0098
a1:SOURCE	4.0925e+06	1	4.0925e+06	8.3289	0.0039
a2:SOURCE	2.2512e+07	1	2.2512e+07	45.8157	1.7413e-11
a3:SOURCE	6.6832e+06	1	6.6832e+06	13.6015	2.3257e-04
Error	9.0360e+08	1839	4.9136e+05	1	0.5000

	SumSq	DF	MeanSq	F	pValue
d2	2.4857e+07	1	2.4857e+07	76.1423	5.8263e-18
a1	6.7191e+06	1	6.7191e+06	20.5824	6.0832e-06
a2	8.8996e+07	1	8.8996e+07	272.6194	3.4155e-57
a3	1.1713e+08	1	1.1713e+08	358.7880	3.3506e-73
SOURCE	1.2563e+07	1	1.2563e+07	38.4839	6.8113e-10
Action	1.4952e+07	2	7.4759e+06	22.9006	1.5030e-10
d2:a2	9.2220e+06	1	9.2220e+06	28.2495	1.1971e-07
d2:a3	3.6319e+06	1	3.6319e+06	11.1256	8.6848e-04
d2:SOURCE	7.2486e+06	1	7.2486e+06	22.2044	2.6362e-06
d2:Action	2.2277e+07	2	1.1139e+07	34.1209	2.8275e-15
a1:a2	7.1312e+07	1	7.1312e+07	218.4474	8.8403e-47
a1:a3	1.6907e+08	1	1.6907e+08	517.9122	3.7966e-101
a1:SOURCE	1.9161e+07	1	1.9161e+07	58.6954	2.9658e-14
a2:a3	1.8530e+07	1	1.8530e+07	56.7624	7.6889e-14
a2:SOURCE	2.3758e+06	1	2.3758e+06	7.2779	0.0070
a2:Action	5.7994e+06	2	2.8997e+06	8.8826	1.4486e-04
a3:SOURCE	4.6693e+06	1	4.6693e+06	14.3034	1.6058e-04
a3:Action	1.0489e+07	2	5.2444e+06	16.0651	1.2123e-07
SOURCE:Action	4.9232e+06	2	2.4616e+06	7.5405	5.4781e-04
Error	5.9675e+08	1828	3.2645e+05	1	0.5000

Figure B.12: ANOVA (ANalysis Of VAriance) table for the same case picked up before. This is to choose the variables for the Linear model regression, using stepwise method. They are basically chosen by their lowest p-value.

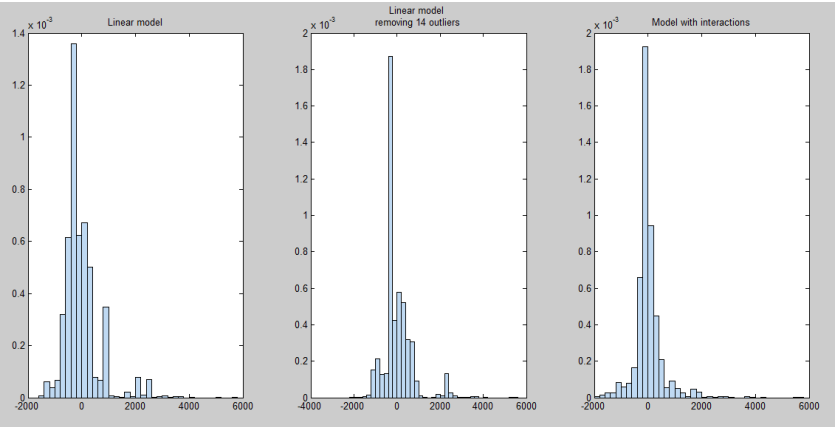


Figure B.13: Residuals from the linear model. The program applies the linear model to the data chosen and refines it. In this image it can be seen how outliers are removed (applying Cook's distance) and thus the Residuals improved.

Once the model has been generated, there is an option to save it. See figure B.14. If user answers affirmative, the file is saved in folder ./OUTPUT.

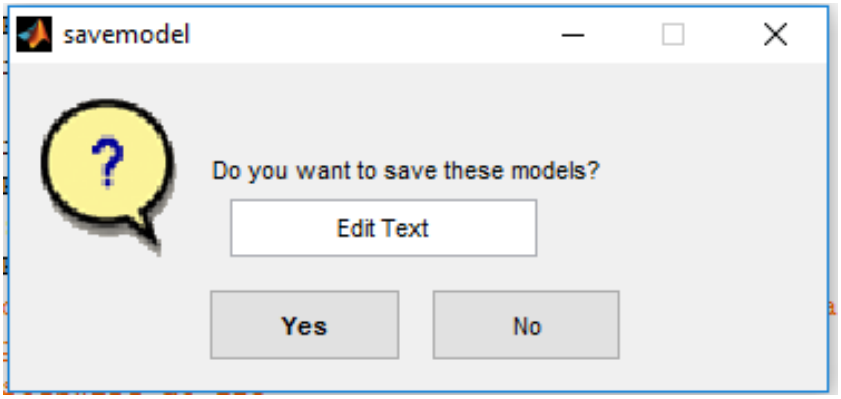
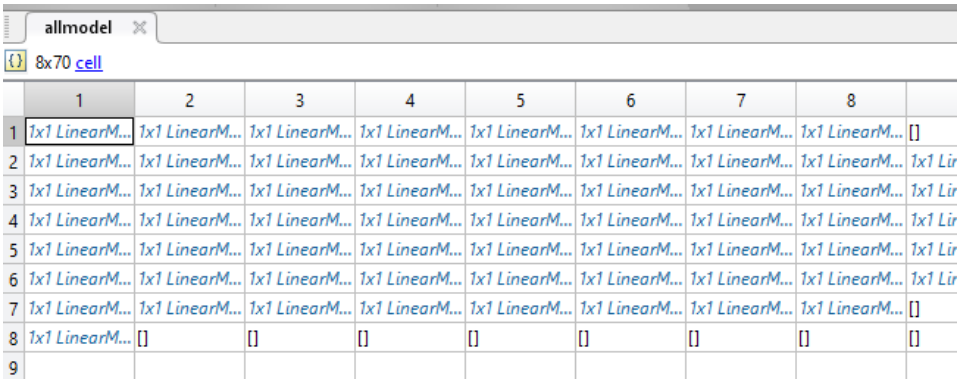


Figure B.14: Save the model interface. Automatically pops up after the figures and results of the model generation, giving the option of saving it.

Clicking on "*Model Generator*" the program creates all possible linear models without interactions. Some models have 1 regressor, others have 2 regressors,... and the last one has all the regressors. All these models are stored in a cell array *allmodel*, like the one seen in figure B.15.



	1	2	3	4	5	6	7	8	
1	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	[]
2	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 Lir
3	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 Lir
4	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 Lir
5	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 Lir
6	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 Lir
7	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	1x1 LinearM...	[]
8	1x1 LinearM...	[]	[]	[]	[]	[]	[]	[]	[]
9									

Figure B.15: *allmodel* cell array. Matlab cell array that contains all the possible generated models saved.

In this cell array, the first row has all the linear models with 1 regressor ($\binom{8}{1} = 8$ models), the second line

has the models with 2 regressors ($\binom{8}{2} = 28$ models) and the last line has the model with 8 regressors ($\binom{8}{8} = 1$ model). The maximum number of regressors depends on the filters selected in the Graphics interface.
At the same time other arrays store *R2*, *R2_adjusted*, *Cp* and *RMSE* values for all these models.
We also have the option to save the data after this program generates the models. If the option is accepted, the data is saved in folder ./LINVARSEL. With this program we are able to generate and store all possible linear models in .mat files in the same folder. Each set of models is in a different file in LINVARSEL folder.

B.2.1. Model completion

Once we have calculated all possible (interesting) linear models with or without categorical regressors, it is time to improve the model adding interactions. An interaction is a new regressor as a result of the product of two simple regressors. To perform this task two programs have been written: "*best_model.m*" and "*improve_model.m*".

Best_model This program asks for a set of linear models stored in LINVARSEL (see figure B.16) folder and generates two tables:

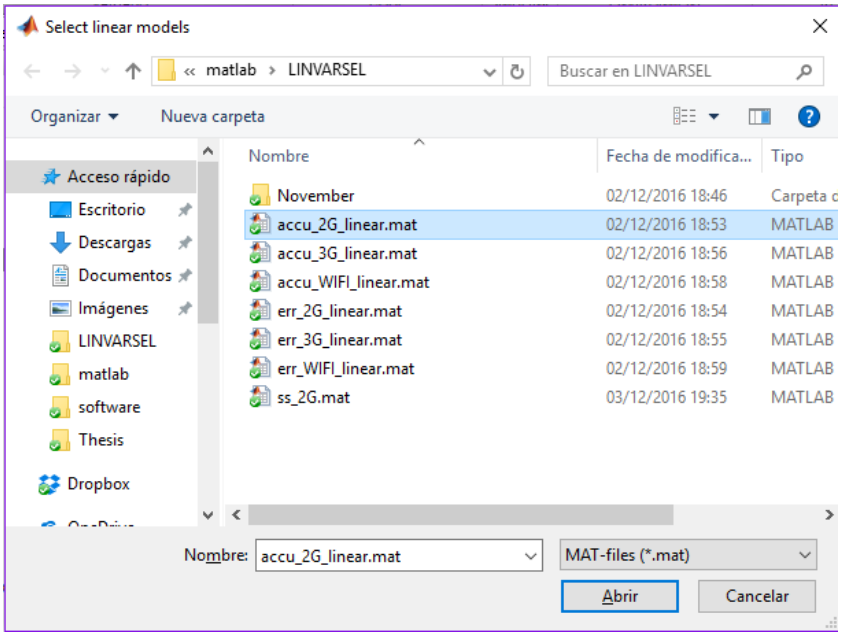


Figure B.16: Saved linear models. They are inside the LINVARSEL folder

Varselection In this table the best 2 models of each group in *allmodel* table are shown. It decides the best models on the lowest *R adjusted* and the lowest *RMSE*. It helps to decide which are the best regressors. See figure B.17

varselection | bestmodel

15x4 table

	1 R_2	2 R_2adj	3 C_p	4 S_
1 1 + a3	0.1073	0.1068	98.6627	735.5144
2 1 + d2	0.0390	0.0385	247.7331	763.1429
3 1 + a3 + Action	0.1243	0.1229	71.5546	728.8679
4 1 + d2 + a3	0.1221	0.1212	68.3886	729.5921
5 1 + d1 + a1 + a3	0.1405	0.1391	30.3329	722.1182
6 1 + d2 + a3 + Action	0.1381	0.1363	43.4426	723.2969
7 1 + d1 + a1 + a3 + Action	0.1532	0.1509	12.6115	717.1490
8 1 + d2 + a1 + a3 + Action	0.1510	0.1487	17.3465	718.0674
9 1 + d1 + d2 + a1 + a3 + Action	0.1547	0.1520	11.2712	716.6945
10 1 + d1 + d3 + a1 + a3 + Action	0.1540	0.1513	12.7194	716.9758
11 1 + d1 + d2 + a1 + a2 + a3 + Action	0.1552	0.1520	12.2452	716.6893
12 1 + d1 + d2 + d3 + a1 + a3 + Action	0.1548	0.1516	13.0203	716.8399
13 1 + d1 + d2 + d3 + a1 + a2 + a3 + Action	0.1553	0.1516	14.0743	716.8503
14 1 + d1 + d2 + a1 + a2 + a3 + SOURCE + Action	0.1552	0.1515	18.1900	716.8728
15 1 + d1 + d2 + d3 + a1 + a2 + a3 + SOURCE + Action	0.1553	0.1512	20	717.0303

Figure B.17: *Varselection* table that shows the best two models of each group. It shows the regressors which were employed to generate the model and the *R, adjusted R, Cp* coefficient and *RMSE* for that model.

Best_model Shows R2adj (R adjusted) for the same models in *varselection* and the pointers to these best models. See figure B.18

varselection | bestmodel | interaction

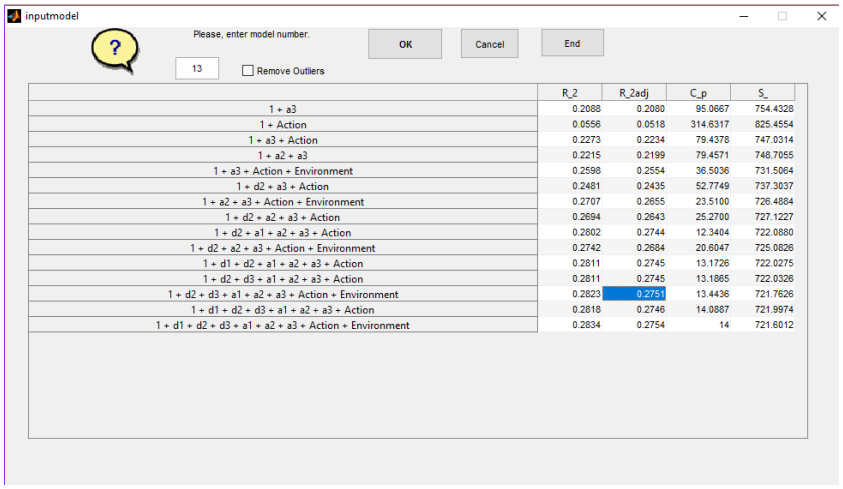
15x3 table

	1 R_2	2 R_2adj	3 S	4
1 + d1:d2	0.1856	0.1821	703.8576	
2 + d1:a1	0.1629	0.1592	713.6202	
3 + d1:a2	0.1659	0.1622	712.3413	
4 + d1:a3	0.1626	0.1590	713.7238	
5 + d1:Action	0.1791	0.1751	706.8498	
6 + d2:a1	0.1656	0.1619	712.4682	
7 + d2:a2	0.1556	0.1519	716.7132	
8 + d2:a3	0.1555	0.1519	716.7406	
9 + d2:Action	0.1562	0.1520	716.6602	
10 + a1:a2	0.2115	0.2081	692.5796	
11 + a1:a3	0.3280	0.3250	639.3890	
12 + a1:Action	0.1622	0.1581	714.0846	
13 + a2:a3	0.1562	0.1526	716.4400	
14 + a2:Action	0.1571	0.1530	716.2676	
15 + a3:Action	0.1618	0.1577	714.2698	

Figure B.18: best_model table. It shows the R adjusted of the models in varselection and their index in the *allmodel* table. For example, if we wanted to choose the seventh 1st column like in the figure, to save it or apply changes to it, we would have to take it from the variable *allmodel.m* like this:allmodel7,2 because it is the seventh model and the O2 (index two) it is 2

The program shows *varselection* table and lets the user select one of the models to start the process of adding interactions. Just click on the table and the number of the model is chosen. See figure B.19.

Then, when clicking "*OK*" next program is executed.



	R_2	R_2adj	C_p	S_
1 + a3	0.2088	0.2080	95.0667	754.4328
1 + Action	0.0556	0.0518	314.6317	825.4554
1 + a3 + Action	0.2273	0.2234	79.4378	747.0314
1 + a2 + a3	0.2215	0.2199	79.4571	748.7055
1 + a3 + Action + Environment	0.2598	0.2554	36.5036	731.5064
1 + d2 + a3 + Action	0.2481	0.2435	52.7749	737.3037
1 + a2 + a3 + Action + Environment	0.2707	0.2655	23.5100	726.4884
1 + d2 + a2 + a3 + Action	0.2694	0.2643	25.2700	727.1227
1 + d2 + a1 + a2 + a3 + Action	0.2802	0.2744	12.3404	722.0880
1 + d2 + a2 + a3 + Action + Environment	0.2742	0.2684	20.6047	725.0826
1 + d1 + d2 + a1 + a2 + a3 + Action	0.2811	0.2745	13.1726	722.0275
1 + d2 + d3 + a1 + a2 + a3 + Action	0.2811	0.2745	13.1865	722.0326
1 + d2 + d3 + a1 + a2 + a3 + Action + Environment	0.2823	0.2751	13.4436	721.7626
1 + d1 + d2 + d3 + a1 + a2 + a3 + Action	0.2818	0.2746	14.0887	721.9974
1 + d1 + d2 + d3 + a1 + a2 + a3 + Action + Environment	0.2834	0.2754	14	721.6012

Figure B.19: Choosing model selection with *best_model*. With this interface, you don't need to actually go to allmodel.m variable, you can directly click and choose the model from here attending the coefficients displayed. Clicking on *OK*, the process continues.

Improve_model This program takes as input the model selected and then it adds all possible interactions, one by one, to the selected model, and generates a cell array with all these possibilities (*intermodel*) and a summary table (*interaction*). See figure B.20.

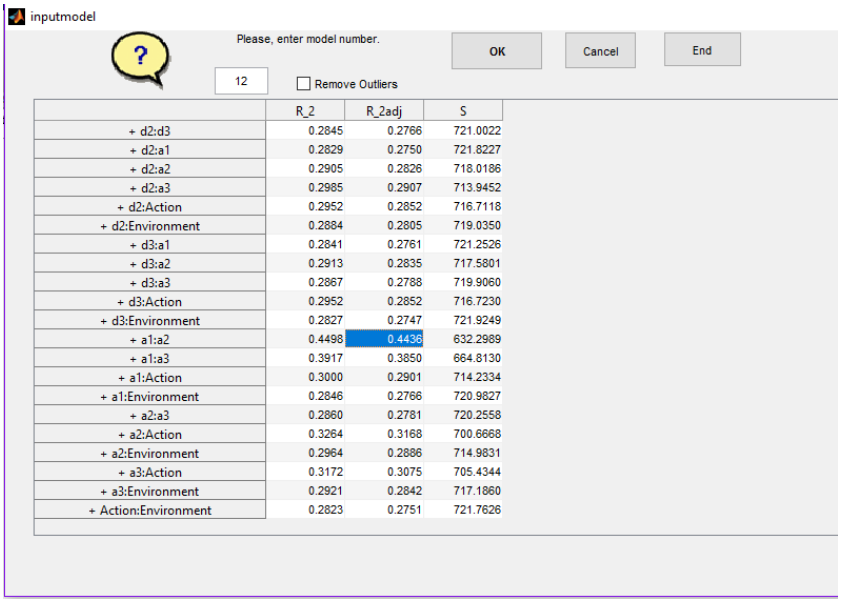
varselection | bestmodel | interaction

15x3 table

	1 R_2	2 R_2adj	3 S	4
1 + d1:d2	0.1856	0.1821	703.8576	
2 + d1:a1	0.1629	0.1592	713.6202	
3 + d1:a2	0.1659	0.1622	712.3413	
4 + d1:a3	0.1626	0.1590	713.7238	
5 + d1:Action	0.1791	0.1751	706.8498	
6 + d2:a1	0.1656	0.1619	712.4682	
7 + d2:a2	0.1556	0.1519	716.7132	
8 + d2:a3	0.1555	0.1519	716.7406	
9 + d2:Action	0.1562	0.1520	716.6602	
10 + a1:a2	0.2115	0.2081	692.5796	
11 + a1:a3	0.3280	0.3250	639.3890	
12 + a1:Action	0.1622	0.1581	714.0846	
13 + a2:a3	0.1562	0.1526	716.4400	
14 + a2:Action	0.1571	0.1530	716.2676	
15 + a3:Action	0.1618	0.1577	714.2698	

Figure B.20: Interaction table. It shows the improved (or worsened) R and R adjusted after making products with the regressors.

The program *Improve_model* shows the results of adding all possible interactions and lets choose the model. If "*OK*" button is pressed, the process is repeated, so many interactions can be added. Each time the program is run, it gives the option to save the models and *intermodel* and *interaction* variables in a file. The folder of these data is ./IMPROVE. See figure B.21.



	R_2	R_2adj	S
+ d2:d3	0.2845	0.2766	721.0022
+ d2:a1	0.2829	0.2750	721.8227
+ d2:a2	0.2905	0.2826	718.0186
+ d2:a3	0.2985	0.2907	713.9452
+ d2:Action	0.2952	0.2852	716.7118
+ d2:Environment	0.2884	0.2805	719.0350
+ d3:a1	0.2841	0.2761	721.2526
+ d3:a2	0.2913	0.2835	717.5801
+ d3:a3	0.2867	0.2788	719.9060
+ d3:Action	0.2952	0.2852	716.7230
+ d3:Environment	0.2827	0.2747	721.9249
+ a1:a2	0.4498	0.4436	632.2989
+ a1:a3	0.3917	0.3850	664.8130
+ a1:Action	0.3000	0.2901	714.2334
+ a1:Environment	0.2846	0.2766	720.9827
+ a2:a3	0.2860	0.2781	720.2558
+ a2:Action	0.3264	0.3168	700.6668
+ a2:Environment	0.2964	0.2886	714.9831
+ a3:Action	0.3172	0.3075	705.4344
+ a3:Environment	0.2921	0.2842	717.1860
+ Action:Environment	0.2823	0.2751	721.7626

Figure B.21: Improve_model table. In this example the interaction number 12 is the one which increases *R2adj* until 0.4436. It gives the option to remove outliers (by Cook's distance) and save it pressing *OK*.

Each time a collection of models are calculated, a figure with factorized cook's distance is shown. It helps to decide if we want to remove outliers for next calculation. At the end of the process, we click "*End*" button and the selected model is stored in folder ./FINAL. See figure B.22

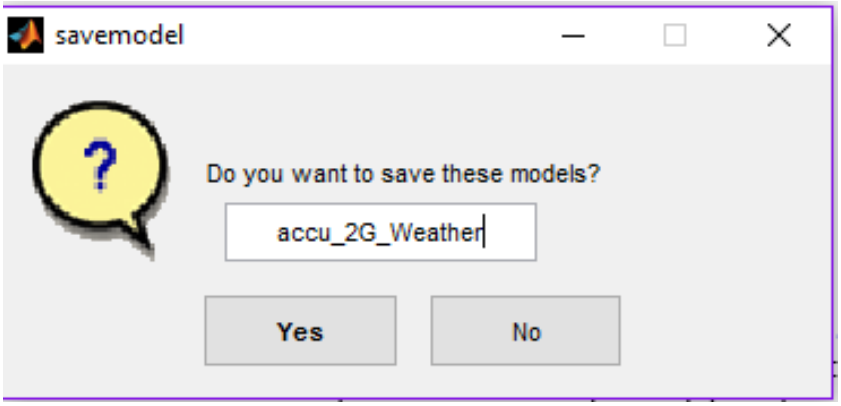


Figure B.22: Saving final model. When we are satisfied with the model we choose, pressing *End* makes this window pop-up and save the result in folder ./FINAL as a Matlab file.

B.3. k-fold validation

Once we have built the models we have them in FINAL folder. A program to test the model has been written for that purpose.

Program name crvalidate_model. This program imports a model, then takes all the observations which were used to generate it and classifies them into 12 bins in a random way. Each bin has the same (or very similar) number of observations. Then each bin is taken as test data, and the rest (the

other 11 bins) are used to generate a new model with the same criteria as the original one. Then the test data is entered as input data to the new model, and then we obtain new predicted values. These predicted values are compared to the real data from the test. As this process is repeated as many times as number of bins, at the end we have a table with the results. The meaning of the values of this table is explained in section 5.7.1

Input A model to evaluate, saved from the program *Improve_model*. The name of the model we want to evaluate has to be written in the program.

Output KFOLD table in matlab format. See figure B.23

meanytest	meanerr	stderr	RMSE
360.85	2.8009	511.33	283.64
410.07	100.89	632.15	328.07
369.72	32.232	480.22	258.51
302.34	-36.87	366.15	192.32
337.4	-7.421	469	236.33
296.22	-92.041	468.74	250.17
417.76	51.006	563.64	291.57
297.19	-63.595	356.36	208.63
367.29	23.899	568.56	300.04
356.45	-16.648	484.07	275.31
396.17	33.399	587.1	332.9
357.42	-1.3558	470.69	264.34

Figure B.23: K-Fold table results. It divides the total number of observations into twelve bins and tests each one of the combinations. It shows the mean test, mean error, standard deviation of the guess and RMSE of the predictions.

Index

List of Figures

List of Tables

Bibliography

[1] Android debug bridge. Web page. URL https://developer.android.com/studio/command-line/adb.html.

[2] About nfi. URL https://www.forensicinstitute.nl/.

[3] What is the rss (received signal strength)? URL https://www.accuware.com/support/knowledge-base/what-is-the-signal-strength-rss/.

[4] Technology dead reckoning (dr). URL http://www.furuno.com/en/gnss/technical/tec_dead.

[5] Gnss frequently asked questions - gps. URL https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/faq/gps/.

[6] Iso 5725-1:1994(en): Accuracy (trueness and precision) of measurement methods and results — part 1: General principles and definitions.

[7] Rijksdriehoeksstelsel. URL https://www.kadaster.nl/rijksdriehoeksstelsel.

[8] Cell phone tracking. URL http://www.largevents.eu/wp/wp-content/uploads/2012/10/Cell_Phone_Tracking.pdf.

[9] Dead reckoning (dr) – ds & ym, 2012. URL https://pzsc.org.uk/shorebased/deadreckoning/.

[10] My current location, 2015. URL https://mycurrentlocation.net/.

[11] Map types, 2016. URL https://developers.google.com/maps/documentation/javascript/maptypes.

[12] Gps receivers use trilateration, 2017. URL http://gisgeography.com/trilateration-triangulation-gps/.

[13] J. S. Abel and J. W. Chaffee. Existence and uniqueness of gps solutions. *IEEE Transactions on Aerospace and Electronic Systems*, 27(6):952–956, Nov 1991. ISSN 0018-9251. doi: 10.1109/7.104271.

[14] David A. Belsley. [collinearity and least squares regression]: Comment: Well-conditioned collinearity indices. *Statistical Science*, 2(1):86–91, feb 1987. doi: 10.1214/ss/1177013441.

[15] David A. Belsley. A guide to using the collinearity diagnostics. *Computer Science in Economics and Management*, 4(1):33–50, 1991. ISSN 1572-9974. doi: 10.1007/BF00426854. URL http://dx.doi.org/10.1007/BF00426854.

[16] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? â€“ arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014. doi: 10.5194/gmd-7-1247-2014. URL http://www.geosci-model-dev.net/7/1247/2014/.

[17] Lina Chen, Binghao Li, 3 Kai Zhao, 3 Chris Rizos, and Zhengqi Zheng1. An improved algorithm to generate a wi-fi fingerprint database for indoor positioning. *Sensors (Basel)*, 2013.

[18] CIE4522. 1.4 – gps signals. In *GPS FOR CIVIL ENGINEERING AND GEOSCIENCES*, Q4 2015-2016.

[19] R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977. ISSN 00401706. URL http://www.jstor.org/stable/1268249.

[20] R. Dennis Cook. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365):169–174, 1979. doi: 10.1080/01621459.1979.10481634. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481634.

[21] R. Dennis Cook and Sanford Weisberg. *Residuals and Influence in Regression*. New York: Chapman and Hall, 1982.

[22] Josh Corbat. What is scattering? - definition & examples. URL http://study.com/academy/lesson/what-is-scattering-definition-examples.html.

[23] Murphy Curtiss. *Believable Dead Reckoning for Networked Games*. 2011.

[24] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaa, and L. E. Meester. *A Modern Introduction to Probability and Statistics*. Springer London Ltd, 2005. ISBN 1852338962. URL http://www.ebook.de/de/product/3054516/f_m_dekking_c_kraaikamp_h_p_lopuhaa_l_e_meester_a_modern_introduction_to_probability_and_statistics.html.

[25] A. Edens. *Cell Phone Investigations: Search Warrants, Cell Sites and Evidence Recovery*. Cell phone investigations series. Police Publishing, 2014. ISBN 9781631800061. URL https://books.google.nl/books?id=vTDSrQEACAAJ.

[26] David L. Fried. Differential angle of arrival: Theory, evaluation, and measurement feasibility. *Radio Science*, 1975.

[27] Susan Garavaglia and Asha Sharma. A smart guide to dummy variables: Four applications and a macro. *Dun & Bradstreet Murray Hill, New Jersey 07974*. URL http://www.ats.ucla.edu/stat/sas/library/nesug98/p046.pdf.

[28] Camillo Gentile, Nayef Alsindi, Ronald Raulefs, and Carole Teolis. *Geolocation Techniques: Principles and Applications*. Springer, 2013.

[29] Onur C. Hamsici and Aleix M. Martinez. Spherical-homoscedastic distributions: The equivalency of spherical and normal distributions in classification. *Machine Learning Research*, 2007.

[30] John W. Harbaugh. *A Computer Method for Four-Variable Trend Analysis Illustrated by a Study of Oil-Gravity : Variations in Southeastern Kansas*. Kansas Geological Survey, 1964.

[31] S. Hartzell, L. Burchett, R. Martin, C. Taylor, and A. Terzuoli. Geolocation of fast-moving objects from satellite-based angle-of-arrival measurements. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(7):3396–3403, July 2015. ISSN 1939-1404. doi: 10.1109/JSTARS.2015.2438865.

[32] Alex Heath. Google can show you everywhere you've been on a map that's surprisingly detailed. *Tech Insider*, 2016.

[33] Google Maps Help. View or edit your timeline, 2017. URL https://support.google.com/maps/answer/6258979?co=GENIE.Platform%3DAndroid&hl=en.

[34] Willy Hereman and William S. Murphy Jr. Determination of a position in three dimensions using trilateration and approximate distances. *Decision Sciences*, 1995.

[35] J. Hoy. *Forensic Radio Survey Techniques for Cell Site Analysis*. Wiley, 2014. ISBN 9781118925744. URL https://books.google.nl/books?id=ZJLVBQAAQBAJ.

[36] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 45–57, New York, NY, USA, 2004. ACM. ISBN 1-58113-868-7. doi: 10.1145/1023720.1023726. URL http://doi.acm.org/10.1145/1023720.1023726.

[37] J.G. Translating wgs84 coordinates, 2004. URL http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=10915.

[38] John. Find x location using 3 known (x,y) location using trilateration, 2014. URL http://math.stackexchange.com/questions/884807/find-x-location-using-3-known-x-y-location-using-trilateration.

[39] M. Khalaf-Allah. Time of arrival (toa)-based direct location method. In *2015 16th International Radar Symposium (IRS)*, pages 812–815, June 2015. doi: 10.1109/IRS.2015.7226229.

[40] Jemima Kiss. Google admits collecting wi-fi data through street view cars, 2010. URL https://www.theguardian.com/technology/2010/may/15/google-admits-storing-private-data.

[41] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. 2015.

[42] John Lekner. *Theory of Reflection, of Electromagnetic and Particle Waves*. Springer, 1987.

[43] Richard G. Lomax and Debbie L. Hahs-Vaughn. *Statistical Concepts: A Second Course, Third Edition*. Routledge Academic, 2007. ISBN 9780805858501. URL https://www.amazon.com/Statistical-Concepts-Second-Course-Third/dp/0805858504?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0805858504.

[44] LTE-Anbieter.info. Asu wert - signalstärke messen und interpretieren. URL http://www.lte-anbieter.info/technik/asu.php.

[45] Albert Madansky. *Testing for Independence of Observations*, chapter 3, pages 92–119. Springer New York, New York, NY, 1988. ISBN 978-1-4612-3794-5. doi: 10.1007/978-1-4612-3794-5_4. URL http://dx.doi.org/10.1007/978-1-4612-3794-5_4.

[46] Wolfram MathWorld. Statistical median, 1999-2017. URL http://mathworld.wolfram.com/StatisticalMedian.html.

[47] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Ganga-Jamuna Press, 2010.

[48] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., 2014.

[49] William L. Mularie. Department of defense world geodetic system 1984: Its definition and relationships with local geodetic systems. Technical report, National Imagery and Mapping Agency (NIMA), 2000.

[50] David Munoz, Frantz Bouchereau Lara, Cesar Vargas, and Rogerio Enriquez-Caldera. *Position Location Techniques and Applications*. Academic Press, 2009.

[51] NIST/SEMATECH. e-handbook of statistical methods, 10 2013. URL http://www.itl.nist.gov/div898/handbook/.

[52] Richard R. Picard and R. Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984. doi: 10.1080/01621459.1984.10478083. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1984.10478083.

[53] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. 2002.

[54] David W. Sabo. The f-distribution, 2003. URL http://commons.bcit.ca/math/faculty/david_sabo/apples/math2441/section9/comp2popvarht/fdist/fpfdist.htm.

[55] Martin Sauter. *From GSM to LTE: An Introduction to Mobile Networks and Mobile Broadband*. John Wiley & Sons, 2010.

[56] John S.Seybold. *Introduction to RF propagation*. Wiley, 2005.

[57] GPS World staff. Innovation: Assisted gps: A low-infrastructure approach, 2002. URL http://gpsworld.com/innovation-assisted-gps-a-low-infrastructure-approach/.

[58] P.J.G. Teunissen, D.G. Simons, and C.C.J.M. Tiberius. *Probability and Observation Theory:an Introduction*. TU Delft, 2005.

[59] Inc. The MathWorks ©. Wilkinson notation - matlab & simulink - mathworks, 1994-2017. URL https://es.mathworks.com/help/stats/wilkinson-notation.html.

[60] Christian Tiberius. *Primer on Mathematical Geodesy*. Faculty of Civil Engineering and Geosciences Delft University of Technology, 2014. CTB3310 / CTB3425.

[61] Hans van der Marel. *Reference Systems for Surveying and Mapping*, chapter 10 Dutch national reference systems, pages 55–64.

[62] Peter Vermeulen. Rd2wgs, 2010-2017. URL https://nl.mathworks.com/matlabcentral/answers/39847-rd-coordinates-to-wgs84.

[63] Stephan von Watzdorf and Florian Michahelles. Accuracy of positioning data on smartphones. *Proceeding LocWeb '10 Proceedings of the 3rd International Workshop on Location and the Web Article No. 2*, 2010.

[64] Shu Wang, Jungwon Min, and Byung K. Yi. *Location Based Services for Mobiles: Technologies and Standards*. LE Electronics Mobile Research, 2008.

[65] Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912934.

[66] Wikipedia. Standar deviation diagram. URL https://en.wikipedia.org/wiki/File:Standard_deviation_diagram.svg.

[67] G. N. Wilkinson and C. E. Rogers. Symbolic description of factorial models for analysis of variance. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 22(3):392–399, 1973. doi: 10.2307/2346786. URL http://www.jstor.org/stable/2346786.

[68] Moustafa Youssef and Ashok Agrawala. The horus location determination system. In *Wireless Networks*, page 357–374. Springer, 1995-2017.

[69] Fred Zahradnik. Assisted gps, a-gps, agps, 2016. URL https://www.lifewire.com/assisted-gps-1683306.

[70] Paul A. Zandbergen. Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 2009.