Pulak ROY

URBAN MOBILITY PREDICTION USING PICTURE METADATA

Final Master's Project

Directed by Dr. Jordi DUCH

Department of Computer Engineering and Mathematics

UNIVERSITAT ROVIRA I VIRGILI

2017

Abstract

Quantitative research about mobility on tourism has traditionally relied on surveys and economic datasets, generally composed of small samples with a low spatio-temporal resolution. The exponential increase of the number of technological devices such as mobile phones, connected cameras, wearables that we carry and use every day has resulted in a large availability of large databases with detailed information, which contain from geolocated information to communication messages. This data availability has led to a new area of research, where researchers can use this information to study human mobility patterns at an unprecedented scale.

In this thesis, we have used the information contained in one of the largest databases available online, the Flickr Dataset which contains more than 100 million pictures, to study mobility patterns of visitors inside of a city. This type of analysis has already been applied on the same dataset to study mobility patterns between cities of the United Kingdom, or using information contained in non-tourism oriented social network (twitter) to analyze attractiveness of some of the most popular touristic or to study the perceived value of tourism destinations by the visitors. We have concentrated on the mobility of tourists inside of a city, to try to identify typical mobility patterns, and try to estimate the likelihood of visiting one spot according to the previous visited locations in the city.

We have clustered the data using DBSCAN algorithm to find out most popular places in Barcelona. We have decided to continue our study by selecting number of clusters. Later, we have applied machine learning system to try to predict the mobility of users from our clustered data. Applying all these methods, we are able to predict with 65% of probability, what is the next movement of the user comparing to what will be 5% of the random.

Acknowledgements

First of all, I would like to thank my advisor Dr. Jordi Duch for his guidance, support and endless patience over the large course of this Master thesis.

All of my Master Professors over the entire course, for their kindness, guidance and motivation.

I would like to thank Julian Vicens for his suggestions in some technical part of my thesis.

I would like to thank my friends Monica, Edgar, Chinh, Alex for their support in entire master course.

Finally, I want to express my gratitude to my Guru Shrii Shrii Anandamurtiji for his endless blessings, guidance in all aspects of my life. I am grateful to my family and friends for all of their unconditional love and affection.

Contents

Abstract **iii**

Acknowledgements **v**

1 Introduction **1**
1.1 Research 2
1.2 Aim 3
1.3 Target Group 4
1.4 Thesis Structure 4

2 Background **7**
2.1 Clustering 7
2.2 Basics 7
2.3 What is Density Based Clustering? 9
2.4 Algorithm of DBSCAN 9
2.5 Random Forest 10
2.6 What is a Decision Tree? 11
2.7 Bootstrap Method 11
2.8 Bootstrap Aggregation (Bagging) 12
2.9 Random Forest Algorithm 13

3 State of the Art **15**
3.1 YFCC100M: The New Data in Multimedia Research 15
3.2 Touristic site attractiveness seen through Twitter 17
3.3 Modeling Human Mobility Patterns Using Photographic Data Shared Online 20

4 Data Characterization **25**
4.1 Data Acquisition and Processing 25
4.1.1 Data Acquisition 25
4.1.2 Data Pre-Processing 27
4.2 Data Characterization 29

5 Clustering Data **39**
5.1 Applying DBSCAN to Filtered Dataset 39
5.1.1 Experiment Result 40
5.2 Applying DBSCAN to Non-filtered Dataset 43
5.3 Discussion 45

6 Prediction of Mobility **49**
6.1 Prediction with Non-filtered Data 49
6.2 Prediction with Filtered Data 54
6.3 Discussion of the Results 56

7 Conclusion and Future Work **57**

A Experiment of DBSCAN Clustering **59**

Bibliography **63**

List of Figures

2.1 Example Real Life Data 8
2.2 Result k-means clustering 8
2.3 Sample Dense Cluster 9
2.4 DBSCAN Algorithm Component 10
2.5 Decision Tree 11
2.6 Bagging technique 12
2.7 Random Forest Technique 13

3.1 Captured and uploaded media objects YFCC100M 16
3.2 Global Coverage YFCC100M Data 16

4.1 Collection of media object inside Barcelona 27
4.2 Histogram Staying Duration 31
4.3 Histogram Captured Pictures 33
4.4 Heatmap staying duration 34

5.1 Filtered Dataset of Barcelona 40
5.2 Result DBSCAN Filter Data 42
5.3 Result DBSCAN Non-Filter Data 44
5.4 Hot Spots Filter Data 45
5.5 Hot Spots Non-Filter Data 46

6.1 Cluster Sequence Distribution Non-filter Data 52
6.2 Cluster Sequence Distribution Filter Data 55

List of Tables

4.1 Attributes in YFCC100M data. 25
4.2 Sample data:users staying duration in Barcelona. 30
4.3 Example data of each user and his total captured photos. . . 32
4.4 Dataset: exact amount users in terms of their photo taken and staying duration. 34
4.5 Dataset shows exact amount users in terms of their photo taken and staying duration. 35
4.6 Dataset reveals exact amount users in terms of their photo taken and staying duration. We considered more number of intervals of photo taken and staying period. 37

5.1 Notably, for each value of eps and minpoints we get legitimate clusters; But eps = 0.05 and minpoints = 70 seems to be promising parameter, because they generate 42 clusters and covers above half of the total users. 41
5.2 Here, for each value of eps and minpoints we get small amount of clusters; But eps = 0.03 and minpoints = 1000 seems to be promising parameter, because they generate 21 clusters and covers about 72002 from the total users. 43
5.3 Clusters centroid founded by applying DBSCAN algorithm on non-filtered data; Parameter values are eps = 0.03 and minpoints =1000. 47

6.1 A sample of dataset tracked by trajectory. 50
6.2 A sample of clustered dataset of Barcelona. We can observe, this user has traveled six places in the city. We can identify his movement through Cluster_id attribute. He visited cluster 2, 5, 6,7,18 and again 6. 50
6.3 A data sample of cluster list for each trajectory of each user. 51
6.4 Prediction of movement using random forest classification algorithm. There were 200 training data and 30 test data. . . 53
6.5 Prediction estimates of sorted and ordered clustered sequences for non-filtered dataset. The length of the cluster sequence is 5 and 6 respectively. 54
6.6 Prediction estimates of sorted and natural ordered clustered sequences for filtered dataset. The length of the cluster sequence is 5 and 6 respectively. 55

A.1 Test Case 1 59
A.2 Test Case 2 59
A.3 Test Case 3 60
A.4 Test Case 4 60
A.5 Test Case 5 60
A.6 Test Case 6 61

A.7 Test Case 7 61
A.8 Test Case 1 61
A.9 Test Case 2 62

Chapter 1

Introduction

One of the most significant current discussions in human mobility leads to analyze human decision making criteria, habits and interest towards popular places. In the last few years, there has been a growing interest in the study of predicting human mobility through different sources of data. The study of human mobility in the context such as anthropology [1], exploration of urban mobility and transport [2], criminology [3], spreading of viruses [4, 5, 6]has shown a remarkable statistical consistencies.

During these days, fast expansion of new communication medium such as mobile phones and strong networked social media has widen up the door to build enhanced models of human mobility. Significant research about mobility on tourism was traditionally confined on surveys and monetary datasets, usually comprised of small samples with a flat spatio-temporal resolution. Due to the dramatic boost of the amount of technological devices such as smart phones, online linked cameras, wearable that we carry and use every day has resulted in an immense availability of large databases with comprehensive information. These databases contain from geo located information to communication messages. Availability of such data has led to a new area of research, where researchers can use this information to study human mobility patterns at an unprecedented scale.

In this thesis, we have used the information contained in one of the largest databases available online, the Flickr Dataset [7, 8] which contains more than 100 million pictures, to study mobility patterns of visitors inside of a city. This type of analysis has already been applied on the same dataset to study mobility patterns between cities of the UK [9], or using information contained in non-tourism oriented social network (twitter) to analyze attractiveness of some of the most popular touristic [10] or to study the perceived value of tourism destinations by the visitors [11].

In this work, we have studied basic characteristics of the Yahoo100M data. It has enabled us to find out detailed parameters that adequately express the characteristics and nature of the dataset. Then we apply clustering technique to identify dense places in Barcelona where people take a lot of pictures. Clustering groups of points that happened in the same places,we put identification on those points. As a result, it automatically detects this space of points where there are ample number of photos taken together. To achieve this goal, we have used some sort of algorithm which processes about three hundred thousands of points, and generates important groups there.

Finally, we have tested a mobility prediction method using data analytics and machine learning. We have experimented with a system which uses the minimum amount of information and capable of determining the most probable next location that an individual would visit based on his previous locations and the information of the mobility of other people. Additionally, we have tested that this predicting approach can be applied to observe a single persons as well as overall tourist's aggregated movement.

1.1 Research

Perceiving human mobility pattern is not a new concept. For several years great effort has been devoted to the study of predicting human mobility through different sources of data. Common significant researches have been carried out by manipulating surveys and economic datasets, which are composed of small samples with a low spatio-temporal resolution. Recently, due to the availability of huge number of shared media objects obtained through services like Flickr and Instagram, people are being able to obtain datasets having diverse attributes comprising geo-tags, timestamps etc. Therefore, research groups continue examining ways to advancing this problem.

Several publications have appeared in recent years documenting diverse approach to address this problem. Since last few years, several authors have conducted a remarkable research endeavor to understand human mobility patterns, where authors take into account individual human route [12, 13] and agreement movements [14, 15, 16].

Study of human mobility patterns at distinct spatial extent and collective levels from an personal to community displacements is an significant research topic for their broad applicability. It includes covering from urban and transportation planning [17, 18] and resource sharing [19, 20] to the forecasting of migration flows [19, 20] and spreading of epidemic at limited, provincial, or global territory [21, 22, 23]. According to the literature on [24], recently the gravity model [25, 26] and radiation model [14] are the most largely popular models. Assumptions in gravity model says that the number of people travelling between two locations has strong correlation with some power of their population size, and degenerates as some power of the distance between them. On the other hand, the radiation model assumes human movements are sort of scattered processes usually relies on the distribution of population over the space.

Success has been found on area like highway flows [25], air-travel [26, 27], commuting [28], and mobile phone calls between cities [29] by using gravity model. Nonetheless, gravity model have some drawback such as the unavailability of adjustable data and the insufficiency of a first principle derivation [14, 30] are major drawback for this model. On the contrary, very good prediction has been found by applying radiation model to infer commuting patterns between US counties by using only population data.

However, people are skeptical about its application at different territorial extents because it is not successful in capturing commuting inside urban or metropolitan areas [30, 31, 32]. As a result and it has never been applied to model high range travel patterns. After analyzing drawbacks of these models, it has been recommended that it is possible to highly improve the quality of their results if additional data is provided [32, 33]. Indeed, certain number of works have investigated records from mobile phone companies to study individual [12] and aggregated mobility [34, 35, 36], depicting that analyzing human activity can help to deduce these flows.

Lenormand et al. [18] studied highway and roadway transportation networks in Europe through using footprint from Twitter, whereas to infer the relationship between user action and place transitions Noulas et al. [15] used Foursquare dataset. International travel pattern has been modeled by Hawelka et al.[16] by analyzing data of twitter users taking into account of their country of residence.

Lenormand et al. have also used Twitter footprint to design commuting from home to work [37]; Using social networks data Grabowicz et al. tried to establish the link between human mobility and interactions using footprint of the users [38];Llorente et al. [39] studied the mobility patterns in Spain using Twitter footprint.

In the very recent years, Barchiesi et al. [9] extracted 16,000 individuals information in UK from Flickr data to model the flows between its 20 largest cities, and assessed their finding with official data. They applied clustering and machine learning approach to infer mobility patterns. We have explained this paper in detail at Section 3. Mariano et al. [24] tried to build a model where they combine classical gravity model with machine learning approach such training real dataset with cross-validation technique.

Summing up the literatures, it can be noticed that researches in this field is moving towards with geo-spatial data sources to predict human mobility. In these works, they are using common technique such as building time stamped trajectories along with geo-location attributes of the dataset to trace the real footprint of the users. In addition, along with statistical inference, clustering and machine learning approaches are becoming common tools to build reliable model for the prediction of human mobility.

1.2 Aim

The aim of this master thesis consists of five parts. First, we want to obtain the flickr dataset, which contains 100 million pictures, filter the pictures according to location criteria, and obtain their metadata. We will do preprocessing and characterization of data to get descriptive parameters that effectively describes the characteristics and behavior of a particular data item. Decision can be made through characterization which will be useful to find patterns, clusters and trends. In addition, we want to have a general overview of the data we collected; we want to present some statistics and charts with all of our experiment conducted.

Second, we want to give an overview of three papers that cover specific aspects. There, we want to explain about general dataset we are going to use which is the flicker one which explains where is the origin of the data. We want to study previous approach of working with tourist based data analysis using data from online data source. Then, we want to review similar work like we are doing that used the same data Yahoo100M to study mobility.

Third, we want to select and apply a clustering mechanism to establish hotspots from the picture geo-location attribute. Through clustering, we want to find out most popular places in Barcelona where tourists take an ample amount of pictures. We want to discover groups of points located in the same place and then we put mark on those points. We want to identify automatically more dense areas according to the user's choice where a lot of photos have been taken together.

Fourth, we want to obtain footprints of all users of Barcelona through accumulating their pictures in time stamped trajectories. Time stamped trajectories will take us to the next step of analysis, in which we will obtain data that will help us to track people's psychology of movement from one place to another.

Finally, we will test a mobility prediction method using data analytics and machine learning. We want to investigate the efficiency of machine learning algorithm to predict the next movement of individual as well as aggregated population.

1.3 Target Group

The targeted group of audience of this thesis can be diverse, specially people involving:

- Study of Data Science.
- Study the Machine learning.
- Study the Human Mobility and Prediction

It is not necessary to have background in clustering and machine learning to understand this thesis. Section 2 will cover the most important areas related to Clustering and Machine Learning techniques.

1.4 Thesis Structure

In Section 2, we introduce the elementary concepts to understand the technical part of our thesis. We explain basic clustering concept and different types of clustering techniques that are used in data analysis. We show the application scenarios on which different kind of clusterings will be more effective. Moreover, we notice the drawback of some clustering technique regarding to some special type of data. We focus on the study of density based clustering which has been used in our work detailing the components and parameters. Then we go forward to understand DBSCAN algorithm. Finally, we enter into detailing of core components and parameters of machine learning based classification mechanism which will be used to predict human mobility.

In Section 3, we review the state of the art in human mobility prediction using social network media datasets. We have presented three important papers in this subject which has high influence on this thesis with some concept.

Section 4 is devoted to describe the data acquisition, pre-processing, and characterization. We illustrate how we have obtained the YahooFlickr dataset. Then we investigate some inconsistencies regarding to the preparation of data for future analysis. We conduct detail characterization on the pre-processed dataset to extract key attributes and statistics from dataset.

Section 5 will outline the clustering mechanism applied on the geo-located data. We detail here how have applied the DBSCAN algorithm by experimenting with large number of different parameters value. At the end of this section we will discuss about the obtained result and do verification with the official data.

In Section 6, we will show how we have used the result of clustering to prepare dataset for predicting user movements. Then we will show how we have applied machine learning algorithm to obtain prediction of next movement of user through analyzing their previous visits. Finally, we will end this section by discussing the result obtained and the performance of prediction comparing with typical random technique

Section 7 draws conclusions about what we obtained from this thesis, the constraints, and different possible ideas for the future work in this context.

Chapter 2

Background

This chapter is devoted in explaining three main methods that have been used in our thesis. First, we have used clustering to find groups of points that were close one to each other. In our case we have used DBSCAN algorithm, and on proceeding towards we will find the impact of applying this method. Second we have used machine learning algorithm. In this case we have used random forest to try to predict movement of the users, and later we will verify why we have used this method.

2.1 Clustering

A clustering method is a technique of grouping a number of vectors or points according to a specific criterion. The possible criteria are generally distance or similarity. Nearness is defined in terms of a particular function of distance, such as Euclidean or geodetic distance.

2.2 Basics

In general, clustering methods such as K-means and hierarchical clustering are suitable for finding spherical-shaped clusters or convex clusters [40]. In other words, they work efficiently when data points tightly close and well separated in nature. In addition, they are also severely affected by the presence of noise and outliers in the data. In addition, there could be many outliers and noise in the experimental data. However, in the real life scenario data can contain clusters of arbitrary shape such as oval, linear and S shape clusters (Figure 2.1). According to the example taken from [40], we can observe some inefficiency of traditional clustering methods. The Figure 2.1 represents a dataset containing non-convex clusters and outliers/noises. We can observe, it contains 5 clusters and outliers, including 2 ovals clusters, 2 linear clusters and 1 compact cluster.

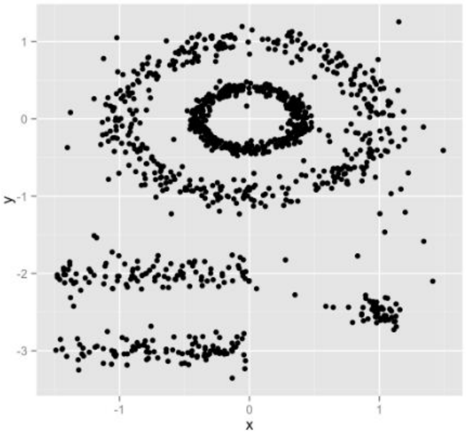


FIGURE 2.1: Sample data for clustering(Source:[40]).

Regarding such data, k-means algorithm has difficulties for identifying theses clusters with arbitrary shape. Figure 2.2 depicts the result of executing k-means algorithm on the above datasets. It clearly indicates the inefficiency of k-means algorithm for this kind of experimental datasets.

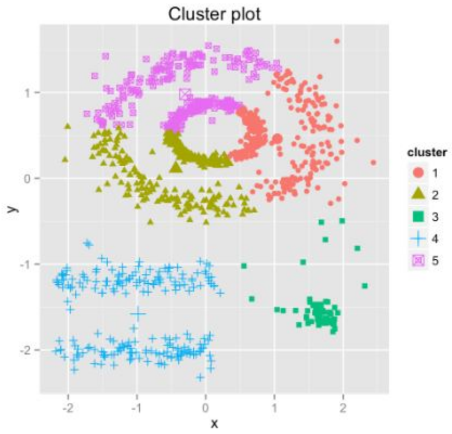


FIGURE 2.2: We know there are 5 five clusters in the data, but it can be seen that k-means method inaccurately identifies the 5 clusters.(Source:[40]).

Usually, k-mean algorithm tries to generate cluster based on calculating variance among data points, where it uses euclidean distance calculation method.In our case, our data points are latitude and longitude, therefore calculating euclidean distance will not work efficiently.For instance, if we consider two geo-local points p((2.1788,51.234) and q(2.0088,51.004), then k-means algorithm probably consider these point in a same cluster. But the actual distance between these points could be 5 kilometers in terms of geodetic

2.3 What is Density Based Clustering?

According to the example presented on [40], the basic idea behind density-based clustering approach is derived from a human intuitive clustering method . In Figure 2.3, one can easily identify four clusters along with several points of noise, because of the differences in the density of points.

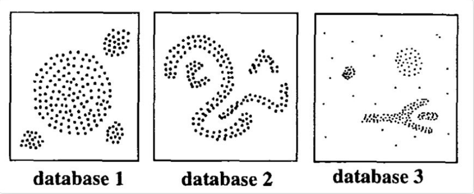


FIGURE 2.3: Clusters are dense regions in the data space, separated by regions of lower density of points.(Source:Ester at al.[41]).

As illustrated in the Figure 2.3, clusters are dense regions in the data space. They are separated by regions of lower density of points. In other words, the density of points in a cluster is considerably higher than the density of points outside the cluster (areas of noise). DBSCAN(Density-Based Spatial Clustering of Applications with Noise), takes into account these kind of situation of clusters and noise. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points. DBSCAN, take into account that clusters are dense groups of points. The idea is that if a particular point belongs to a cluster, it should be near to lots of other points in that cluster[42].

2.4 Algorithm of DBSCAN

The main purpose of DBSCAN algorithm identify dense regions, which can be measured by the number of objects close to a given point. Two important parameters are required for DBSCAN: epsilon eps and minimum points MinPts. The parameter eps defines the radius of neighborhood around a point x. It is called the e-neighborhood of x. The parameter MinPts is the minimum number of neighbors within eps radius.

Any point x in the dataset, with a neighbor count greater than or equal to MinPts, is marked as a core point. We say that x is border point, if the number of its neighbors is less than MinPts, but it belongs to the e-neighborhood of some core point z. Finally, if a point is neither a core nor a border point, then it is called a noise point or an outlier.
The Figure 2.4 shows the different types of points (core, border and outlier points) using MinPts = 6.

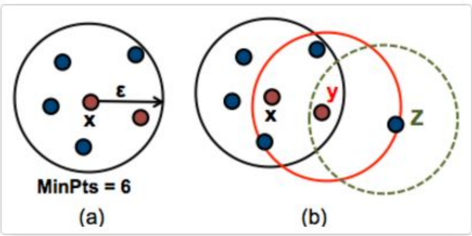


FIGURE 2.4: Here x is a core point because neighbors of x within radius e is 6, y is a border point because neighbors of y is less than MinPts, but it belongs to the e-neighborhood of the core point x. Finally, z is a noise point(Source:[37]).

There are three main component to understand DBSCAN algorithm:

Direct density reachable: *A point A is directly density reachable from another point B if A is in the e-neighborhood of B and B is a core point.*

Density reachable: *A point A is density reachable from B if there are a set of core points leading from B to A.*

Density connected: *Two points A and B are density connected if there are a core point C, such that both A and B are density reachable from C.*

Pseudo code of DBSCAN algorithm taken from [37] works as follow:

1. For each point x_i, compute the distance between x_i and the other points. Find all neighbor points within distance eps of the starting point x_i. Each point, with a neighbor count greater than or equal to MinPts, is marked as core point or visited.

2. For each core point, if it is not already assigned to a cluster, create a new cluster. Find recursively all its density connected points and assign them to the same cluster as the core point.

3. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are treated as outliers or noise.

2.5 Random Forest

In the context of regression and classification tasks, random Forest is an intuitive and effective machine learning method. Although, it can handle dimensional reduction methods, missing values and outlier rather it can perform quite well. It is a sort of ensemble learning method, where a group of weak models combine to form a powerful model. In the following subsections, we will take the idea of basic component of random forest algorithm.

2.6 What is a Decision Tree?

To explain decision tree, we have taken an example from [43]. Decision tree is a type of supervised learning algorithm that is mostly used in classification problems. It deals with both categorical and continuous input and output variables. This approach split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter or differentiator in input variables.

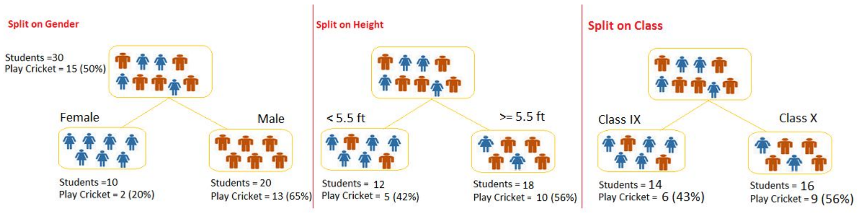


FIGURE 2.5: Decision Tree(Source:[43]).

We can observe from Figure 2.5, a sample of 30 students with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft). Among them 15 out of these 30 play cricket in leisure time. Now, we want to create a model to predict who will play cricket during leisure period? In this case, we need to classify students who play cricket in their leisure time based on highly significant input variable among all three.

In such situation decision tree works good because it will classify the students based on all values of three variables. It will trace the variable, which creates the best alike sets of students. We can observe that, in this case Gender variable is able to identify best uniform sets in comparison to the other two variables.

2.7 Bootstrap Method

To estimate a quantity from a data sample, bootstrap is a effective statistical approach . It is simple to understand if the quantity is a descriptive statistic such as a mean or a standard deviation. According to the example[44], let us assume we have a sample of 100 values of x. Now, we would like to get an estimate of the mean of the sample. We can calculate the mean by mean(x) = sum(x)/100.

In such cases, if we know we have small sample size and our mean has error in it. Improvement of the estimate of our mean using the bootstrap procedure:

1. Create many random sub-samples of our dataset with replacement. To put it differently, we can choose the same value several times).

2.Calculate the mean of each sub-sample.

3.Calculate the average of all of our collected means and use that as our estimated mean for the data.

For example, suppose we have re-sampled our data three times and got the mean values 2.3, 4.5 and 3.3. Taking the average of these we could take the estimated mean of the data to be 3.367.

2.8 Bootstrap Aggregation (Bagging)

According to the example [43], bagging is a method used to reduce the difference of our predictions by combining the result of multiple classifiers modeled on different sub-samples of the same data set. From Figure 2.6 we can have a clearer concept about this.

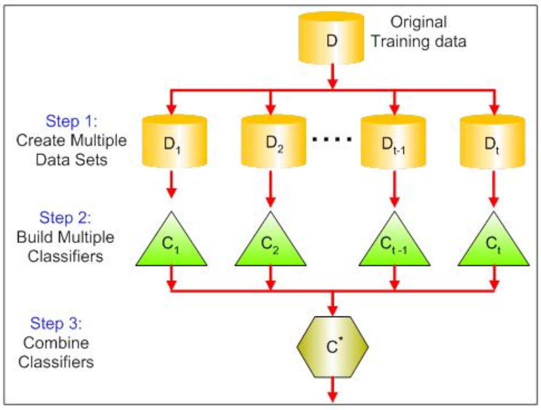

D
Original Training data
Step 1: Create Multiple Data Sets
D_1
D_2
D_{t-1}
D_t
Step 2: Build Multiple Classifiers
C_1
C_2
C_{t-1}
C_t
Step 3: Combine Classifiers
C^*

FIGURE 2.6: Bagging technique; Higher number of models are always better or may give similar performance than lower numbers(Source:[43]).

There are three steps followed in bagging are:

1. **Create Multiple DataSets:**

- *Sampling is done with replacement on the original data and new datasets are formed.*
- *The new data sets can have a fraction of the columns as well as rows, which are generally hyper-parameters in a bagging model.*
- *Taking row and column fractions less than 1 helps in making robust models, less prone to over-fitting.*

2.**Build Multiple Classifiers:**

- *Classifiers are built on each data set.*
- *Generally the same classifier is modeled on each data set and predictions are made.*

3. **Combine Classifiers:**

- *The predictions of all the classifiers are combined using a mean, median or mode value depending on the problem at hand.*
- *The combined values are generally more robust than a single model.*

2.9 Random Forest Algorithm

Rather than building a single decision tree, random Forest approach creates multiple trees. To classify a new object based on attributes, each tree gives a classification result or votes. Then these votes are aggregated to predict the category from inputs. To put it in another way, it takes the average of outputs by different trees. According to [44],it works in the following manner. Each tree is planted and grown as follows:

1. Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.

2. If there are M input variables, a number m < M is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.

3. Each tree is grown to the largest extent possible and there is no pruning.

4. Predict new data by aggregating the predictions of the number of trees (i.e., majority votes for classification, average for regression)

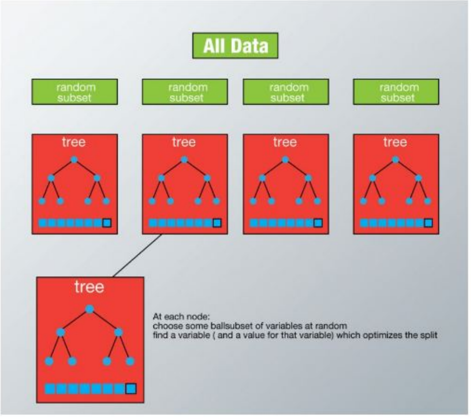

All Data
random subset
random subset
random subset
random subset
tree
tree
tree
tree
tree
At each node:
choose some ballsubset of variables at random
find a variable (and a value for that variable) which optimizes the split

FIGURE 2.7: Random forest aggregates the predictions of the number of trees(Source:[44]).

Estimated Performance

Predicted result for each bootstrap sample taken from training data. There will be samples left behind that were not included. These samples are called Out-Of-Bag samples or OOB. The prediction of each model on its left out samples when averaged can provide an estimated accuracy of the bagged models. This estimated performance is often called the OOB estimate of performance. These performance measures are reliable test error estimate and correlate well with cross validation estimates.

Variable Importance

When baggeing is done, we can say that decision trees have been build already. Then we can calculate how much the error function reduces for a variable at each split point.

Reduction of error can be averaged across from all decision trees and output to provide an estimate of the importance of each input variable. When there is high reduction in error then that variable has high importance.

Chapter 3

State of the Art

In this chapter, we are going to do a short review of three papers that cover specific aspects. First one is about general dataset we are going to use which is the flicker one which explains where is the origin of the data. Second one is an example of tourist based data analysis using data from another source; in this case it is Twitter. And the third one is a paper which is similar to this work that used the same data Yahoo100M to study mobility in United Kingdom. It is the idea similar to ours but in the different contexts with different goals.

3.1 YFCC100M: The New Data in Multimedia Research

In this part we are going to present general overview of YFCC100M data set from [8]. In modern ages, photograph tells a diverse story of people. People usually take photographs by the guidance of their variety of thoughts. Technological developments allow them to view, share, and interact on the device that captured them. Several publications and statistical reports [45, 46], have appeared in recent years documenting as of 2013, to Facebook alone more than 250 billion photos had been uploaded and on average received more than 350 million new photos each day. The literature on this paper reveals a large number of shared digital media objects have been uploaded to services like Flickr and Instagram. We can find their metadata which provides a convenient platform for finding solutions of many research questions at different field.

Authors has discussed that, collectively multimedia data serves as knowledge beyond what is captured in any individual snapshot. Moreover, provides information on trends, evidence of phenomena or events, social context, and societal dynamics.

Yahoo Webscope program is the root source of Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M) which came into existence in 2014. It is considered as a repository of stimulating and scientifically advantageous dataset. Our experiments and analysis were carried out with metadata of this multimedia data. According to the authors, YFCC100M is the largest public multimedia collection ever released, with a total of 100 million media objects, of which approximately 99.2 million are photos and 0.8 million are videos.

Inside the metadata, each media object has the attributes such as Flickr identifier, user who created it, camera that took it, time it was taken and uploaded, location where it was taken and so on. However, people always make question such as camera clocks camera clocks are not always set to the correct date and time. Moreover, some photos and videos may be found captured that have high time difference. In this case, authors has shown an interesting approach to address this issue.Figures 2 [8]depicts the moment of photo taken and upload time of photos and videos during the period 2000–2014, or 99.6% of the media objects in the dataset.

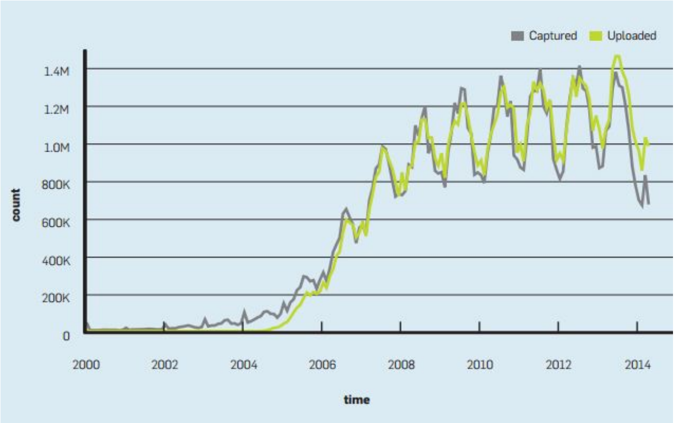


FIGURE 3.1: Number of captured and uploaded media objects per month in the YFCC100Mdataset, 2000–2014; the number of uploads closely follows the number of captures, with the number of more-recent uploads exceeding the number of captures as older media is uploaded (Source: Thomee et al.[8]).

According to authors, this dataset covers 249 different territories (such as countries and islands) and includes photos and videos taken in international waters and international airspace (see Figure 3.1).

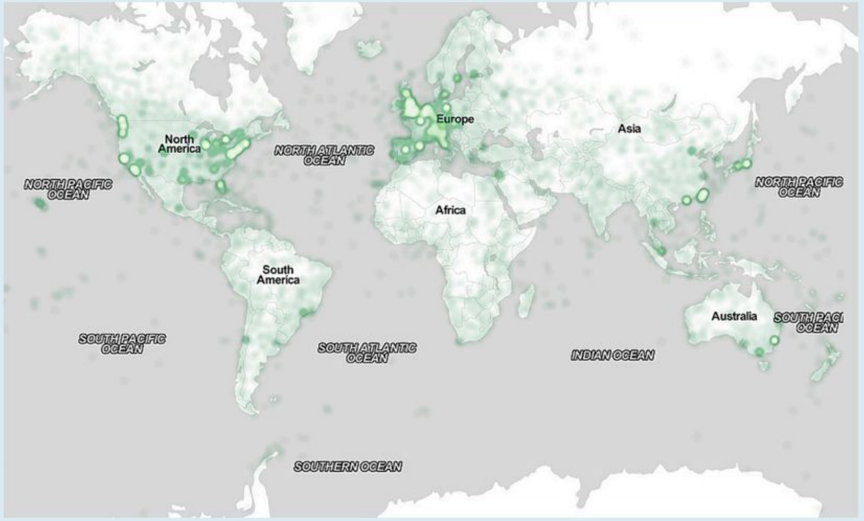


FIGURE 3.2: Global coverage of a sample of one million photos from the YFCC100M dataset; One Million Creative Commons (Source: Thomee et al.[8]).

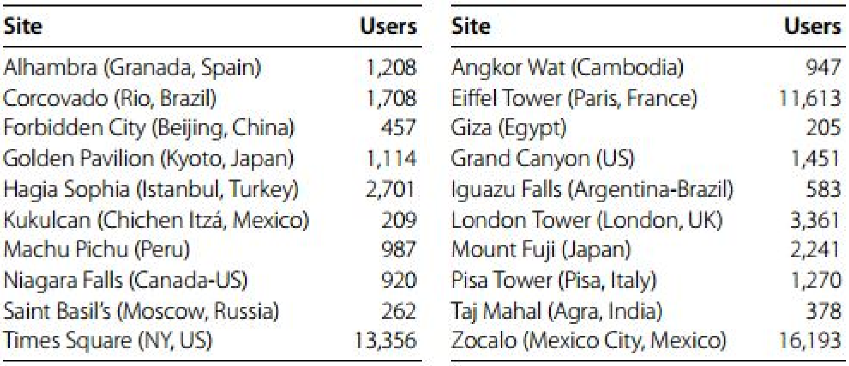
3.2 Touristic site attractiveness seen through Twitter

In this subsection we are going to present a recent paper by Aleix Bassolas et al. [10]. They apply techniques to measure the attractiveness of 20 of the most popular touristic sites worldwide using geolocated tweets as a alternative for human mobility. They select 20 out of the most popular touristic sites of the world and investigate their attractiveness using a dataset containing about 10 million geolocated tweets. They claimed that their analysis have demonstrated efficiency as useful source of data to study mobility at a world scale along with other researchers finding [16, 18].

They proposed three types of ranking of touristic sites. First, they ranks the touristic sites regarding the spatial distribution of the visitors place of residence. Second, they study the touristic site's but this time they consider local residence of that country. Finally, they conclude their analysis by concentrating on users detected in more than one site and explore the relationships between the 20 touristic sites by building a network of undirected trips between them.

Discretization of the space by dividing the world into squares of equal area (100 x 100 km^2) using a cylindrical equal-area projection enables them to identify a user's place of residence by a user as the cell from which he or she has spent most of his/her time.

To validate that this most frequented location is the actual user's place of residence, they inject the constraint that at least one third of the tweets has been posted from this location. Figure 3.3 shows the total number of users obtained by their method.



Site	Users	Site	Users
Alhambra (Granada, Spain)	1,208	Angkor Wat (Cambodia)	947
Corcovado (Rio, Brazil)	1,708	Eiffel Tower (Paris, France)	11,613
Forbidden City (Beijing, China)	457	Giza (Egypt)	205
Golden Pavilion (Kyoto, Japan)	1,114	Grand Canyon (US)	1,451
Hagia Sophia (Istanbul, Turkey)	2,701	Iguazu Falls (Argentina-Brazil)	583
Kukulcan (Chichen Itzá, Mexico)	209	London Tower (London, UK)	3,361
Machu Pichu (Peru)	987	Mount Fuji (Japan)	2,241
Niagara Falls (Canada-US)	920	Pisa Tower (Pisa, Italy)	1,270
Saint Basil's (Moscow, Russia)	262	Taj Mahal (Agra, India)	378
Times Square (NY, US)	13,356	Zocalo (Mexico City, Mexico)	16,193

FIGURE 3.3: Number of valid users by touristic sites(Source: Aleix Bassolas et al.[10]).

Authors considered two aspects to measure the attractiveness of a touristic site based on the spatial distribution of the places of residence of users who have visited this site.

- **Radius:** Average distance between the places of living and the touristic site. Applying the Haversine formula between the latitude and longitude coordinates of the centroids of the cells of residence and the centroid of the touristic site they computed the distance.

- **Coverage:** It represents the area encompassed by the users' places of residence computed as the number of distinct cells (or countries) of residence.

Figure 3.4 (a,b) depicts the ranking of the touristic sites according to the radius of attraction based on the distance traveled by the users from their cell of residence to the touristic site and the area covered by the user's cells of residence.

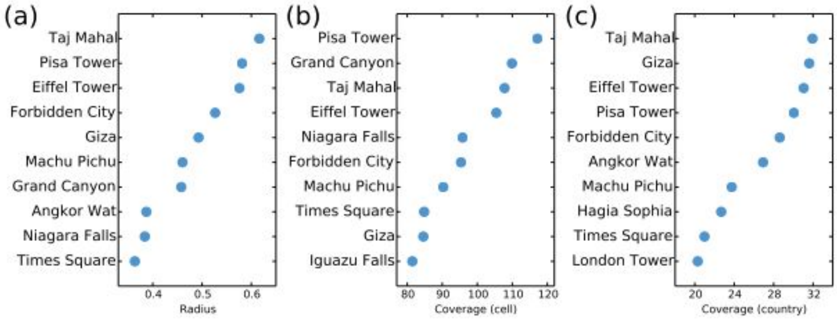


FIGURE 3.4: Ranking of the touristic sites according to the radius and the coverage. (a) Radius. (b) Coverage (cell). (c) Coverage (country).
(Source: Aleix Bassolas et al.[10]).

To verify their findings, they averaged results over 100 random selections of 200 users. Accuracy of the results were checked with different sample sizes (50, 100 and150 users), and their claim is that they obtained globally the same rankings for the two metrics. Both of their measures are much correlated and for most of the site the absolute difference between the two rankings is lower or equal than 2 positions. Nonetheless, their metrics are responsive to slightly different information; both rankings also display some difference. For instance, the Grand Canyon and the Niagara Falls show a high indemnity because of a high number of visitors from many specific places in the US but a low radius of attraction at the global scale.

In addition, they also found the origin of the visitors. In Figure 3.5, it can be seen that the visitors of the Grand Canyon are mainly coming from the US.

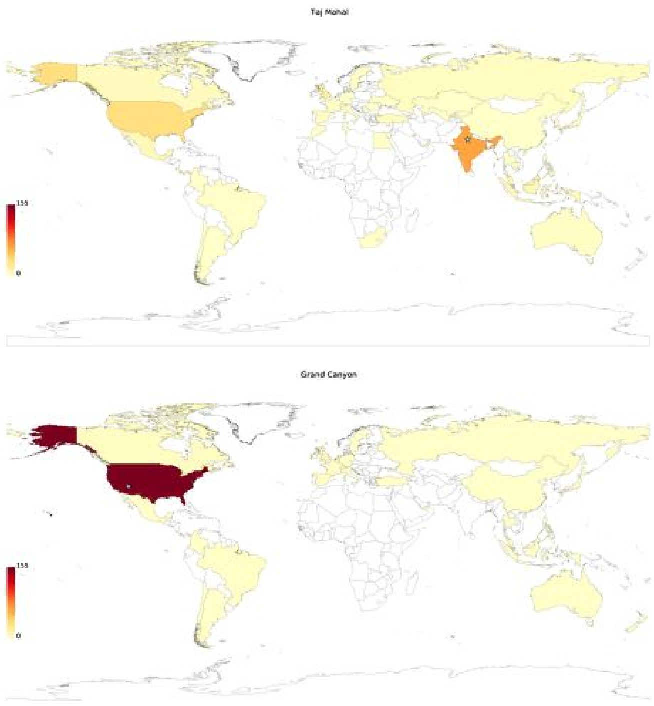


FIGURE 3.5: Heat map of the spatial distribution of the visitor's country of origin for the Taj Mahal and the Grand Canyon(Source: Aleix Bassolas et al.[10]).

In the second step of the experiment, authors intend to find touristic site's visiting figures by country of residence. They continue by performing a hierarchical cluster analysis to group together countries exhibiting similar distribution of the number of visitors according to the touristic sites. Authors used the ascending hierarchical clustering method with the average linkage clustering as agglomeration method and the Euclidean distance as similarity metric.

Figure 3.6 depicts the result of their clustering analysis. They could obtain two natural clusters emerged from the data. Surprisingly they found countries which tend to visit in a more significant way touristic sites located in countries belonging to their cluster. We can observe that first cluster gather countries of America and Asia whereas the second one is composed of countries from Europe and Oceania.

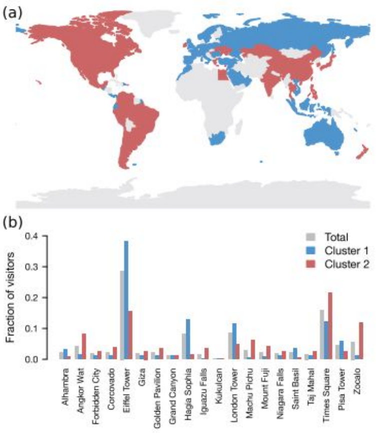


FIGURE 3.6: Clustering analysis. (a) Map of the spatial distribution of the country of residence according to the cluster. (b) Fraction of visitors according to the touristic site(Source: Aleix Bassolas et al.[10]).

In the last part of their work, authors try to establish some relationships between touristic sites based on the number of Twitter users who visited more than one site in trajectory of time. To show the relationship they generated undirected spatial network graph through which every link between two touristic sites represents at least one user who has visited both sites.

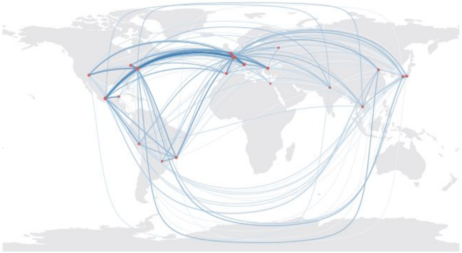


FIGURE 3.7: Network of undirected trips between touristic sites. The width and the brightness of a link is proportional to its weight. The size of a node is proportional to its weighted degree(Source: Aleix Bassolas et al.[10]).

By summing up their work, they try to show about the capability of geolocated data to provide global information regarding leisure related mobility. The data and the application methods could be completely general and can be applied to a large range of geographical locations, travel purposes and scales.

3.3 Modeling Human Mobility Patterns Using Photographic Data Shared Online

The most interesting approach to this issue has been proposed by Daniele Barchiesi et al. [9], they apply density based clustering to track user's points of movement, use machine learning algorithm to deduce the probability of locate people in geographical locations and the probability of movement between pairs of locations.

They also used Yahoo100M data as we used in our work. They extracted about 16000 individuals who uploaded geo-tagged images from locations within the UK to the Flickr photo-sharing website. Authors endeavors was to model behaviors of peoples moving pattern by clustering the geo-tagged information for each user into local groups of photos taken in distinct geographical areas, and studying the statistical properties of sequences of photos within and between clusters. To cluster groups of local geo-tagged pictures they applied DBSCAN algorithm.

Figure 3.8a illustrates the set of locations and the trajectory obtained from geo-tagged photos uploaded by a user, and Figure 3.8b displays the result of clustering. Six distinct clusters have been spotted by the DBSCAN algorithm and are located, from south-west clockwise, around Bristol, northern Wales, Glasgow, North York Moors National Park, Norfolk and Suffolk.

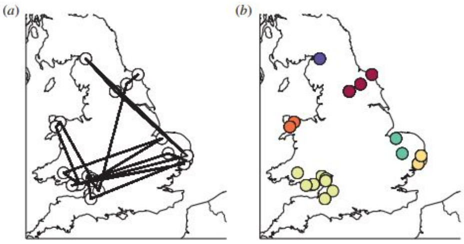


FIGURE 3.8: Model of an individual's mobility. (a) Individual trajectory depicting the location of geo-tagged photos uploaded by one of the users in the Flickr database. (b) Different colors indicate clusters discovered by the DBSCAN algorithm(Source: Daniele Barchiesi et al. [9]).

In addition, authors used the HMM model by setting a number of hidden states equal to the number of clusters identified by the DBSCAN algorithm, and by using the coordinates of the centroid of each cluster as the initial mean value of the corresponding Gaussian emission.

Figure 3.9 illustrates the model learned on the data depicted in Figure 3.8a. Authors drawn a number of observations from this model: firstly, the clusters identified by the DBSCAN algorithm have been retained by the HMM.

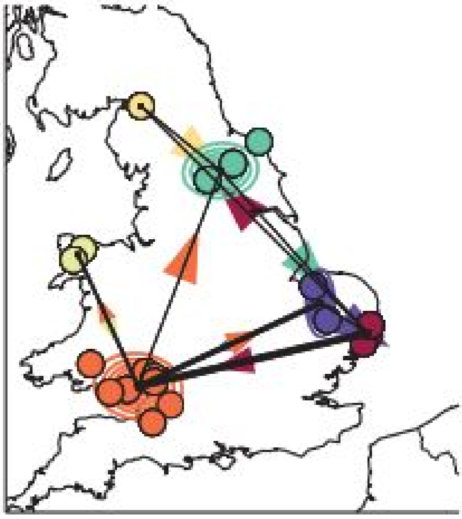


FIGURE 3.9: Different colors identify hidden states learned by a Hidden Markov Model, while the contour plots indicate Gaussian distributions learned for each state. The thickness of lines between different clusters is proportional to the number of times the user has moved between the two states, as estimated by the Viterbi algorithm. Arrows indicate the relative proportion of incoming and outgoing movements from one hidden state to the other(Source: Daniele Barchiesi et al. [9]).

Although, it is not guaranteed to be true in general, as DBSCAN only takes into account the spatial distribution of geo-tagged photos, whereas the HMM also incorporates information about the sequence of visited places that might determine a different mapping between locations and hidden states.

In the second phase, after analyzing the trajectory of a single user, authors derive aggregate results for all the users in the dataset. They try to deduce general patterns that describe the probability of finding any Flickr user in a given geographical area, and the probability of transition between pairs of areas. Figure 3.10(a) displays likelihood of finding a Flickr user in a given geographical location.

Figure 3.10(a) depicts aggregate model of mobility. Probability of an individual's location derived from data uploaded by all the users in the Flickr dataset.

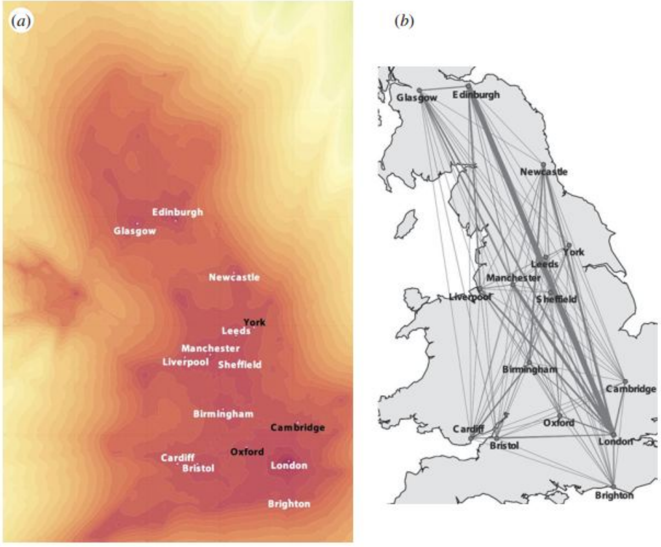


FIGURE 3.10: Aggregate model of mobility. (a) Probability of an individual's location derived from data uploaded by all the users in the Flickr dataset. (b) Aggregate transition probability between pairs of main UK cities(Source: Daniele Barchiesi et al. [9]).

The dataset contains photos uploaded in the UK; it simulates the shape of the UK. The points in the map are local maxima identified with a maximum filter and threshold, and correspond to the location of main UK cities. The names indicated in black indicate cities that do not appear in the list of the 20 most populous UK cities. On the other hand, Figure 3.11b depicts aggregate transition probability between pairs of main UK cities.

The line widths are proportional to the probability of observing a transition between any two pairs of cities, aggregated over all the users in the dataset.

Finally, authors compare Flickr estimates with official data. In Figure 3.11(a,b), number of journeys for each pair of cities of origin and destination as estimated by the model based on Flickr data and by the National Travel Survey (NTS) data. Each matrix displays values on a logarithmic scale from light blue (small number of journeys) to dark blue (large number of journeys).

However, according to the claim of authors, the evaluation of their method is sometimes difficult due to the lack of extensive official surveys on mobility at the country level. But their findings appear to be in general agreement with the evidence available, providing a novel statistical tool for the analysis of online data sources, and adding to the evidence that online data can be used to quantify human travel.

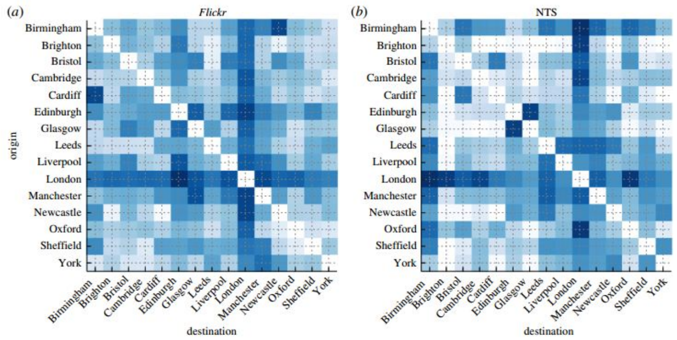


FIGURE 3.11: Comparison between Flickr estimates with official data(Source: Daniele Barchiesi et al. [9]).

Chapter 4

Data Characterization

4.1 Data Acquisition and Processing

This section aims at detailing our experimental data, how to acquire dataset, and pre-processing necessary for further analysis. First, we give concise introduction of dataset and context based on which this dataset has been designed. Second, we describe how do we acquire data, and finally we explain what sort of operations needed to prepare workable dataset to conduct supplementary analysis.

4.1.1 Data Acquisition

We downloaded dataset from the cloud in the form of ten separated zip file. In order to collect data, we selected to extract photos meta-data captured on Barcelona city. For the extraction of data, we developed a script that extracts data on the basis of latitude and longitude. In addition, our written script stores extracted data to Mongo DB database system. While processing data through our script we faced some memory issue. For instance, we tried to insert data to our database in batched fashion but we failed. Rather than the strategy of batch insertion, we had to insert each object as a single input.

We observed that overall process of data extraction and storing took approximately 12 hours with a single core i5 processor machine. It is worth remembering the whole dataset content can be processed in minutes to hours on a distributed computing cluster. But in the case of a single machine it might take a few hours to days. However, on the very first phase of data collection, we picked up attributes presented on Table 4.1.

photo_id	user_tags	license_url
user_id	machine_tags	server
Username	longitude	farm
date_taken	latitude	secret
upload_time	accuracy	original
camera_type	page_url	extension
Title	download_url	image_or_video
Description	license	license_url

TABLE 4.1: Attributes in YFCC100M data.

From our investigation we found the total size of the metadata for Barcelona city was about 240 MB. In this case, we used latitude range (41.302571, 41.43860847)and longitude range(2.029724,2.28858947) to extract media objects inside Barcelona.In addition, there was 288009 multimedia object's information inside that dataset. We can observe contents of a single media data from the following:

```
{"_id":{"\$oid":"576320f3634b825020bf6c63"},
"photo_id":"3706167951",
"user_id":"36449657@N00",
"username":"evas\%C3\%A8e",
"date_taken":"2009-07-10 17:25:56.0",
"upload_time":"1247222385",
"camera_type":"",
"title":"228+Mark+Jones+plays+Wall+of+Sound+Sonar+2009",
"description":"",
"user_tags":"evasee,jones,mark,sonar+2009",
"machine_tags":"",
"longitude":"2.1674",
"latitude":"41.383047",
"accuracy":"16",
"page_url":"http://www.flickr.com/photos/36449657@N00/3706167951/",
"download_url":"http://farm4.staticflickr.com/3505/3706167951_37c031fcef.jp",
"license":"Attribution-NonCommercial-ShareAlike License",\\
"license_url":"http://creativecommons.org/licenses/by-nc-sa/2.0/",
"server":"3505",
"farm":"4",
"secret":"37c031fcef",
"original":"abee1b73e2",
"extension":"jpg",\\
"image_or_video":"0"}
```

In order to verify the validity of our data extraction method, we carried out an experiment with help of a tool named CartoDB. CARTO is Software as a Service (SaaS) cloud computing platform that provides GIS and web mapping tools for display in a web browser [47]. They provide services through their location intelligence system to visualize geo-localized data. To investigate, we extracted latitude and longitude attribute's value from each of the media object and uploaded them as file to their system. As a result, we obtained a map containing our dataset depicted on Figure 4.1.

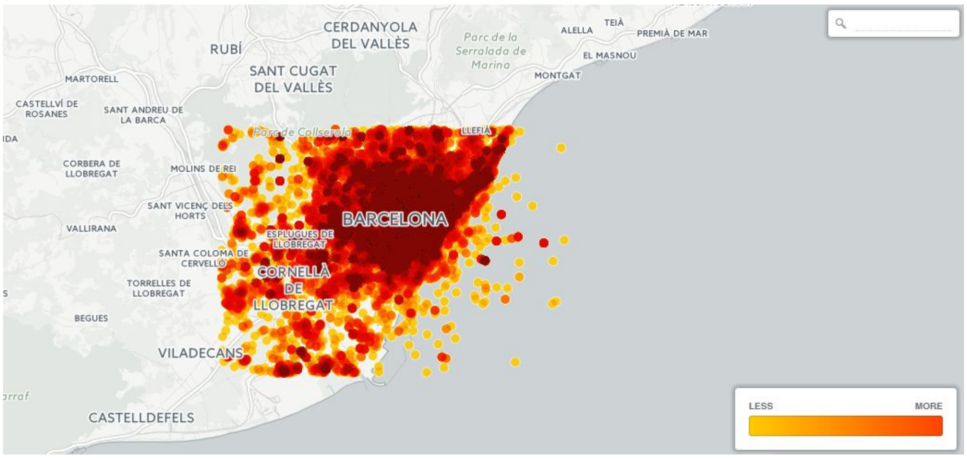


FIGURE 4.1: Verification of our extracted data; Collection of media object inside Barcelona.

After studying the Figure 4.1 carefully, we were confirmed that we have collected subset of Yahoo Flicker data appropriately. From CartoDB map we see that all of the media objects lie inside Barcelona city.

4.1.2 Data Pre-Processing

Preprocessing is an important stage of the knowledge extraction process in which dataset is being reconstructed so that it can be used of further analysis. It is indispensable to put dataset into proper input form if we want to characterize, apply clustering and machine learning methods to predict mobility pattern of the people. Activity such as: removing unnecessary attributes from dataset, convert data into proper computational format, cleaning of noisy data, solving the problem of incompleteness and redundancies, discretization etc. can be considered as parts of preprocessing.

Eliminate Unnecessary Attributes

From the data acquisition section we have noticed, in order to do characterization, clustering and application of machine learning techniques we need eliminate unnecessary attributes from our data set. We kept attributes such as: _id, user_id, date_taken, longitude, latitude. Applying query to our database we executed attribute elimination.

Transform data into proper computational format

From our data set we observed all values of "date_taken" attributes are in simple text format. We investigated, according to the requirement of Mongo DB we need time taken of the picture capturing in standard ISO date format. Analysis like data characterization, application of other analysis becomes convenient with such format. To do that, we developed a script that converts all of simple text format to standard date format. For instance, our script took "2010-11-06 15:09:42.0" as input and provided ISODate format like 2010-11-06T15:09:42Z. Transformation of attribute like this yielded advantage to conduct further analysis.

Cleaning of noisy data

Cleaning of noisy data refers detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Moreover, identifying inaccurate parts of data, and finally, replacing, modifying, or deleting the dirty or coarse data [48].In our case, we were fortunate enough that we had very few noisy data. We found some noisy data value inside 'date_taken' attribute .We eliminated all these noisy data from our dataset because we were facing some computational error in our data characterization section.

Manipulation of redundant data

Data redundancy is the existence of data that is additional to the actual data and permits correction of errors in stored or transmitted data. Although, additional data can simply be a complete copy of the actual data [48]. Rather than handling redundant data in pre-processing stage, we dealt with that while conducting supplementary analysis.

Processing Numerical Values

While attributes have real or integer domain are said to be numerical attribute. These domains are crucial to handle by human perception. Knowledge representations become intuitive if they are handled properly. Thus, they turn into useful, if they are based on as small domains as possible. To obtain intuitiveness we need to apply discretization techniques. Discretization is a technique of transforming numerical attributes into discrete, general ones. It divides original domain of the numerical attribute into a certain number of cut-points and assigning certain symbolic codes to those cut-points.

There are two numerical attributes: latitude and longitude in our experimental dataset. We noticed that they did not have same number of digits after decimal point. We found, when we were dealing with clustering we needed to have latitude and longitude attribute value of same numerical precision. To apply discretization technique, we developed a script to run on our MongoDB shell. Following example gives an overall idea about appearance of our preprocessed dataset.

```
{"_id":{"\$oid":"576451dda0cc8de40be3371b"},
"user_id":"10006319@N00",\\
"date_taken":"2013-07-31T18:54:43.000Z",
"longitude":"2.171473",
"latitude":"41.384600",
}
```

4.2 Data Characterization

In the context of knowledge extraction, characterization of data is used for generating descriptive parameters that effectively describe the characteristics and behavior of a particular data item [48]. Later, it can be used to find patterns, clusters and trends without incorporating class labels that may have biases. In order to have a general overview of the data we collected, we want to present some statistics and charts with all of our experiment conducted.

Presented statistics will let us know about how many total pictures there in our dataset, how many unique Flicker users who captured all those pictures in Barcelona. Furthermore, it will give potential information about distribution of picture taken by users, estimation of staying period in Barcelona and so on.

From our collected dataset, we have found out there are **288009** pictures and **7320** unique users who captured all those pictures. It will be interesting to know about distribution of staying period of users in Barcelona measured in terms of days. To generate histogram we made a data frame that consist user identifier and staying period (in days). In order to calculate staying period, we had to find out the time taken of first picture and last picture of each user. Then from these two time period we deducted the duration of stay of each user. Following Table 4.2 shows a sample data frame of users staying period in Barcelona city.

User_id	Staying duration
11327011@N00	1
79387032@N00	1
16210667@N02	3
103929680@N03	1
11929105@N00	1
13282784@N00	59
42257695@N04	31
35396334@N05	376
42439038@N00	8
9069821@N06	1
7249631@N04	1
74785688@N00	1
25638116@N00	2
8433185@N03	1
96488378@N04	125
98318718@N00	12
57761496@N02	55
47732493@N00	3
78535190@N00	3
94318878@N00	1
10266245@N04	1
88499020@N00	4

TABLE 4.2: Sample data:users staying duration in Barcelona.

From this data frame and we plotted the histogram depicted on Figure 4.2. There were 7320 unique users who had been visiting in Barcelona. Figure 4.2 represents that the distribution of users staying period is skewed right. It outlines, about 5000 users stayed in Barcelona for one day. Then we can see, second major portion of user's staying period is between 3-12 days. After that, we notice a downward tendency on users staying period. However, very few users staying period are more than 100 days. Finally, we can identify some outliers who stayed more than two years.

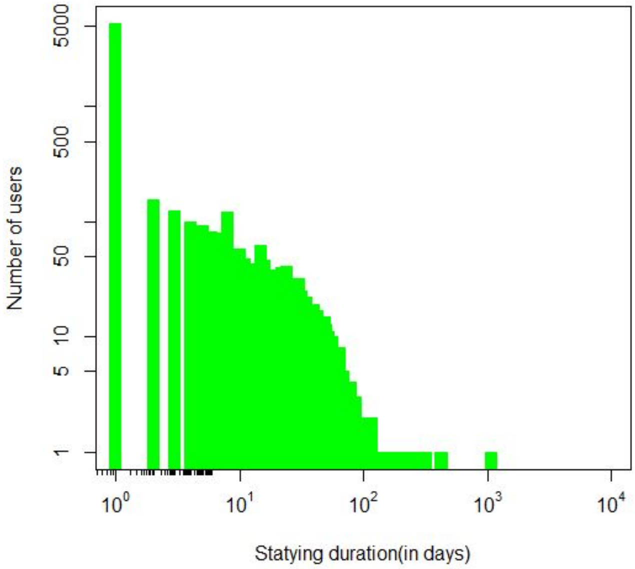


FIGURE 4.2: Overall distributions of users regarding their staying duration.

Now we want to investigate, what is the average number of picture a user takes. In order to predict user's mobility pattern, it is very important to know about the distribution of pictures taken by users. This distribution will guide us to consider only those users, who have taken pictures above a certain value. To put it another way, we can deal with only those users who has taken more than three or four pictures in different spots in Barcelona. To that end, it will enable us to get those users data that contains enough history of individual user. That said, data yielded by this observation will be used to predict overall movement pattern of users. We started by preparing a data frame that contains user identifier and total number of pictures a user has taken. A sample of data frame is shown on Table 4.3.

user_id	total_picture
49022038@N06	2
51035677132@N01	47
22214399@N06	145
54459811@N07	1
34279272@N00	1
92725684@N08	8
72679966@N00	312
95674706@N00	98
65207447@N00	39
62439258@N08	68
36449657@N00	1317
31226641@N00	26
64565145@N00	49
52206424@N00	29
13282784@N00	44
53048790@N08	43
23739340@N02	68
46426144@N05	3
20927180@N07	1
92854825@N00	1
61698672@N03	1
10416006@N05	1

TABLE 4.3: Example data of each user and his total captured photos.

We plotted histogram by using the data frame explained above. Consider Figure 4.3, which plots number of users against number of pictures taken in Barcelona.

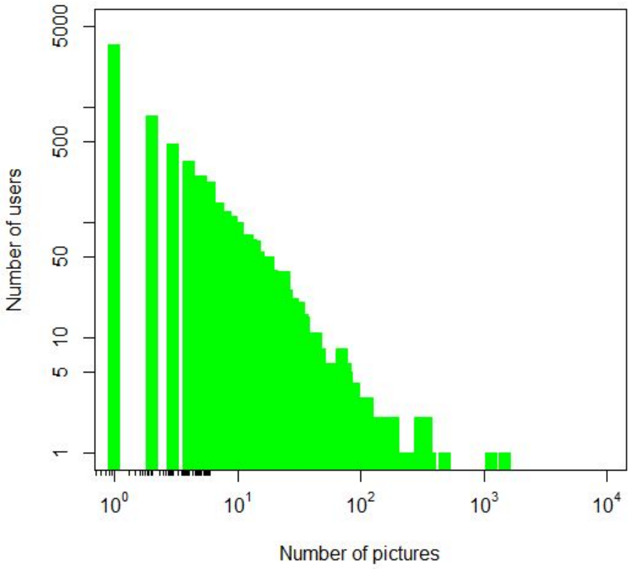


FIGURE 4.3: Histogram: reveals distribution of users regarding to their captured pictures.

Again, there were 7320 unique users who have taken of total 28009 pictures. From this figure it can be seen that, the distribution of number of users against number of pictures taken skewed right. A closer look at the graph indicates that, significant amount users have taken only one picture. Notably, second large portion of user's captured pictures about 3 to 10. We can also observe a gradual downward trend on number of picture taken by users. Moreover, there are very few users who have captured more than 100 pictures. Also, we can identify some outliers who took more than thousand pictures.

We wanted to check more broadly about how many of users in Barcelona stayed a particular amount of time and captured a specific amount of pictures. To put it another way, we wanted to find precisely, how many users stayed 1-2 days and captured 1-5 pictures and so on. In previous histograms, we have found overall idea about users staying duration and number of pictures taken in Barcelona. But we wanted to observe these both aspects precisely in a single structure. For this reason, we prepared dataset illustrated in Table 4.4.

	1 day	1 week	>1 year
1–5	2491	64	248
6–10	203	52	176
11–15	109	23	108
16–20	65	22	79
21–25	34	22	68
26–30	24	20	63
31–35	28	10	41
36–40	14	11	40
41–45	18	10	38
46–50	12	9	29
51–55	8	9	30
56–60	11	9	15
61–65	4	2	26
66–70	3	9	23
71–75	5	6	13
76–80	4	1	16
81–85	2	3	15
86–90	8	4	20
91–95	2	5	14
96–100	2	3	11
>100	16	72	325

TABLE 4.4: Dataset: exact amount users in terms of their photo taken and staying duration.

To explore more conveniently, we have generated a heat map from this table dataset that is shown in Figure 4.4.

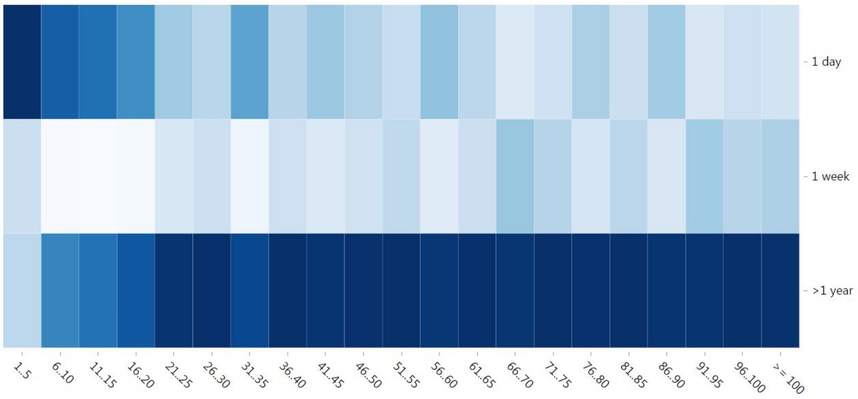


FIGURE 4.4: Heatmap- Quantity of user is represented by color, x axis projects number of pictures taken and y axis reveals staying duration in Barcelona.

First, from Table 4.4, we will observe users who captured pictures in range 1-5. We can see, about 2491 of users stayed for one day and this cell was dark blue. Notably, there are 64 users who stayed for one week and 248 users have staying durations of over a year. In the heat map, this can be seen as light blue color. The numbers of users tend to decrease gradually, as the color turns lighter from up to down. Secondly, if we consider ranges: 6-10, 11-15 and 16-20, we can notice that 203, 109, 65 users stayed for one day respectively and 176, 108, 79 user's staying duration was more than one year. Comparatively, small numbers of users stayed for a week in those ranges. Finally, significant amounts of users stayed over the year that captured more than 21 pictures. It seems, all the dark blue cells on last row represent the high number of users resides in that group.

For exploring more concretely, we prepared data set that helps us to classify the amount of users who took pictures between range: 1-10, 11-20, 21-30............and greater than or equal to 100. In addition, we captured more time duration for the case of users staying period. In Table 4.5, we can see staying durations are classified as 1-2 days, 1-2 weeks and more than one year. Doing so, we can differentiate the users who stayed Barcelona for a long time or potential tourists.

	1-2 day	1-2 week	>1 year
1–10	2972	105	424
11–20	262	39	187
21–30	108	33	131
31–40	62	16	81
41–50	49	14	67
51–60	30	10	45
61–70	26	7	49
71–80	21	6	29
81–90	17	3	35
91–100	9	6	25

TABLE 4.5: Dataset shows exact amount users in terms of their photo taken and staying duration.

Figure 4.5 depicts that, about 2972 users stayed for 1-2 days and captured images between 1 to 10. Dark blue cell of the heatmap indicates highest numbers of users are within these range. Also, 424 users stayed over the year and took pictures between 1-10. Then, we can observe about users who have photos between 11-20, large amount of users were regular tourist. There are also handsome amounts of users who may be the dweller of Barcelona have pictures same amount of photos. It is worth mentionable, there are very less amount users who stayed about 1-2 weeks and have photos more than 11. We can see white color rows in the middle of Figure 4.5 reveals that scenario more clearly. It is worth notable, there are mentionable number of users who captured more than 11 images or higher. Moreover, we can notice through the gradual increasing dark blue color from that heatmap.

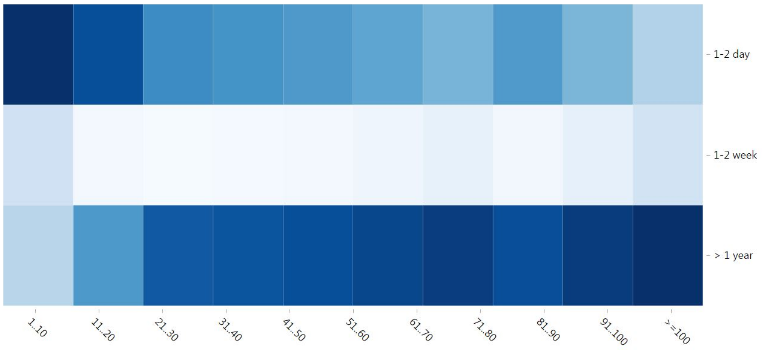


FIGURE 4.5: Heatmap- Quantity of user is represented by color, x axis projects number of pictures taken and y axis reveals staying duration in Barcelona.

We wanted to make exhaustive search to see the specific amount of users who captured photos between 1-2 and 3-5 and so on. Because this observation will help us to identify rejected users from our consideration. Digging these characteristics inside our data, we will be able to know about approximate amount of users, whose movement will be analyzed to predict mobility pattern. To do that, we made query to our data set in MongoDb and obtained following data table. Later, we wrote a script to generate a heatmap from that.

	1-2 day	1-2 week	>1 year
1–2	2166	24	69
3–5	505	38	179
6–10	301	43	176
11–15	158	24	108
16–20	104	15	79
21–25	64	17	68
26–30	44	16	63
31–35	36	8	41
36–40	26	8	40
41–45	24	6	38
46–50	25	8	29
51–55	13	4	30
56–60	17	6	15
61–65	13	2	26
66–70	13	5	23
71–75	14	2	13
76–80	7	4	16
81–85	7	3	15
86–90	10	0	20
91–95	3	2	14
96–100	6	4	11
>100	50	66	325

TABLE 4.6: Dataset reveals exact amount users in terms of their photo taken and staying duration. We considered more number of intervals of photo taken and staying period.

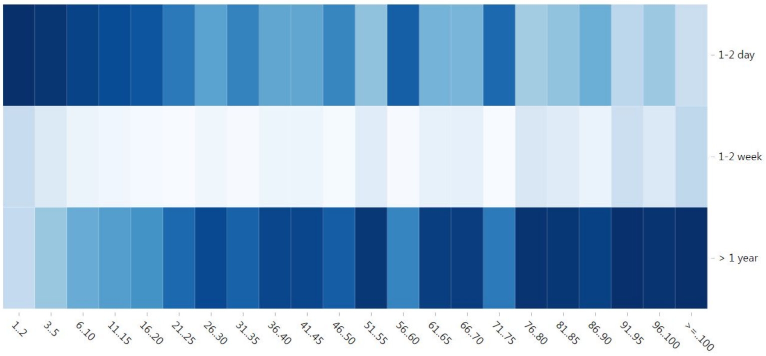


FIGURE 4.6: Heatmap reveals exact amount users in terms of their photo taken and staying duration. Blue color and white color indicates high and low intensity correspondingly.

Figure 4.6 illustrates, about 2166 users have only captured 1-2 pictures and they seem to be tourists. Dark blue cell in the upper left corner represents the high intensity of discarded users. But, there are handsome amount of users who have pictures more than three but most of them in the group of short or long stayer. In Figure 4.6, we can observe this scenario in first row and third row from range 3-5 to 26-30. Also, gradual dark blue cells on the third row reveals that noteworthy amount of users have images more than 30, although they are long stayer in Barcelona. Described characteristics of our data set indicate, we will get sufficient amount of users to make training and test data for mobility prediction experiment.

Chapter 5

Clustering Data

Knowing basic characteristics of the data from Section 4, next we try to study the data. In this section, we get into clustering, what are important places in Barcelona where people take a lot of pictures. We wanted to find groups of points that happened in the same places and then we put tag on those points. Ideally, we needed to identify automatically these areas of points where there are abundant of photos together.

In order to do that, we need to employ some sort of algorithm which takes about three hundred thousands of points, and returns important groups there. As we do not know the number of clusters we will obtain and the size of the clusters, this is why we selected DBSCAN algorithm, where we intended to study density based clustering, and then we will be able to group points that are closer in space. Thereafter, we launched DBSCAN on filter data and non-filter data.

5.1 Applying DBSCAN to Filtered Dataset

The aim of these experiments is to apply DBSCAN algorithm appropriately so that, most visited places will appear as natural clusters in Barcelona. We were interested about people who visit different places, not who take a lot of pictures in a single place. As we mention earlier in data pre-processing Section 4, we have found lot of users who have more than one picture in same latitude and longitude in our dataset.

We need to keep in mind, if a user takes 500 pictures in a place that does not mean that place is most popular. Rather than, if there are lot of distinct users take pictures in a place then this place can be considered as popular. To put it another way, assuming a user is moving on different part of the city, if he takes ten pictures in a place then we need to consider only one picture among them for that place.

After all, we want to know, which unique places a user had been visited. In order to get a single photo of each user in a place, we wrote a script that checks a photo of a user, and then checks next one. If that photo is within 50 m of geodesic distance of first one, then we discard the second picture of that user from our consideration. If the next picture is not within that radius of 50m, then we do not discard. In this way, we removed extra photos of a user from a single place.

Later filtering the data, we have found total number of photos in reduced dataset is about 78851. We prepared a .csv in which there were two columns: latitude and longitude. We used this file as input to the DBSCAN algorithm. Again with this .csv file we tried to visualize the reduced data set using CARTODB , and Figure 5.1 shows the scaled down users.

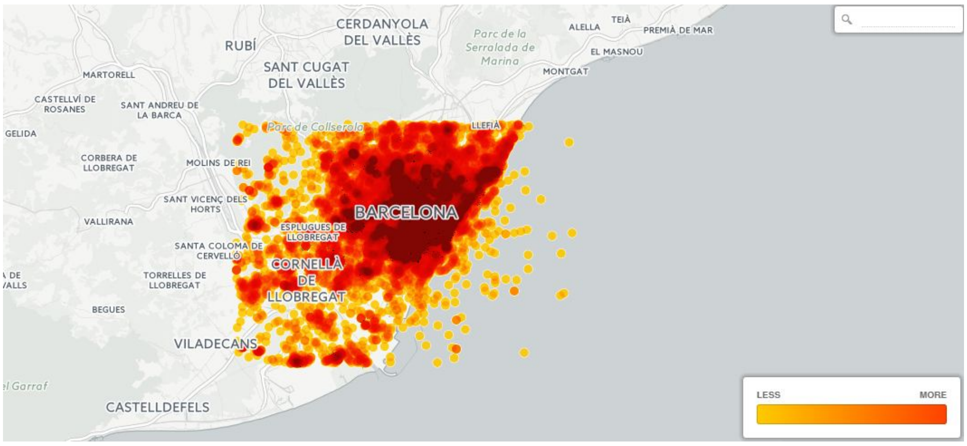


FIGURE 5.1: Filtered Dataset of Barcelona.

5.1.1 Experiment Result

To carry out experiment, we wrote a script where we called API of DBSCAN algorithm. In order to point out appropriate value of parameters eps and min-values for DBSCAN algorithm, we have conducted several numbers of experiments. Our point of interest is to find out eps and min value such that, we get reasonable number of clusters, and these clusters encompass justifiable number of users. We tried to investigate the experiment through visualizing clustered points, and calculating total number of clusters, averages number of users in each cluster and total numbers of users in all of these clusters. Red points depict the clustered points generated by DBSCAN algorithm, and black points represent all geo-tagged photos in Barcelona.

Among these results, we see that for eps = 0.1 and minpoints = 50 we get 45 important places in Barcelona and these clusters cover 81% of users. Also, we get 29 clusters for eps = 0.01 and minpoints = 50, but they only covers 15% of users.

Regarding each eps = 0.05, 0.04, 0.03 and minpoints = 20, 30 and 40 we obtain large number of clustered points. According to our aim, these values of parameter may not sound interesting. They generate many clusters that could represent too many hot spots for Barcelona city.

One of the experiment rsesult sample can be seen on Table 5.1. To observe it graphically we can take a look on Figure 5.2

	minpoints= 50	minpoints= 60	minpoints= 70
eps = 0.05	Cluster: 54 Avg Size: 904 Pictures: 62% = 48893	Cluster: 49 Avg Size: 938 Pictures: 58% = 45739	Cluster: 42 Avg Size: 1037 Pictures: 55% = 43373
eps = 0.04	Cluster: 54 Avg Size: 779 Pictures: 53% = 41796	Cluster: 55 Avg Size: 719 Pictures: 50% = 39430	Cluster: 46 Avg Size: 803 Pictures: 47% = 37064
eps = 0.03	Cluster: 55 Avg Size: 618 Pictures: 43% = 33910	Cluster: 50 Avg Size: 632 Pictures: 40% = 31544	Cluster: 40 Avg Size: 737 Pictures: 37% = 29178

TABLE 5.1: Notably, for each value of eps and minpoints we get legitimate clusters; But eps = 0.05 and minpoints = 70 seems to be promising parameter, because they generate 42 clusters and covers above half of the total users.

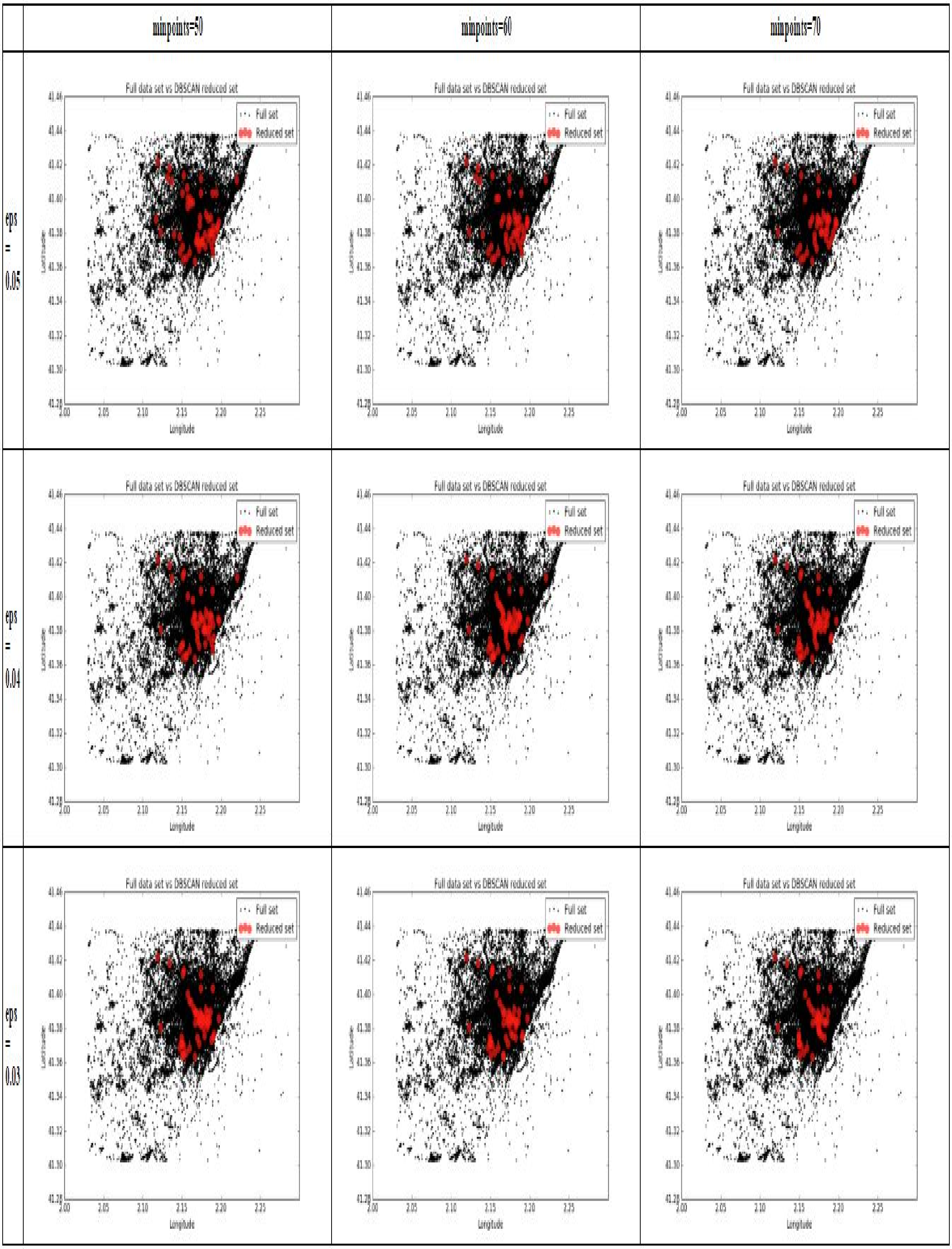


FIGURE 5.2: Visually, for most of eps and minpoints gives legitimate clusters; But third cell and 9th cell give the aspect of proper representation of hot spots.(Filter Data)

5.2 Applying DBSCAN to Non-filtered Dataset

Like filtered dataset, now we will apply DBSCAN algorithm to non-filtered dataset. We prepared a .csv in which there were two columns: latitude and longitude of all photos of Flicker user in Barcelona. We used this file as input to the DBSCAN algorithm. We will look for fitting value of eps and minpoints. Although the size of the non-filtered dataset is about four times of filtered data, and we found minpoints = 70 seem to be likely to get proper number of cluster in filtered data, thus we choose starting value of minpoints is four times of filtered data.

We do not know which eps and minpoints value will best fit of our interest, therefore we will investigate through gradual increasing of minpoints value. Detail of the experiment can be seen on Appendix A. To start the process we provide a data frame through a .csv file which contains two columns: latitude and longitude. From the following Table 5.2 and Figure 5.3 we can observe the total numbers of clusters we obtain, average number of users in each clusters and percentage of users covered in all these obtained clusters.

	minpoints= 900	minpoints= 950	minpoints= 1000
eps = 0.05	Cluster: 26 Avg Size: 4312 Pictures:39% = 112323	Cluster: 26 Avg Size: 4265 Pictures:39% = 112323	Cluster: 26 Avg Size: 4199 Pictures:38% = 109443
eps = 0.04	Cluster: 27 Avg Size: 3598 Pictures:34% = 97923	Cluster: 27 Avg Size: 3560 Pictures:33% = 95042	Cluster: 26 Avg Size: 3630 Pictures:33% = 95042
eps = 0.03	Cluster: 26 Avg Size: 3052 Pictures:28% = 80642	Cluster: 24 Avg Size: 3197 Pictures:27% = 77762	Cluster: 21 Avg Size: 3489 Pictures:25% = 72002

TABLE 5.2: Here, for each value of eps and minpoints we get small amount of clusters; But eps = 0.03 and minpoints = 1000 seems to be promising parameter, because they generate 21 clusters and covers about 72002 from the total users.

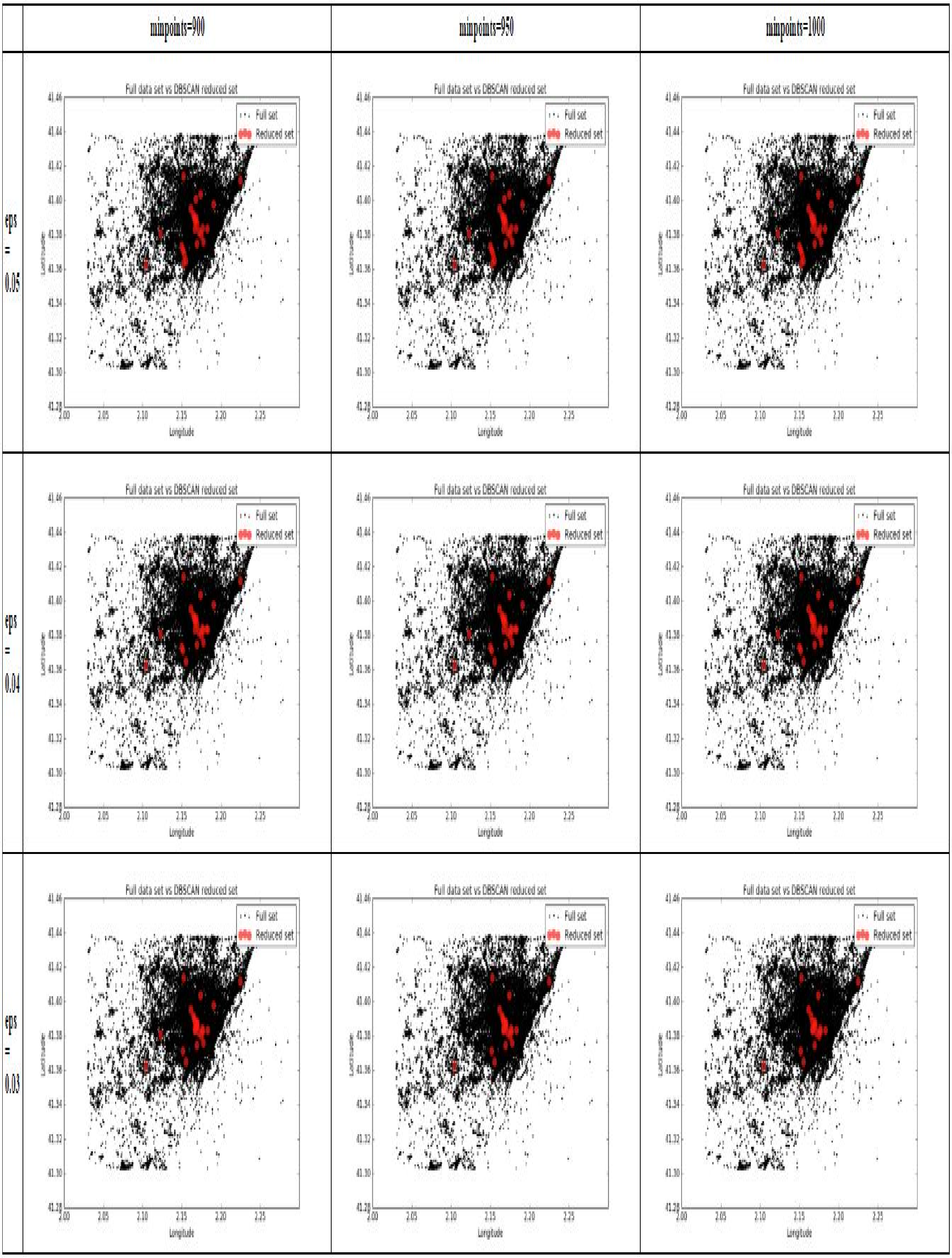


FIGURE 5.3: We can observe for most of eps and minpoints gives legitimate clusters; But third cell and 9th cell gives the aspect of proper representation of hot spots(Non-Filter Data).

5.3 Discussion

In this section, the discussion will point to selecting one of the experiment results to perform further analysis. In Section 5.1 we have found clusters based on filtered data where we may consider result generated for eps = 0.03 and minpoints = 70. Because, using these parameters we get 40 clusters as well as 37% users of filtered data

Parallel to that, we may look at eps = 0.05 and minpoints = 70 which render 42 clusters along with 55% users. Figure 5.4 represents clusters/hotspot founded based on filtered data where eps = 0.03 and minpoints = 70.

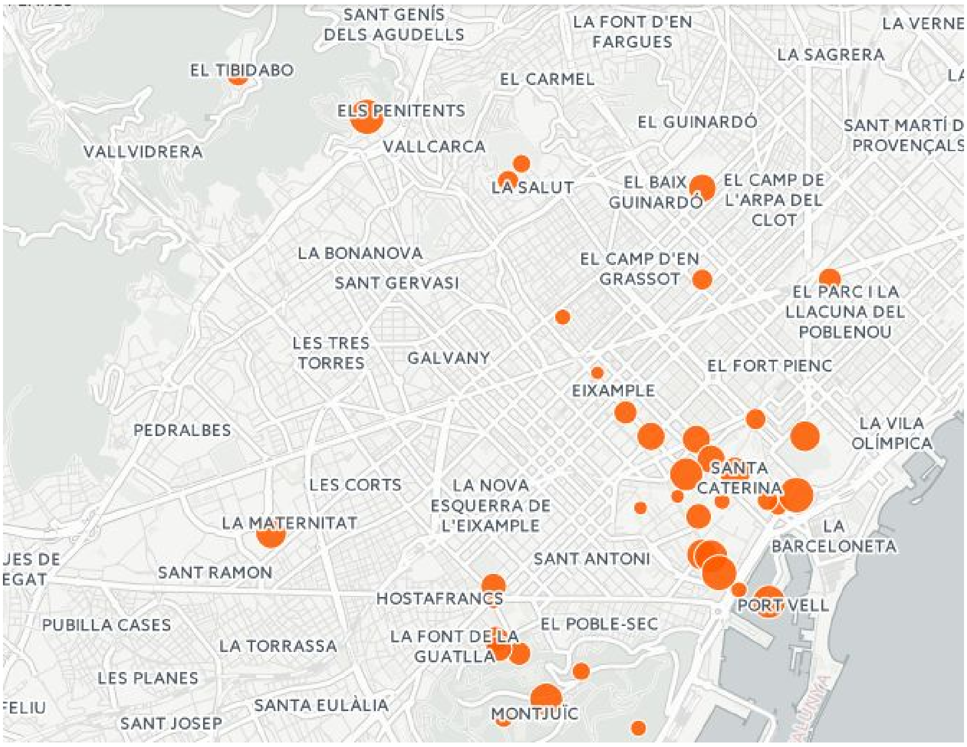


FIGURE 5.4: Clusters founded by applying DBSCAN algorithm on filtered data; Parameter values are eps = 0.03 and minpoints = 70.

The results thus obtained from Section 5.2, we can select eps = 0.03 and minpoints = 1000, because it gives us 21 clusters and covers 25% users from non-filtered data. From the observation of clustering on filtered data and non-filtered data, we can notice that we are obtaining acceptable number of densed place/clusters from non-filtered data. As mentioned earlier, we are interested in finding a shortlist of most visited places by Flickr users, thus we have decided to choose these parameter values. Alongside, we have the convenience to select other parameter values if we find lacking in our decision, as we have done exhaustive experiment in Section 5.2. We have visualized the hotspot through CARTODB depicted on Figure 5.5.

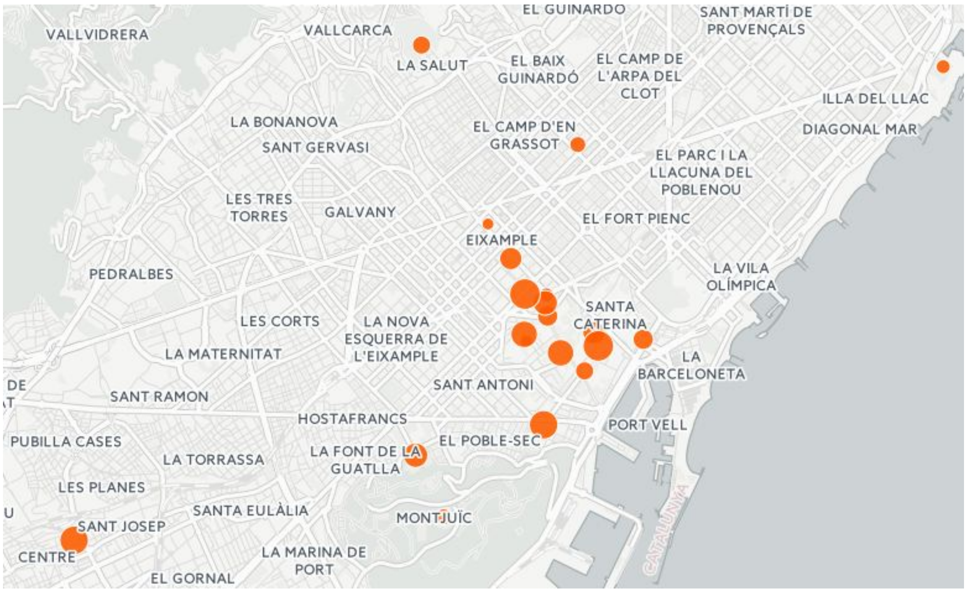


FIGURE 5.5: Clusters founded by applying DBSCAN algorithm on non-filtered data; Parameter values are eps = 0.03 and minpoints =1000.

We verified our findings with many of the surveys conducted by different renowned websites. They provide services to the tourist and recommend most popular tourist spots for new comer to Barcelona, although most of the surveys result came from users rating to different places of the city. We also take into account information provided in Wikipedia about Barcelona.

ClusterID	Longitude	Latitude	Place
0	2.167095	41.3831	Museum of Contemporary Art in Barcelona
1	2.161758	41.39531	Casa Milà
2	2.169885	41.38793	Plaza de Catalunya
3	2.224951	41.41167	Barcelona bosc urba
4	2.17422	41.40356	Sagrada Familia Botiga
5	2.155584	41.36482	Lluís Companys Olympic Stadium
6	2.152547	41.41392	Park Guell
7	2.175182	41.38001	Plaça Reial
8	2.170033	41.38572	La Rambla
9	2.1833	41.3833	Pla De Palau
10	2.176478	41.38401	Catedral de Barcelona
11	2.164933	41.39171	Casa Batlló
12	2.15171	41.37123	Font Màgica de Montjuïc
13	2.169713	41.38709	Plaza de Catalunya
14	2.166789	41.38382	Facultat de Geografia i Història
15	2.171838	41.3819	Mercado de La Boqueria
16	2.10431	41.36238	Centre Comercial La Farga
17	2.169477	41.3744	Carrer Nou de la Rambla
18	2.166881	41.38801	Gran Via de les Corts Catalanes
19	2.177063	41.38261	Ajuntament de Barcelona

TABLE 5.3: Clusters centroid founded by applying DBSCAN algorithm on non-filtered data; Parameter values are eps = 0.03 and minpoints =1000.

Our results coincide with places such as Sagrada Familia, Casa Milà, Casa Batlló, Museu Nacional d'Art de Catalunya (MNAC), Las Ramblas street, Park Guell, Placa De Cataluniya and so on(Table 5.3). We aggregated the entire tourist spots from different sources mentioned earlier, and we have found that our results converge for most of them. In addition, we can observe other popular places are found according to the number of photos taken on these places.

Chapter 6

Prediction of Mobility

In this step of our study we move into prediction through Flickr data. We intend to find a prediction mechanism through which we will be able to guess tourists mobility behavior in a city. Interestingly, predicting could be applied to observe a single persons as well as overall tourist's collective movement. For instance, we are interested to know if someone visited cluster 5, 6, 3, 9, then which could be the possible next movement.

We applied a machine learning approach named random forest classification to obtain prediction. In this part of our study we intend to compare prediction result with both non-filtered and filtered data as well as for multiple cluster sequences.

6.1 Prediction with Non-filtered Data

In order to apply random forest classifier, we will need to prepare a data set which will contain columns of long sequence of clusters. Prior to that, we analyzed it would be more rational if we can gather each users photos in a sequence. To put it differently, it will be more logical of having each user taken pictures tracked by a time-stamped trajectory. For instance, in our dataset we have observed cases where some users take a picture and then take a break more than a week even a month. This could result of generating moving sequence of a person irrelevant of general moving pattern, or this person could be a long stayer in Barcelona.

For this reason, we first sorted our data set by user_id and date_taken attribute of the picture, and then we track each picture with an attribute called trajectory. In this case, we check the time stamp of a picture of a person and then calculate the time difference between immediate prior photo of that person. If the time difference is less than or equal to 7 days then we mark it by same trajectory. In order to prepare data set like this we wrote a script in MongoDB, and Table 6.1 shows a sample of dataset tracked by trajectory.

User_id	Date_taken	Longitude	Latitude	Trajectory
22277135@N06	2001-08-19T19:56:16.000Z	2.167707	41.38787	T1
22277135@N06	2001-08-19T22:29:05.000Z	2.17045	41.38501	T1
22277135@N06	2001-08-19T22:29:16.000Z	2.17045	41.38501	T1
22277135@N06	2001-08-19T22:31:34.000Z	2.168608	41.38739	T1
22277135@N06	2001-08-19T22:32:11.000Z	2.168608	41.38739	T1
22277135@N06	2001-08-19T22:32:33.000Z	2.168608	41.38739	T1
22277135@N06	2001-08-20T00:33:09.000Z	2.169992	41.38717	T1
22277135@N06	2001-08-20T01:02:14.000Z	2.171137	41.38673	T1
22281620@N00	2005-04-20T14:58:02.000Z	2.112636	41.38793	T1
22281620@N00	2005-04-20T14:58:26.000Z	2.112636	41.38793	T1
22281620@N00	2010-02-03T12:14:03.000Z	2.158309	41.39253	T2
22281620@N00	2010-02-12T11:07:14.000Z	2.168941	41.43681	T3
22281620@N00	2010-09-15T19:30:05.000Z	2.172074	41.38189	T4
22281620@N00	2012-03-04T17:05:28.000Z	2.168716	41.38729	T5
22281620@N00	2012-04-27T16:36:07.000Z	2.17231	41.38442	T6

TABLE 6.1: A sample of dataset tracked by trajectory.

If we observe date_taken attribute from the first two row then we can notice that time difference between these two pictures is about two hours, as a result they are marked by same trajectory. In addition, we can notice that trajectory T2, T3, T4, T5, T6 contains only one picture in each of them. So it will be more rational to consider the photos belonging to trajectory T1, because they will reveal more appropriate moving pattern of that user.

Using the same parameter values as we selected for non-filtered data to find out hotspot in Barcelona, we now process those users that belong to 21 clusters. Through writing a script, we executed DBSCAN algorithm again for non-filtered data, but this time we write points inside each cluster with corresponding cluster id. Table 6.2 represents the sample data frame that we obtained through clustering.

User_id	Date_taken	Trajectory	Cluster_id
10100147@N06	2012-10-11T16:54:53.000Z	T1	2
10100147@N06	2012-10-13T05:40:42.000Z	T1	5
10100147@N06	2012-10-11T14:48:45.000Z	T1	6
10100147@N06	2012-10-11T17:06:35.000Z	T1	7
10100147@N06	2012-10-11T17:11:19.000Z	T1	7
10100147@N06	2012-10-11T14:55:01.000Z	T1	18
10100147@N06	2012-10-13T01:59:24.000Z	T1	6

TABLE 6.2: A sample of clustered dataset of Barcelona. We can observe, this user has traveled six places in the city. We can identify his movement through Cluster_id attribute. He visited cluster 2, 5, 6,7,18 and again 6.

It is worth mentioning, when we obtained all the points from 21 clusters, they were not organized as in Table 6.2. In order to make data frame like that, we wrote a script in MongoDB that sort the dataset by user_id and date_taken. After these above step, we move in the part where we would like to generate a data frame, this frame will contain user_id, trajectory and cluster_list attribute.

In order to generate, we need to write a script in MongoDB. In this script, we instruct MongoDB to build a cluster list for each trajectory of each user. To put it another way, if we consider data in Table 6.2 we see that this user has cluster list 2,5,6,7,7,18 and 6 for trajectory T1. At the same time he could have cluster list 18, 9 for trajectory T2. From this user data, as we sorted by date this is why we find that this person has taken two photos at almost same time in cluster 7. In such kind of cases, to avoid ambiguity we discard cluster_id if they appear in consecutive order. On the other hand, although this user has visited cluster 6 two times but they are not consecutive because he visited 7, 18 before visiting 6 again. Table 6.3 displays a sample of cluster list for each user.

user_id	Trajectory	Cluster_list	Total Cluster
10100147@N06	T1	6,18,2,7,6,5	6
10226584@N06	T1	5,1,10,7,2	5
10297518@N00	T1	0,5	2
10297518@N00	T2	9,0	2
10297518@N00	T3	8,7,0,5,1,14,9	7
10585427@N00	T21	13	1
10585427@N00	T30	12	1
10913760@N04	T1	1,16,13,1,6,9,0,6	8
10982500@N03	T1	0,9,1,5,0	5
11468468@N00	T1	10,1,6,2,5,8,14,9,0,10	10
15889861@N00	T1	10,9,0,6,7	5
15889861@N00	T3	5	1
15911882@N02	T1	13	1
15920034@N04	T1	13	1
15992105@N08	T1	5	1

TABLE 6.3: A data sample of cluster list for each trajectory of each user.

We want to look on the distribution of cluster list so that we can have an idea about overall dataset to use for prediction. To do that we generated a histogram and Figure 6.1 depicts the distribution of the cluster list against trajectory.

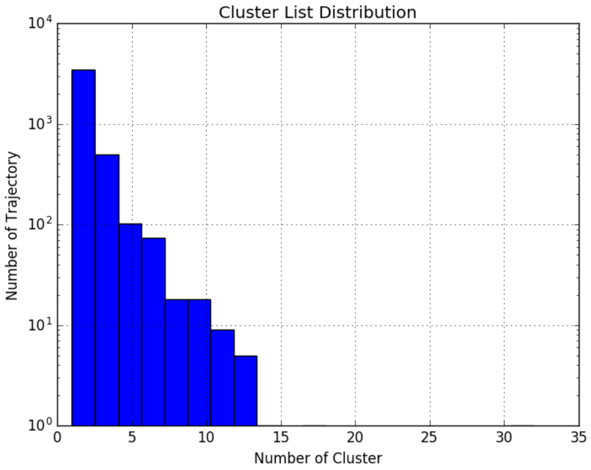


FIGURE 6.1: Distribution of length of cluster sequence against number of trajectory for non-filtered data.

According the Figure 6.1 we can observe that a significant number of trajectories contain cluster sequence of size between 1-2, whereas more than 500 trajectory contains cluster sequence of size between 3-4. Finally, we can notice that more than few hundreds of trajectory have cluster sequence greater than or equal five clusters. It is essential to know exactly the number of cluster sequence greater than or equal to five, because we intend to obtain data set of at least four inputs and 1 output. To put it in a different way, we start our experiment of prediction where observing consecutive four cluster the algorithm will guess what could be the next cluster a person probably goes. We counted from our dataset that there were 230 trajectory containing more than or equal to 5 cluster.

It will be interesting to observe what the prediction rate is when we employ random forest algorithm. Table 6.4 shows the correct prediction rate with four inputs and one output.

Number of Tree	Number of Feature	Correct Prediction Rate(%)
500	1	21.5
	2	21
	3	21
	4	20.5
600	1	22
	2	21
	3	23.5
	4	21.5
700	1	20
	2	21
	3	20
	4	21
800	1	20
	2	21.5
	3	20
	4	21
900	1	20.5
	2	21.5
	3	20
	4	21.5
1000	1	20.5
	2	20
	3	21
	4	20

TABLE 6.4: Prediction of movement using random forest classification algorithm. There were 200 training data and 30 test data.

We can observe from Table 6.4, using random forest algorithm we obtain average prediction rate about 20 %. We want to investigate what happens if we try with five clusters as input and then try to predict the sixth one. After executing a query to MongoDB we found that there were 127 trajectories containing a cluster sequence of length 6. We employed random forest and found the average prediction rate is about 20.56%.

In this stage of our experiment, we wanted to see what happens if we sort the cluster sequence. To say it differently, currently in each trajectory we have cluster sequence like 9, 5,0,2,7 and we want to transform this sequence as 0, 2, 5, 7, 9. Then random forest takes first four 0,2,5,7 as input and try to predict the fifth one. The idea is to predict, if a person visited cluster 0,2,5,7 then in which cluster that person will be interested to visit. In other word, we can try to discover a person based on places he already visited.

In order to do that, we sorted the cluster sequence by writing a script on MongoDB. For instance we have transformed a sequence such as 2, 9, 0, 1, 0, 10 to 0, 1, 2, 9, 10. After preparing sorted sequence of cluster, first we conducted experiment with four inputs and one output, and later we try to predict with five inputs and one output. After sorting cluster sequence, there were **199** trajectories with cluster sequence five and **104** trajectories with cluster sequence six in our dataset.

On the other hand, before sorting there were **230** trajectories having cluster sequence of length 5 and **127** trajectories having cluster sequence of length 6. Regarding both cases of sorted data, we had 20 test data and the rest of data were used as training dataset. But in the case of ordered cluster sequence of length 5, among 230 data, 30 data were test data and rest 200 were used as training data. Since there were 127 trajectories having cluster sequence of length 6, thus we choose 20 of them as test data and the rest were used as training data. We can see the result of our experiment in Table 6.5.

	Sorted	Ordered
4 input and 1 output	40%	20%
5 input and 1 output	65%	20.56%

TABLE 6.5: Prediction estimates of sorted and ordered clustered sequences for non-filtered dataset. The length of the cluster sequence is 5 and 6 respectively.

In Table 6.5, we can notice that average prediction rate is comparatively larger for sorted cluster sequence.

6.2 Prediction with Filtered Data

Similar kind of preprocessing has been done for filtered data. We tracked each user photos by a time-stamped trajectory, we executed DBSCAN algorithm again for filtered data, but this time we write points inside each cluster with corresponding cluster id. Finally, we obtain cluster sequence for each trajectory of each user. Figure 6.2 depicts the distribution of length of cluster sequence against number of trajectories.

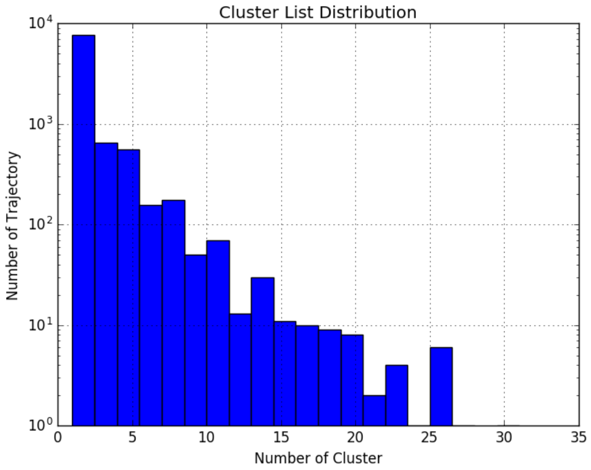


FIGURE 6.2: Distribution of length of cluster sequence against number of trajectory for filtered data.

From the Figure 6.2 we can notice that a large number of trajectories contain cluster sequence of size between 1-2, whereas more than 500 trajectories have cluster sequence of size between 3-4. Finally, we can observe that more than several hundreds of trajectories have cluster sequence greater than or equal five clusters. We counted from filtered dataset that there were 770 trajectories containing more than or equal to 5 cluster.

Likewise the non-filtered data, we experimented with both ordered and sorted cluster sequences of length five and six. It is worth mentioning, for sorted cluster list there were **511** data with cluster sequence of length 5, whereas there were **550** data with cluster sequence of length 6.On the other side, regarding ordered cluster list there were **770** data with cluster sequence of length 5, whereas there were **549** data with cluster sequence of length 6. Since we obtained 40 clusters for filtered dataset, this is why there are more trajectories in comparison of non-filtered dataset. Regarding all cases of filtered dataset, 50 data were picked as test dataset and the rest of them were picked as training dataset.

	Sorted	Ordered
4 input and 1 output	23.64%	33.33%
5 input and 1 output	31.16%	31.86%

TABLE 6.6: Prediction estimates of sorted and natural ordered clustered sequences for filtered dataset. The length of the cluster sequence is 5 and 6 respectively.

A close observation on Table 6.6 reveals, there is no significant difference among prediction rate for both sorted and ordered cluster sequence.

6.3 Discussion of the Results

We have studied the prediction of human mobility pattern by considering the aspect of applying sorting among the clusters along with natural order of the cluster sequences. In the case of non-filtered dataset, we have learned that prediction with sorted cluster sequence is much better than natural ordered cluster sequence. Although the context is very different for ordered and sorted cluster sequence, but both of them can be useful in two different contexts.

Regarding to the prediction of next movement from one cluster to another, random forest classification with natural sequence of clusters can be useful. On the other side, when the point of interest is to find a possible position of a person by analyzing his previous movements of that individual, then random forest classification with sorted sequence of clusters can be useful. We have analyzed our prediction rate with random prediction. Let us consider the case of non-filtered data where we obtained 21 clusters. If a system wants to predict randomly the next movement of an individual, then the probability will be 5% our prediction gives 25%. In addition, if we consider the sorted clustered sequence of length 6, the system with random technique will have the probability of 6% in comparison of 65% in our prediction.

Chapter 7

Conclusion and Future Work

We have reviewed relevant papers that cover specific aspects in Section 3. There we have seen about general overview of dataset we used. Also, we have seen the origin of data. We have studied previous approach of working with tourist based data analysis using data from other online data sources. Then, we have reviewed similar works like we have done that used the same data Yahoo100M to study mobility.

In this thesis, we have studied the elementary properties of the Yahoo100M data. Through pre-processing, we have extracted important attributes of the data and then we have shown basic characteristics of data. In Section 4, we have demonstrated general statistics of the data. Finding statistics about data helps to take decision for further analysis to select data subset. In this case, statistics of our extracted data gave us more understanding about category of users, number of possible movements and so on.

However, there are several difficulties to work with big dataset. It is not easy when you are going to work with file which is more than 1 GB, has millions of records. It is not same such as conducting small analysis. To understand the data, we have faced memory issues along with computation time to do all the pre-processing and filtering. Usually there is lots of learning in-side the data.

In Section 5, we have applied clustering technique to identify hot spots in Barcelona where people take a lot of pictures. In this case, we have chosen DBSCAN algorithm which works efficiently to build cluster with geospatial data. This clustering resulted groups of points that happened in the same places and we have put identification on those points. As a result, we have found most popular places in Barcelona according to the number of photos taken on these places. We have made exhaustive experiment to obtain appropriate result. For instance, we have tried to obtain clusters by continuously experimenting with different values of parameters.

From this section we have learned that clustering is a very important technique to group number of points together by matching some criteria of the points. It is always important to choose appropriate clustering mechanism along with proper parameter. One needs to decide according to the problem domain too. We need to study the spectrum of the parameters, and justify should we choose one or the other. Choices of good algorithm with good set of parameters enhance the study towards effective results. In our case, we choose to maximize the number of points and minimize the number of clusters. Parameters are the criteria we use to select to move towards to the solution of our problem.

On first part of Section 6, we have captured footprints of all users of Barcelona through aggregating their pictures in time stamped trajectories. Later, time stamped trajectories has taken us to the next step of analysis, in which we have obtained data that has helped us to track people's tendency of movement from one place to another. We have tested a mobility prediction method using data analytics and machine learning. In this we have chosen random forest which is one of the best approaches for doing prediction through classifying data. We have investigated the efficiency of machine learning algorithm to predict the next movement of individual as well as aggregated population.

Machine learning can be a good option for dealing with prediction problem. When it is not possible to define a function to obtain properties of some data then machine learning is the possible recommendation by several numbers of researchers as we observed in Section 3. In addition, we need to focus on the relativity of the obtained results. To put it differently, obtaining certain value of prediction, the number itself does not give you a lot. If you obtain prediction about 99% of something it is good but sometime you get some number that you need to put in the context. Therefore, is it good or is it bad depends on the particular context. In this thesis, we just wanted to experiment something random. In this case, we are extremely better instead of a random prediction system.

Also we have shown in Section 6, right now these are the best way to deal the domain of human movement prediction.The result we have obtained is in initial stage.

This was a very long project where we can do a full PhD if we want, but we wanted to learn all the steps in this process. We can explore in all the stages of the project with more details. For instance, in the clustering we can go and we can validate the clusters against the position in Barcelona. We can try different types of clustering. We can do two or three things extra to make sure which the actual real cluster is. Regarding machine learning, we have tried only one algorithm with one set of parameter, but there are many algorithms there. We can try, explore and make comparison among them.

And finally, the long term goal in this context could be designing standardized recommendation system for users based on the prediction from what others users have done on the similar situation.

Appendix A

Experiment of DBSCAN Clustering

Experiment Non-Filtered Dataset:

	minpoints= 200	minpoints= 240	minpoints= 280
eps = 0.05	Cluster: 110 Avg Size: 1785 Pictures: 68% = 195846	Cluster: 95 Avg Size: 1984 Pictures: 65% = 187205	Cluster: 73 Avg Size: 2440 Pictures: 62% = 178565
eps = 0.04	Cluster: 116 Avg Size: 1533 Pictures: 62% = 178565	Cluster: 96 Avg Size: 1747 Pictures: 58% = 167045	Cluster: 85 Avg Size: 1897 Pictures: 56% = 161285
eps = 0.03	Cluster: 127 Avg Size: 1236 Pictures: 55% = 158404	Cluster: 109 Avg Size: 1363 Pictures: 52% = 149764	Cluster: 88 Avg Size: 1592 Pictures: 49% = 141124

TABLE A.1: Test Case 1

	minpoints= 200	minpoints= 240	minpoints= 280
eps = 0.10	Cluster: 67 Avg Size: 3604 Pictures: 84% = 241927	Cluster: 63 Avg Size: 3754 Pictures: 82% = 236167	Cluster: 57 Avg Size: 4048 Pictures: 80% = 230407
eps = 0.05	Cluster: 110 Avg Size: 1785 Pictures: 68% = 195846	Cluster: 95 Avg Size: 1984 Pictures: 65% = 187205	Cluster: 73 Avg Size: 2440 Pictures: 62% = 178565
eps = 0.01	Cluster: 136 Avg Size: 703 Pictures: 33% = 95042	Cluster: 104 Avg Size: 834 Pictures: 30% = 86402	Cluster: 83 Avg Size: 971 Pictures: 28% = 80642

TABLE A.2: Test Case 2

	minpoints= 300	minpoints= 325	minpoints=350
eps = 0.05	Cluster: 68 Avg Size: 2560 Pictures: 60% = 172805	Cluster: 65 Avg Size: 2631 Pictures: 59% = 169925	Cluster: 62 Avg Size: 2693 Pictures: 58% = 167045
eps = 0.04	Cluster: 76 Avg Size: 2069 Pictures: 55% = 158404	Cluster: 77 Avg Size: 2002 Pictures: 54% = 155524	Cluster: 74 Avg Size: 2036 Pictures: 52% = 149764
eps = 0.03	Cluster: 85 Avg Size: 1619 Pictures: 48% = 138244	Cluster: 78 Avg Size: 1712 Pictures: 46% = 132484	Cluster: 75 Avg Size: 1739 Pictures: 45% = 129604

TABLE A.3: Test Case 3

	minpoints= 370	minpoints= 390	minpoints= 410
eps = 0.05	Cluster: 52 Avg Size: 3120 Pictures: 56% = 161285	Cluster: 52 Avg Size: 3091 Pictures: 56% = 161285	Cluster: 50 Avg Size: 3161 Pictures: 55% = 158404
eps = 0.04	Cluster: 69 Avg Size: 2132 Pictures: 51% = 146884	Cluster: 63 Avg Size: 2293 Pictures: 50% = 144004	Cluster: 63 Avg Size: 2273 Pictures: 50% = 144004
eps = 0.03	Cluster: 70 Avg Size: 1804 Pictures: 44% = 126723	Cluster: 67 Avg Size: 1838 Pictures: 43% = 123843	Cluster: 62 Avg Size: 1930 Pictures: 42% = 120963

TABLE A.4: Test Case 4

	minpoints= 450	minpoints= 475	minpoints= 500
eps = 0.05	Cluster: 52 Avg Size: 2989 Pictures: 54% = 155524	Cluster: 51 Avg Size: 3019 Pictures: 53% = 152644	Cluster: 47 Avg Size: 3199 Pictures: 52% = 149764
eps = 0.04	Cluster: 57 Avg Size: 2408 Pictures: 48% = 138244	Cluster: 56 Avg Size: 2419 Pictures: 47% = 135364	Cluster: 52 Avg Size: 2542 Pictures: 46% = 132484
eps = 0.03	Cluster: 57 Avg Size: 2014 Pictures: 40% = 115203	Cluster: 55 Avg Size: 2058 Pictures: 39% = 112323	Cluster: 53 Avg Size: 2094 Pictures: 39% = 112323

TABLE A.5: Test Case 5

	minpoints= 600	minpoints= 650	minpoints= 700
eps = 0.05	Cluster: 45 Avg Size: 3163 Pictures: 49% = 141124	Cluster: 41 Avg Size: 3344 Pictures: 48% = 138244	Cluster: 41 Avg Size: 3255 Pictures: 46% = 132484
eps = 0.04	Cluster: 47 Avg Size: 2674 Pictures: 44% = 126723	Cluster: 45 Avg Size: 2695 Pictures: 42% = 120963	Cluster: 40 Avg Size: 2888 Pictures: 40% = 115203
eps = 0.03	Cluster: 43 Avg Size: 2344 Pictures: 35% = 100803	Cluster: 39 Avg Size: 2463 Pictures: 33% = 95042	Cluster: 35 Avg Size: 2609 Pictures: 32% = 92162

TABLE A.6: Test Case 6

	minpoints= 750	minpoints= 800	minpoints= 850
eps = 0.05	Cluster: 40 Avg Size: 3224 Pictures: 45% = 129604	Cluster: 38 Avg Size: 3303 Pictures: 44% = 126723	Cluster: 34 Avg Size: 3538 Pictures: 42% = 120963
eps = 0.04	Cluster: 36 Avg Size: 3042 Pictures: 38% = 109443	Cluster: 33 Avg Size: 3219 Pictures: 37% = 106563	Cluster: 28 Avg Size: 3599 Pictures: 35% = 100803
eps = 0.03	Cluster: 32 Avg Size: 2757 Pictures: 31% = 89282	Cluster: 32 Avg Size: 2715 Pictures: 30% = 86402	Cluster: 26 Avg Size: 3130 Pictures: 28% = 80642

TABLE A.7: Test Case 7

Experiment Filtered Dataset:

	minpoints= 20	minpoints= 30	minpoints= 50
eps = 0.1	Cluster: 70 Avg Size: 1015 Pictures: 90% = 70974	Cluster: 73 Avg Size: 940 Pictures: 87% = 68609	Cluster: 45 Avg Size: 1417 Pictures: 81% = 63877
eps = 0.05	Cluster: 139 Avg Size: 436 Pictures: 77% = 60722	Cluster: 96 Avg Size: 581 Pictures: 71% = 55991	Cluster: 54 Avg Size: 904 Pictures: 62% = 48893
eps = 0.01	Cluster: 123 Avg Size: 164 Pictures: 26% = 20503	Cluster: 59 Avg Size: 257 Pictures: 19% = 14983	Cluster: 29 Avg Size: 395 Pictures: 15% = 11829

TABLE A.8: Test Case 1

	minpoints= 20	minpoints= 30	minpoints= 40
eps = 0.05	Cluster: 139 Avg Size: 436 Pictures: 77% = 60722	Cluster: 96 Avg Size: 581 Pictures: 71% = 55991	Cluster: 65 Avg Size: 799 Pictures: 66% = 52048
eps = 0.04	Cluster: 146 Avg Size: 382 Pictures: 71% = 55991	Cluster: 93 Avg Size: 541 Pictures: 64% = 50471	Cluster: 75 Avg Size: 610 Pictures: 62% = 48893
eps = 0.03	Cluster: 169 Avg Size: 290 Pictures: 62% = 48893	Cluster: 97 Avg Size: 433 Pictures: 53% = 41796	Cluster: 80 Avg Size: 473 Pictures: 48% = 37853

TABLE A.9: Test Case 2

Bibliography

[1] D. A. Raichlen, B. M. Wood, A. D. Gordon, A. Z. Mabulla, F. W. Marlowe, and H. Pontzer, “Evidence of lévy walk foraging patterns in human hunter–gatherers”, *Proceedings of the National Academy of Sciences*, vol. 111, no. 2, pp. 728–733, 2014.

[2] M. Berlingerio, F. Calabrese, G. Di Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio, “Allaboard: A system for exploring urban mobility and optimizing public transport using cellphone data”, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2013, pp. 663–666.

[3] S. Chaturapruek, J. Breslau, D. Yazdi, T. Kolokolnikov, and S. G. McCalla, “Crime modeling with lévy flights”, *SIAM Journal on Applied Mathematics*, vol. 73, no. 4, pp. 1703–1720, 2013.

[4] L. Hufnagel, D. Brockmann, and T. Geisel, “Forecast and control of epidemics in a globalized world”, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 42, pp. 15 124–15 129, 2004.

[5] V Colizza, A Barrat, M Barthélemy, A. Valleron, and A Vespignani, “Modeling the world-wide spread of pandemic influenza: Baseline case and containment interventions supporting information”, 2006.

[6] B. Gonçalves, D. Balcan, and A. Vespignani, “Human mobility and the worldwide impact of intentional localized highly pathogenic virus release”, *Scientific reports*, vol. 3, p. 810, 2013.

[7] W. Y. Labs. (2017). MS Windows NT kernel description, [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67. (visited on 12/01/2016).

[8] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research”, *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[9] D. Barchiesi, T. Preis, S. Bishop, and H. S. Moat, “Modelling human mobility patterns using photographic data shared online”, *Royal Society open science*, vol. 2, no. 8, p. 150 046, 2015.

[10] A. Bassolas, M. Lenormand, A. Tugores, B. Gonçalves, and J. J. Ramasco, “Touristic site attractiveness seen through twitter”, *EPJ Data Science*, vol. 5, no. 1, p. 12, 2016.

[11] M. Jabreel, A. Moreno, and A. Huertas, “Semantic comparison of the emotional values communicated by destinations and tourists on social media”, *Journal of Destination Marketing & Management*, 2016.

[12] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns”, *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[13] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility", *IEEE/ACM transactions on networking (TON)*, vol. 19, no. 3, pp. 630–643, 2011.

[14] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns", *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[15] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo, "A tale of many cities: Universal patterns in human urban mobility", *PloS one*, vol. 7, no. 5, e37027, 2012.

[16] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti, "Geo-located twitter as proxy for global mobility patterns", *Cartography and Geographic Information Science*, vol. 41, no. 3, pp. 260–271, 2014.

[17] C. Roth, S. M. Kang, M. Batty, and M. Barthélemy, "Structure of urban movements: Polycentric activity and entangled hierarchical flows", *PloS one*, vol. 6, no. 1, e15923, 2011.

[18] M. Lenormand, A. Tugores, P. Colet, and J. J. Ramasco, "Tweets on the road", *PloS one*, vol. 9, no. 8, e105407, 2014.

[19] E. G. Ravenstein, "The laws of migration", *Journal of the Statistical Society of London*, vol. 48, no. 2, pp. 167–235, 1885.

[20] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns", *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[21] A. Wesolowski, N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, and C. O. Buckee, "Quantifying the impact of human mobility on malaria", *Science*, vol. 338, no. 6104, pp. 267–270, 2012.

[22] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale mobility networks and the spatial spreading of infectious diseases", *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 484–21 489, 2009.

[23] S. Merler and M. Ajelli, "The role of population heterogeneity and human mobility in the spread of pandemic influenza", *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 277, no. 1681, pp. 557–565, 2010.

[24] M. G. Beiró, A. Panisson, M. Tizzoni, and C. Cattuto, "Predicting human mobility through the assimilation of social media traces into mobility models", *EPJ Data Science*, vol. 5, no. 1, p. 30, 2016.

[25] W.-S. Jung, F. Wang, and H. E. Stanley, "Gravity model in the korean highway", *EPL (Europhysics Letters)*, vol. 81, no. 4, p. 48 005, 2008.

[26] T. Grosche, F. Rothlauf, and A. Heinzl, "Gravity models for airline passenger volume estimation", *Journal of Air Transport Management*, vol. 13, no. 4, pp. 175–183, 2007.

[27] Y. Liu, Z. Sui, C. Kang, and Y. Gao, "Uncovering patterns of interurban trip and spatial interaction from social media check-in data", *PloS one*, vol. 9, no. 1, e86026, 2014.

[28] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale mobility networks and the spatial spreading of infectious diseases", *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 484–21 489, 2009.

[29] G. Krings, F. Calabrese, C. Ratti, and V. D. Blondel, "Urban gravity: A model for inter-city telecommunication flows", *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 07, p. L07003, 2009.

[30] A. P. Masucci, J. Serras, A. Johansson, and M. Batty, "Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows", *Physical Review E*, vol. 88, no. 2, p. 022 812, 2013.

[31] X. Liang, J. Zhao, L. Dong, and K. Xu, "Unraveling the origin of exponential law in intra-urban human mobility", *ArXiv preprint arXiv:1305.6364*, 2013.

[32] Y. Yang, C. Herrera, N. Eagle, and M. C. González, "Limits of predictability in commuting flows in the absence of data for calibration", *ArXiv preprint arXiv:1407.6256*, 2014.

[33] J. Truscott and N. M. Ferguson, "Evaluating the adequacy of gravity models as a description of human mobility for epidemic modelling", *PLoS Comput Biol*, vol. 8, no. 10, e1002699, 2012.

[34] M. G. Demissie, F. Antunes, C. Bento, S. Phithakkitnukoon, and T. Sukhvibul, "Inferring origin-destination flows using mobile phone data: A case study of senegal", pp. 1–6, 2016.

[35] V. Palchykov, M. Mitrović, H.-H. Jo, J. Saramäki, and R. K. Pan, "Inferring human mobility using communication patterns", *ArXiv preprint arXiv:1404.7675*, 2014.

[36] L. Alexander, S. Jiang, M. Murga, and M. C. González, "Origin–destination trips by purpose and time of day inferred from mobile phone data", *Transportation research part c: Emerging technologies*, vol. 58, pp. 240–250, 2015.

[37] M. Lenormand, M. Picornell, O. G. Cantu-Ros, A. Tugores, T. Louail, R. Herranz, M. Barthelemy, E. Frias-Martinez, and J. J. Ramasco, "Cross-checking different sources of mobility information", *PLoS One*, vol. 9, no. 8, e105184, 2014.

[38] P. A. Grabowicz, J. J. Ramasco, B. Gonçalves, and V. M. Eguíluz, "Entangling mobility and interactions in social media", *PloS one*, vol. 9, no. 3, e92196, 2014.

[39] A. Llorente, M. Garcia-Herranz, M. Cebrian, and E. Moro, "Social media fingerprints of unemployment", *PloS one*, vol. 10, no. 5, e0128692, 2015.

[40] (2017). Dbscan: Density-based clustering for discovering clusters in large datasets with noise - unsupervised machine learning - easy guides - wiki - sthda, [Online]. Available: http://www.sthda.com/english/wiki/dbscan-density-based-clustering-for-discovering-clusters-in-large-datasets-with-noise-unsupervised-machine-learning#application-of-dbscan-on-a-real-data (visited on 12/11/2016).

[41] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.”, vol. 96, no. 34, pp. 226–231, 1996.

[42] (2017). Visualizing dbscan clustering, [Online]. Available: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/ (visited on 12/01/2016).

[43] (2017). A complete tutorial on tree based modeling from scratch (in r and python), [Online]. Available: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/ (visited on 12/12/2016).

[44] J. Brownlee. (2017). Bagging and random forest ensemble algorithms for machine learning - machine learning mastery, [Online]. Available: http://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/ (visited on 12/01/2016).

[45] C. L. Borgman, “The conundrum of sharing research data”, *Journal of the American Society for Information Science and Technology*, vol. 63, no. 6, pp. 1059–1078, 2012.

[46] J. Choi, B. Thomee, G. Friedland, L. Cao, K. Ni, D. Borth, B. Elizalde, L. Gottlieb, C. Carrano, R. Pearce, *et al.*, “The placing task: A large-scale geo-estimation challenge for social-media videos and images”, pp. 27–31, 2014.

[47] (2017). Cartodb, [Online]. Available: https://en.wikipedia.org/wiki/CartoDB (visited on 12/01/2016).

[48] (2017). What is characterization? - definition from techopedia, [Online]. Available: https://www.techopedia.com/definition/30312/characterization-data-analysis (visited on 12/12/2016).