Project Report

Large Scale Customer Location Inference

Professor: Pietro MICHIARDI

Student: Amarnath CHIGURUPATI

Table of contents

Acknowledgment

Though my name appears on the cover of this report, a number of great people have contributed to its production. I owe my gratitude to all those people who have made this possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor / supervisor, Professor. Pietro Michiardi. I have been amazingly fortunate to have an advisor who gave me the freedom to explore and at the same time, the guidance to recover when my steps faltered. I would like to extend my regards to Trung Nguyen who taught me how to question thoughts, express ideas and solve problems. These people's patience and support helped me overcome many crisis situations and bring up to the finish line. I hope that one day I would become as good as the people I talked about.

Introduction

The context of this project lies in mobile network service provisioning, and it is related to the quest for better customer experience when offering mobile network services to customers.

In the mobile network that we consider in this work, some critical components are– labelled enforcement points (EP) – execute traffic engineering tasks to decide, for example, given the current state of the mobile network for a particular user, which media encoding to use to deliver data to a mobile terminal. Today, EPs may behave similarly to an admission control mechanism and they accept all traffic or none.

The endeavour of this work is to inject additional information in EPs, such that better decisions and not only all-or-nothing) could be taken. Specifically, we consider radio access network (RAN) conditions, which include: congestion in the RAN, when EP should implement their decisions, knowledge about the serving cell (SC) for each customer.

Now, the knowledge about the serving cell for each user is hard to obtain in real time, which may limit the usefulness of such information to EP. In particular, due to the current state of the technology used to operate the mobile network, it is possible to measure the serving cell for a user only with a non-negligible delay (in the order of tens of minutes, e.g. 15 minutes). Given such limitation, the customer location inference problem can be tackled with simple heuristics, such as bounding the region surrounding critical EPs, where the bounding box should be sufficiently loose to allow recovering the delay in obtaining the SC information. These heuristic may work in practice, but their merit still need to be assessed. Moreover, the heuristic approach is non-scalable, and can only be applied to a selected subset of "hot" EPs.

As such, in this work we tackle the problem of providing estimates of customer location, in the sense of trying to estimate (or predict), at a given point in time, which users will be served by which SC. Based on delayed (and possibly stale) measurement data, combined with historical data of customer location, we want to produce accurate estimates of customers' SCs.

Among the two basic system architectures to build the model: Batch processing and Real time data processing system [5]. Currently, in our work, we only focus on batch processing as we are not working with the real time data.

With the batch processing architecture and by using machine learning algorithms, historical data is used to train a model, which will predict the next location for users. Also here, we should consider the scope of models. In literature, there are two kinds of models: Individual models and Global models. Individual models are the trained models for each user, and then used to predict the next location of

this user alone. Whereas global models are built and used to make predictions for all users. The former is more widely used. However, in this project, we will build and evaluate both model scopes.

In this project, we use tree-based approach to train models. As mentioned earlier, we use Decision Trees and Random Forest to predict the target feature: Next location. Decision Tree will be used as the baseline to compare the performance with other algorithms like Random Forest. The models are built using the mllib - machine learning library in SPARK - a very powerful scalable framework.

The rest of this document is organized as follows: Chapter 1 is related works and some background knowledge will be used in this project. In chapter 2, we introduce and describe in detail the data we use to build our statistical models, and to validate our algorithms. Chapter 3 we talk about tree-based approach and Finally, we present our experiment results and conclusion.

Chapter 1

Background and related works

1.1 Related works

Our goal in this project is to build models using mllib in SPARK on the Geo-life dataset from Microsoft to predict the next location of users. So, our focus is on studies of location inference problems.

1.2 State of the art in location inherence

We studied number of related work that addresses the similar problem we focus on this project. Most of the research works that we reviewed are efforts of machine learning challenge in 2012 from NOKIA and ORANGE. However, the original datasets used in the research papers are no longer available.

V. Etter et. al. [6] predicted the next destination of a user given the current context, by building user-specific models that learn from their mobility history, and then applying these models to the current context to predict where the users go next. The data was collected from the mobile phone of the user (date, location of the user, cell tower id, WLAN, phone calls, etc.). The features used for building the model are as follows : Location, Start time of visiting (Hour, Day, Weekday), End time of visiting (Hour, Day, Weekday). They addressed the problem based on graphical models, neural networks, decision trees and some blending strategies. In there, they listed three characteristics of data they think are critical to the prediction task which can significantly improve the prediction accuracy:

- **User Specificity:** It is not possible to build joint models over the user population to learn from someone to make prediction for another.
- **Non-Stationarity :** User can change his/her habit overtime (when moving house or office)
- **Data Gaps:** For some user, there is no information about them in long periods. And these gaps are sometimes followed by change of mobility habit.

Additionally, to overcome non-stationarity, they developed Aging techniques in the learning process based on the observation that, when a user changes his habits, recent history is more representative of his/her future behaviour than the accumulated information. Thus aging technique helps reduce the contribution of old samples. To solve the problem and to build the predictors, they used Dynamical Bayesian Network, Artificial Neural Networks and Gradient Boosted Decision Trees to build three different models for each user, compare them. Due to the diversity of predictors, the accuracies of the predictors are not equal, but more importantly, each predictor makes different errors: samples for

which a predictor fails might be those on which another excels, which led then to use the blending strategy, in which they combined several predictors, in order to take advantage of their diversity. The average of accuracy is more than 56%.

In 2012, J. Wang et. al. [7] wants to predict the next location of a user based only on his trajectory. They assume that user behaviour exhibits strong periodic patterns. The model for each user will be built based on Periodicity Based Model and Multi-class Classification algorithms. The features used here are extracted from: start time of a visit, end time of a visit and current location. The features which we are interested in contain day of week, hour of day, hour of week, weekend, weekday, morning, noon, afternoon, evening, midnight. The accuracy of this approach when applying on Nokia challenge's data is around 55%. This approach doesn't consider the relationship between the next location and the previous locations. There are some downsides in the context of multi-class classification problem. Where, for each entry, extract the features from the current context and use the next place as the label. However, the class labels are highly imbalanced because of the dominant places. Meanwhile, many of the minority classes have very few data samples. This will cause the classifier to favour the majority classes. And since there are not enough samples for the minority classes, the accuracy on the minority classes will be very low.

In [8] A. Noulas et. al. Addressed the problem of predicting the next venue a mobile user will visit, by exploring the predictive power offered by different facets of user behaviour. Their analysis is based on the check-in extracted from the Foursquare. They formalized the Next Check-in Problem, where the aim is to predict the exact place a user will visit next given historical data and the current location. Essentially, the next check-in problem is a ranking task, where they computed a ranking score r for all venues that the user might visit. They considered the features that exploit different information dimensions about users' movements: those include historical visits or social ties, and features extracted by mining global knowledge about the system such the popularity of places, their geographic distance and user transitions between them. Also, they leverage the features that provide temporal information about users' movements like Hour and day of visiting the place. Next, they combine the predictive power of individual features in a supervised learning framework. By training two supervised regressor's, a regularized linear model and M5 model trees, on past user movements, the latter achieves noticeably higher performance with an accuracy of 60%.

Chapter 2

Data

2.1 Introduction

After reviewing the past work which tried to solve the similar problem like the one we are addressing, it is found that, most of the papers were using datasets from Nokia and Orange which are not available anymore. However, we learnt about the GPS dataset available, which can be used for location inference problems. This is introduced below. But the dataset in its native form doesn't support or cannot be used to predict the next location of the customer (in our case) as it does not contain any servicing cell information (user connected to) rather it has time stamped geo location coordinates trajectories of users. Thus our goal is to place or consider a fake servicing cell or basestation over the area covered by the dataset and then assign each sample with a basestation id based on its geo coordinates, which is presented in the data transformation section below.

2.2 Original Dataset

2.2.1 Introduction

This is a portion of GPS trajectory dataset collected in (Microsoft Research Asia) GeoLife project. A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude, height, speed and heading direction, etc. These trajectories were recorded by different GPS loggers or GPS-phones, and have a variety of sampling rates. 95 percent of the trajectories are logged in a dense representation, e.g., every 2~5 seconds or every 5~10 meters per point, while a few of them do not have such a high density being constrained by the devices. Note that each trajectory has a set of transportation mode labels (file), which is irrelevant for this project.

Although this dataset is wildly distributed in over 30 cities of China and even in some cities located in the USA and Europe, the majority of the data was created in Beijing, China.

2.2.2 Data Format - Trajectory file

Every single folder of this dataset stores a user's GPS log files, which were converted to PLT format. Each PLT file contains a single trajectory and is named by its starting time.

Plt format:

Line 1…6 are useless in this dataset, and can be ignored. Points are described in following Table 2.1, one for each line.

Field	Column Name	Description
1	**Latitude**	Latitude in decimal degrees.
2	**Longitude**	Longitude in decimal degrees.
3	**Field 3**	All set to 0 for this dataset.
4	**Altitude**	Altitude in feet (-777 if not valid).
5	**Date - Number of days**	Number of days (with fractional part) that have passed since 12/30/1899
6	**Date**	Date as a string.
7	**Time**	Time as a string.

Table 2.1: Data structure and format of the geo life dataset for Microsoft

Note that field 5 and field 6&7 represent the same date/time in this dataset. You may use either of them.

Example:

39.906631,116.385564,0,492,40097.5864583333,2009-10-11,14:04:30

39.906554,116.385625,0,492,40097.5865162037,2009-10-11,14:04:35

2.3 Data pre-processing

2.3.1 Introduction

In order to facilitate pouring data to different processing frameworks, algorithms...and mainly to reuse the existing (previous) work on this project, we need to transform the data in .plt format (with latitude & longitude) into the text format having basestations assigned to each sample with the following schema: userID, year, month, day of month, day of week, hour of day, minute, quarter, basestation id.

The idea is to use the Geolife dataset with latitude & longitude and transform these values into the basestation ids.

2.3.2 Facts and Assumptions

We consider the following facts and assumptions based on some the information inferred from the dataset which are presented alongside.

2.3.2.1 Facts

From the maximum and minimum values of latitude and longitude extracted from the dataset, the bounding box is found to be 3000 Km in length with an area of $2.2601\times10^{+08}$ km^2.

Follow link [4] to see the area that is covered by the bounding box.

2.3.2.1 Assumptions

Since this dataset is widely distributed in cities of China and even in some cities located in the USA and Europe, the area under consideration seems to be huge ($2.2601\times10^{+08}$ km^2). And it is practically not feasible to consider a base stations spread across the entire area. So we consider dividing the area into number of regions each with an area equivalent to the area of Beijing (where the majority of the data is created), intern each region is divided into sub regions of feasible size to place the base station.

Hence, we have divided the entire area ($2.2601\times10^{+08}$ km2) into 13452 possible regions with an area of 16801km^2 each, which is equivalent to the area of the Beijing (where the most of the samples belong / created). In theory, it is same as dividing the area in to a grid (matrix) with 116×116 Cells.

Also, we have placed or considered 400 possible base stations spread in each region with one basestation per 42km^2. In theory, it is as good as dividing each region into the grid having 20×20 Cells each with 42km^2 by area. Moreover, a cell with 42km^2 is sufficient to have single base station place at the centre to have coverage up to 5 km.

2.3.3 Data Transformation: Standard schema

We have segregated users samples into different regions based on their geo coordinates and then assigned a basestation id for each sample from the possible 400 base stations available in the region where the user was segregated earlier. Thus, we have transformed, filtered and saved the dataset with geo coordinates to the dataset with base stations ids for each time stamped sample. The size of the final dataset is 700.5 MB with more than 2.3 million records (or observations) with the schema below: as explained in the Table 2.2.

userID, year, month, day of month, day of week, hour of day, minute, quarter, basestation id.

Name	Description
userID	Number - which uniquely identifies the User
Year	Integer number - Representing the year during which the sample was collected
Month	Integer number - Representing the month of the year during which the sample was collected
Day Of Month	Integer number - Representing the date of the month of the year during which the sample was collected
Day Of Week	Integer number - Representing the week during which the sample was collected. Possible values from 1-7 (7 days in a week).
Hour Of Day	Integer number - Representing the hour in 24hr format. (0-23hrs)
Minute	Integer number - Representing the minute (0-59 mins)
Quarter	Integer number - Representing the quarter of the year. 1-4 for each quarter in a year.
Basestation Id	Integer number - which uniquely identifies the Base station which the user is connected to.

Table 2.2 : Data structure and data format of the transformed dataset.

For instance:

153,2012,5,11,4,13,4,2,8

128,2009,2,22,6,15,22,1,24

128,2009,2,22,6,15,25,1,260

The column basestation id is the name given to the base station placed / considered inside the regions.

2.3.4 Data Properties

In order to visualize our idea of segregating users into different regions and assigning each user records with basestation id based on the geo-coordinates. we now proceed with an exploratory data analysis and plot cumulative distribution function (CDFs) to find the distribution of users to base stations and vice-versa. As an illustrative example, Figure 2.1a and Figure 2.1b shows the distribution of base stations to the users and the distribution of users to the base stations respectively, over the entire area (considering all the regions). Specifically: Figure 2.1a: x-axis is the number of users; y-axis is the probability of base stations with users and Figure 2.1b: x-axis is the number of base stations; y-axis is the probability of users connected to the base stations.

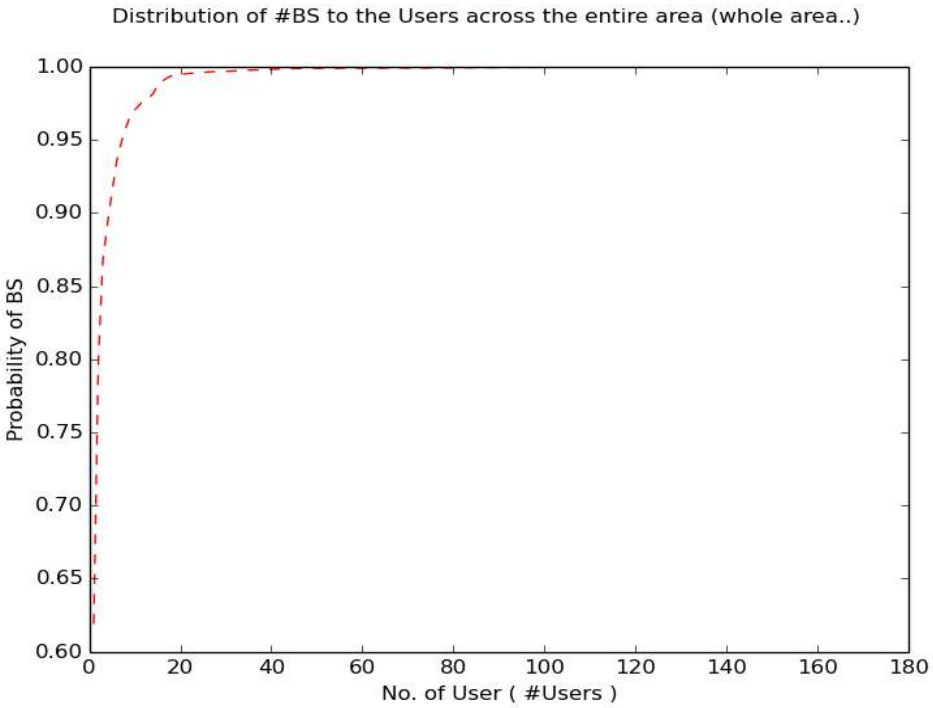

Figure 2.1a: Distribution of base stations to the users.

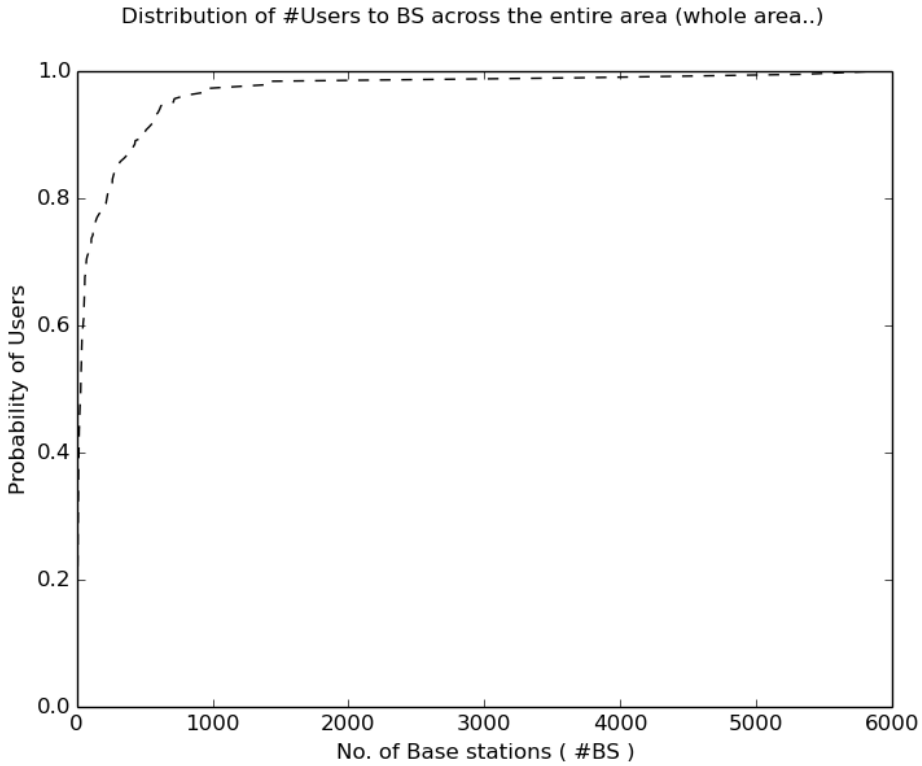

Figure 2.1b: Distribution of users to the base stations.

We can draw the following interpretations, from the Figure 2.1a: 63% of the base stations have at least 1 user; 98% of the base stations have 20 or less number of users. From the Figure 2.1b: 10% of the users have 20 or less number of base stations; 98% of the users have 3000 or less number of Base stations. And so on...

Having seen the distributions of users and base stations across the entire area, the following cdfs depicts the distributions in the individual regions. As an example, Figure 2.2(a & b) and Figure 2.3(a & b) corresponds to the regions 11090 (with 179 users) and 11083 (with 16 users) respectively. It is evident from the Figure 2.2a that this region as samples from 179 users who are connected to different base stations at different times. Thus this region could possibly be the Beijing where most of the data is created. And also, In the Figure 2.3 shows that this particular region as fewer number of users, thus, it could be another city / place other than Beijing where fewer samples were created.

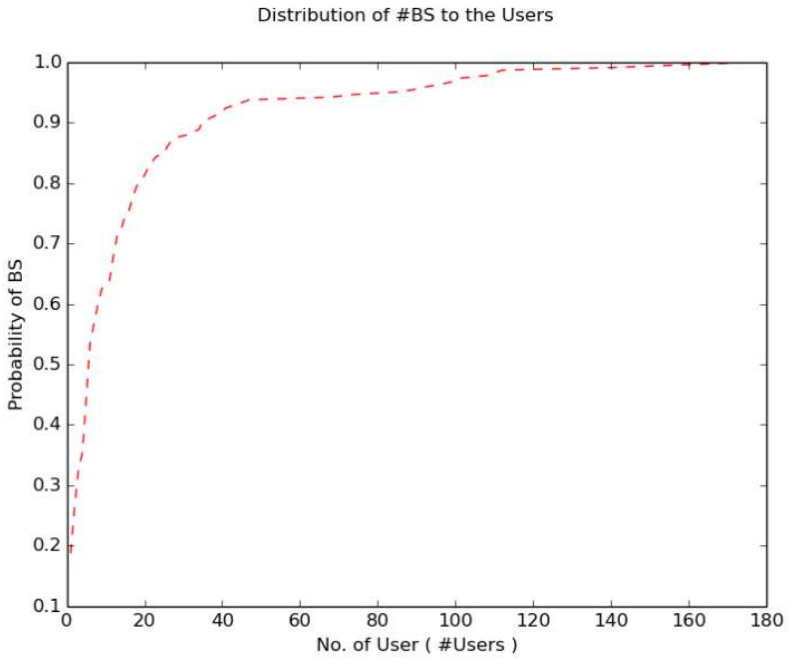

Figure 2.2a: 11090: Distribution of base stations to the users.

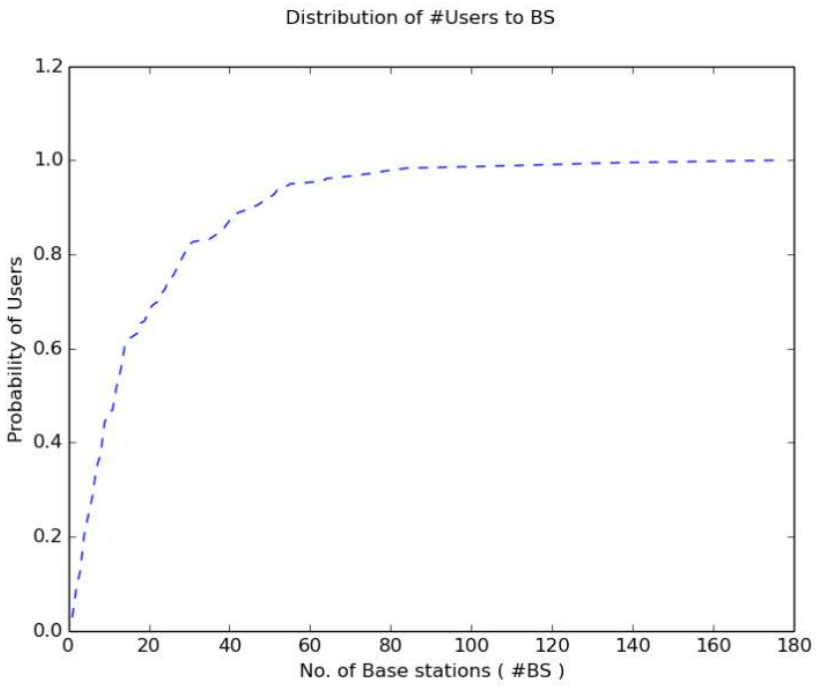

Figure 2.2b: 11090: Distribution of users to the base stations.

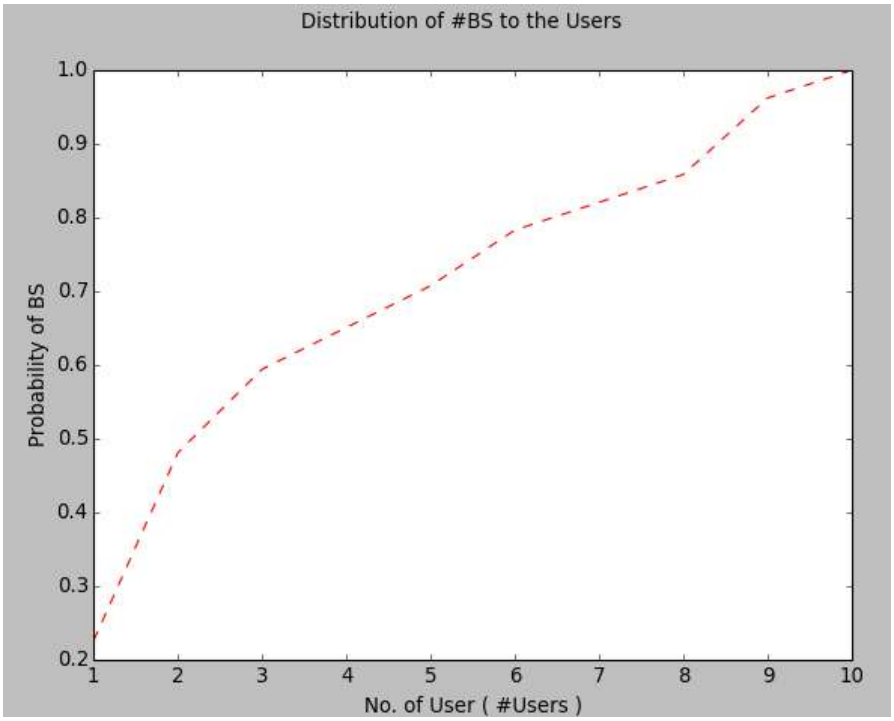

Figure 2.3a: 11083: Distribution of base stations to the users.

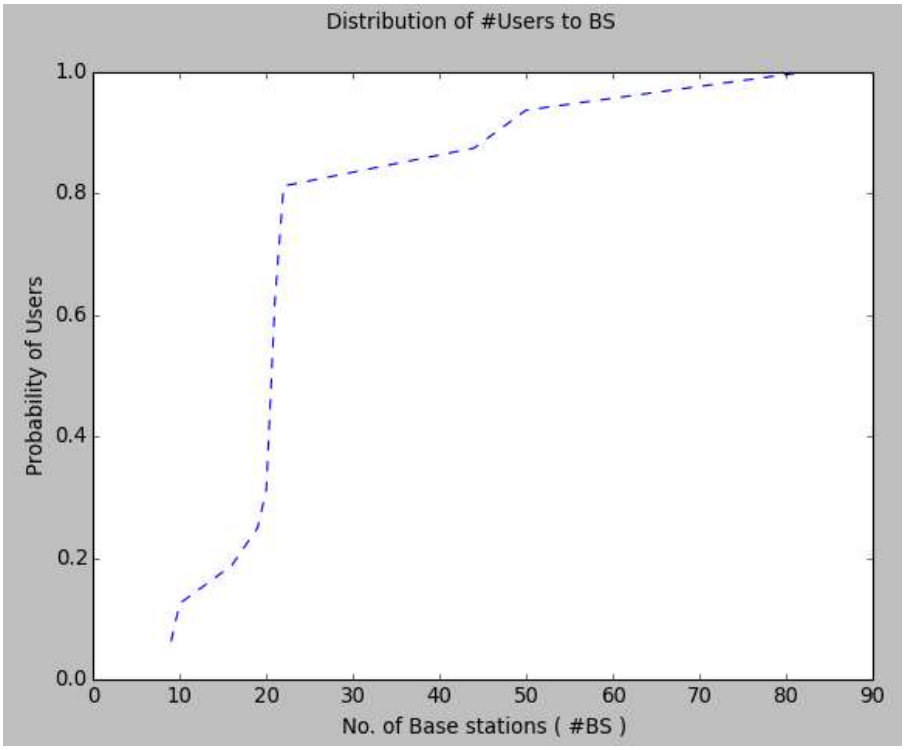

Figure 2.3b: 11083: Distribution of users to the base stations.

Chapter 3

Approach: Decision Trees

3.1 Introduction

Decision trees are a simple, but powerful form of multiple variable analysis. Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions. With many advantages such as inexpensive to construct models, extremely fast at classifying unseen records, the accuracy is comparable to other classification techniques and decision tree is often used to address the location inference problem.

In this chapter, we use SPARK's decision tree module in MLLIB to build models and introduce some techniques to create an implicit variables/features in order to increase the performance accuracies of the models.

The main idea of our approach is that, we consider the next location (base station) of the user as the target feature and the other information as predictors / features used in the prediction task using tree based learning algorithms. We consider the problem as multiclass classification problem and we treat the next location (target feature) as discrete value. In our opinion, it is hard to see the problem as regression. For the reason that, we try to predict the next location of the user as one of the places he/she had been to before but shouldn't be a new location all together. Also, the predicted value of the leaf node in regression trees is the average over the samples that belong to the leaf node which might lead to a new non existing base station.

Therefore, with this approach, we only use classification tree to build models to forecast the basestation id which the users will connect to.

3.2 Strategies in building models

Considering our main idea of segregating user samples in the whole area into different regions (485 actual regions created in total). Thus we propose the following strategies for Global and Individual models:

3.2.1 Global models:

Strategy 1: From the area point of view - One model for the whole area. Same model has been used by all the regions in the area i.e., the same model will be used by all the users in the area. Thus, there will be 1 model.

Strategy 2: From the region point of view - One model per region i.e., the same model will be used by all the users in the particular region, in other words, separate model for same users (say user 90) in different regions. Thus, there will be 'R' number of models. Where R is the total number of regions.

3.2.2 Individual models:

Strategy 3: From the area point of view - One model for each user in the whole area. Thus, there will be 'N' number of models. Where N is the total number of users in the whole area.

Strategy 4: From the region point of view - One model per user in each region. Thus, there will be $\mathbf{N}^{'}$ number of models per region. Where $\mathbf{N}^{'}$ is the number of users in a particular region.

3.3 Feature selection

We consider feature/variable selection as an important aspect in order to increase the accuracy of the models. As an input, we want to use all the provided information from our standard schema discussed in section 2.2.3 to make our predictors more accurate, but we could not extract meaningful patterns as such. Thus in order to increase the prediction accuracies, we consider generating some implicit predictors, by using both temporal and spatial information of the users along with the help of time-series knowledge to prediction the next locations of a user at a given time in near future. The following section 3.4 introduces our idea of creating the implicit predictors using the time series mechanism.

3.4 Implicit predictors: using time series approach

A time series data is a sequence of data points, typically consisting of successive measurements made over a time interval. In order to generate the implicit predictor using time series approach, we need to transform the data into the suitable schema.

As mentioned in section 2.2.3, the standard dataset as the following schema:

userID, year, month, dayOfMonth, dayOfWeek, hourOfDay, minute, quarter, basestationID.

We call the data in above schema as *'location information'* i.e., for a given user, at a given time, he/she is connected to the base station Y. Because time is a continuous variable, we can discrete time to the smaller time intervals.

Therefore our goal is, for each user, transform the location information into the time intervals in a given discrete time granularity. We select such granularity to be a daylong of data. The granularity factor M in minutes divides a day into discrete time intervals. As a result, the new data samples have the following schema and so on.

$user_1$, $dayOfMonth_1$, $currentTimeInterval_1$, $currentLocation_1$, $nextTimeInterval_2$, $nextLocation_2$

$user_1$, $dayOfMonth_1$, $currentTimeInterval_2$, $currentLocation_2$, $nextTimeInterval_3$, $nextLocation_3$

.
.
.
.
.

$user_N$, $dayOfMonth_N$, $currentTimeInterval_{N-1}$, $currentLocation_{N-1}$, $nextTimeInterval_N$, $nextLocation_N$

Which can be read as: a particular user '$user_1$' on a particular day ' $dayOfMonth_1$' was at location '$currentTimeInterval_1$' during the time interval '$currentLocation_1$' and the user will be at '$nextLocation_2$' during the next time interval ' $nextTimeInterval_2$'.

Where *'*$nextTimeInterval_2$', '$nextLocation_2$' are same as '$currentTimeInterval_2$', '$currentLocation_2$' respectively for the two consecutive samples/observations, if they belong to the same day for a particular user.

However, it is challenging if the user has changed and (or) connected to many base station during the particular time interval. For example, during the time interval from 14:00 hrs to 15:00 hrs, the user connect to basestation X for 15 mins, then to basestation Y for 35 minutes, and then reconnect to basestation X for the rest of time. So what is the basestation that we consider for this particular time interval? One solution is choosing the basestation (Y in this case) to which the user has connected most of time during the time interval. It means, we choose the basestation that has the highest probability that the user will connect to during this time interval. Because we select only one basestation to make data, we will ignore some other information (other base stations) which can affect the accuracy. If the granularity factor 'M' is small enough, this problem can disappear. But how can we choose value for M? It depends on the rate at which the user's movements were captured. Since the data recorded rate is 5 to 10 sec, the user moments are dense, thus we use the minimum value for M i.e., M = 1 minute.

In summary, the implicit predictors generated are (*currentTimeInterval, currentLocation*) and *(nextTimeInterval, nextLocation).* Where, in a single observation/record the *(currentTimeInterval, currentLocation)* and *(nextTimeInterval, nextLocation)* are two pairs of time/location information in two time intervals not necessarily adjacent. The size of the data after transformation is 29 MB.

3.5 Experiment Evaluation

In this section, we used decision trees and random forest to predict the next location of mobile user from the data collected as part of Microsoft Geolife dataset.

3.5.1 Set up

Since the data size after pre-processing is quite small around 700MB and 29MB for standard schema and time series schema respectively, we run it on the personal computer. The specifications of computer are as follows: Ubuntu 14 , Intel Core i5 2.3 GHz, 6GB RAM, 150 GB HD.

3.5.2 Data

In the experiments, we have used Microsoft Geolife dataset which we introduced in chapter 2. This data is in schema (standard):

userID, year, month, dayOfMonth, dayOfWeek, hourOfDay, minute, quarter, basestationID

In order to increase the accuracy of our models, we have considered some implicit features. Since time is a continuous variable, we tried to discrete it into L equal time intervals with length M (in minutes). Thus we have created the data samples in the new schema (shown below) as mentioned in section 3.3 & 3.4

$user_1$, $dayOfMonth_1$, $currentTimeInterval_1$, $currentLocation_1$, $nextTimeInterval_2$, $nextLocation_2$

We have selected first 80% of the samples of each user in each region to form up the training data. The remaining 20% of the data has been used as testing data. After transforming, section 3.4, we have 965500 observations as training data and 241741 samples as testing data.

As mentioned before, we have evaluated with two model scopes: Global model and Individual model. Using classification trees, we try to predict the next location of the user. We have performed experiments by considering 3 different set of predictors to evaluate the performances for the strategies discussed in section 3.2. which are discussed below.

Note: '*nextLocation' and 'basestationID' are used interchangeably.*

Predictor set-1: Using data in the standard schema:

userID, year, month, dayOfMonth, dayOfWeek, hourOfDay, minute, quarter, basestationID

Model has been built using predictors as shown below:

basestationID ← *month* + *dayOfMonth* + *dayOfWeek* + *hourOfDay* + *minute* + *quarter*

The performance of the models are worst i.e., 30% accuracy.

Predictor set-2: Using data with time series schema as shown below:

User, dayOfMonth, currentTimeInterval, currentLocation, nextTimeInterval, nextLocation

Model has been built using predictors as shown below:

nextLocation ← *currentTimeInterval* + *currentLocation* + *nextTimeInterval*

The following experiments have be performed using the above model (Predictor set-2), we have used decision trees and Random Forest implementation in SPARK's mllib library to test the performances, with different model scopes:

- From region point of view:

- **Global model with DT:** Build one model per region and use this tree to make predictions for every user in that region.
- **Global model with Random Forest:** Build one model per region and use this forest to make predictions for every user in that region.

- From area point of view:

- **Global model with DT**: Build one model for the whole area (for all regions). And use the tree built from this model to make predicts for everyone.

Predictor set-3: Using data with time series schema as shown below:

user, dayOfMonth, currentTimeInterval, currentLocation, nextTimeInterval, nextLocation

Model has been built using predictors as shown below:

nextLocation ← *dayOfMonth* + *currentLocation* + *nextTimeInterval*

dayOfMonth is used as a categorical feature.

Following experiments have been performed using the above model (Predictor set-3), we have used decision trees and Random Forest implementation in SPARK's mllib library to test the performances, with different approaches and parameters:

- From region point of view:

- **Global model with DT:** Build one model per region and use this tree to make predictions for every user in that region.
- **Global model with Random Forest:** Build one model per region and use this forest to make predictions for every user in that region.
- **Individual model with DT:** Build one model per user in each region and use this model only to make predictions to that particular user in that region only.

- From area point of view:

- **Global model with DT**: Build one model for the whole area (for all regions). And use the tree built from this model to make predicts for everyone.
- **Individual model with DT:** Build one model per user in the whole area irrespective of the regions and use this to make predictions to that user alone.

Accuracies are plotted in the following section 3.5.3

3.5.3 Results

Experiment results for **Predictor set-2:**

Although the accuracies are not optimal, the global model from the area point of view is shown to have better accuracy with 69%. The Figure 3.1a and Figure 3.1b below shows the accuracy of different approaches for area and region point of view respectively.

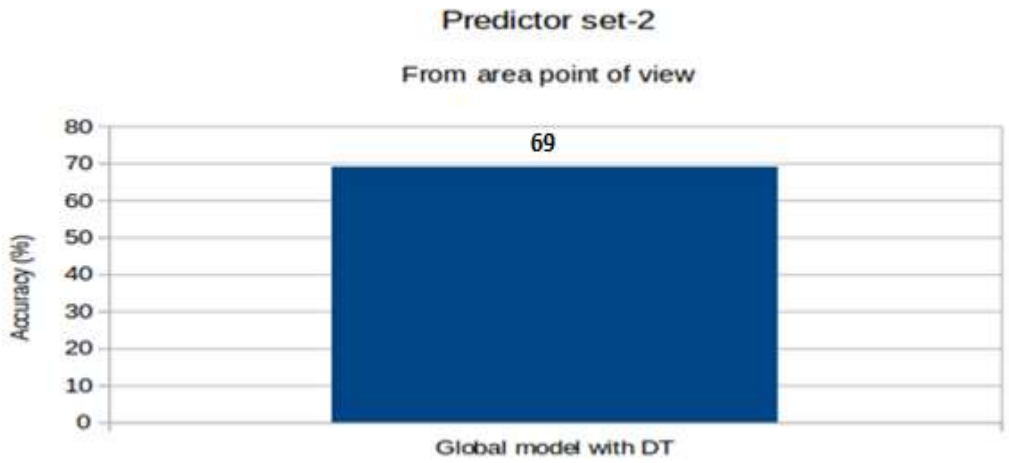

Figure 3.1a

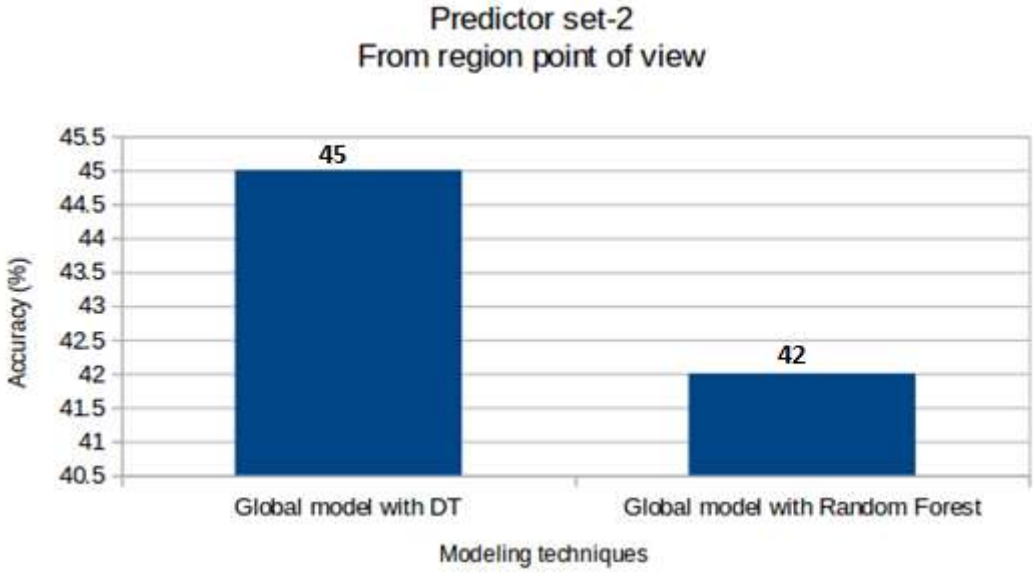

Figure 3.1b

Experiment results for **Predictor set-3:**

The models from the area point of view are shown to have better accuracy with 76%. The Figure 3.2a and Figure 3.2b below shows the accuracy of different approaches for area and region point of view respectively.

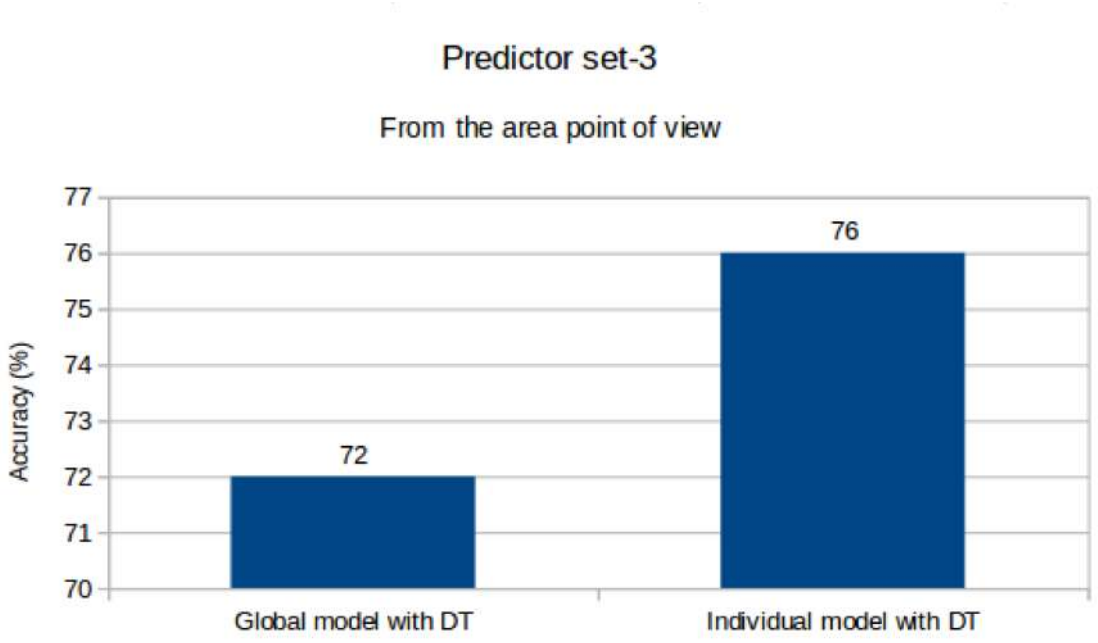

Figure 3.2a

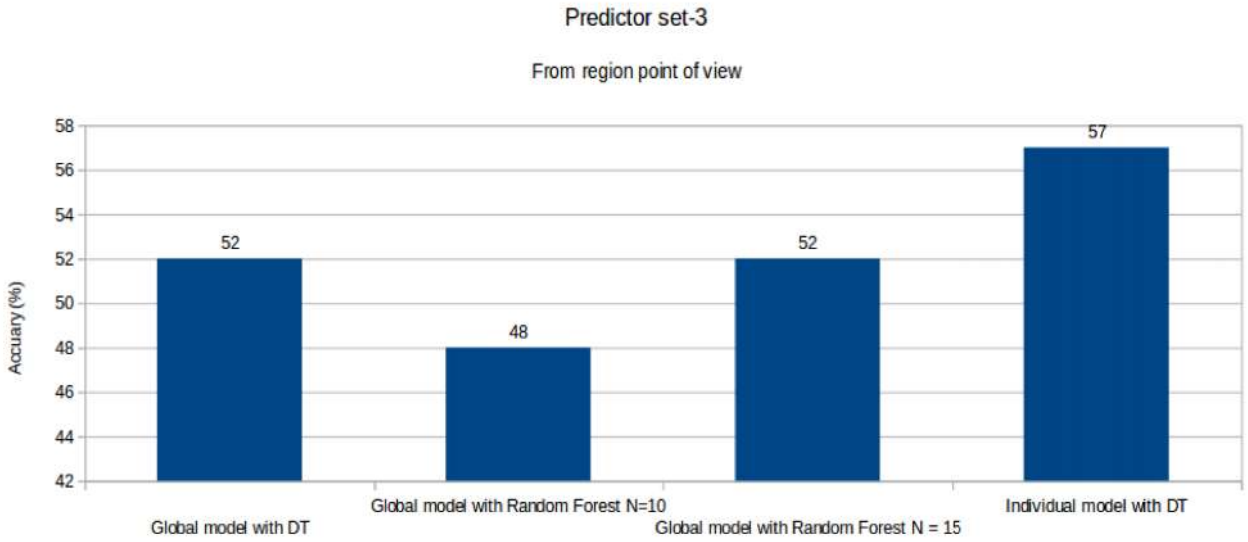

Figure 3.2b

Conclusion

In this project, we have addressed the problem of location inference by providing estimates of customer location, in the sense of trying to estimate (or predict), at a given point in time, which users will be served by which SC. We have employed tree-based approaches for modelling and predictions.

As learnt for the past works, the tree-based approach, using Decision with the support of Random Forest, give us a very good performances. Thus far, we have used Decision trees & Random Forest for modelling with different set of predictors / features and obtained varied performance accuracies as presented earlier.

Bibliography

[1] Yu Zheng, Like Liu, Longhao Wang, Xing Xie. Learning Transportation Modes from Raw GPS Data for Geographic Application on the Web, In Proceedings of International conference on World Wild Web (WWW 2008), Beijing, China. ACM Press: 247-256

[2] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie. Understanding Mobility Based on GPS Data. In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008), Seoul, Korea. ACM Press: 312–321.

[3] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, Wei-Ying Ma. Understanding transportation modes based on GPS data for Web applications. ACM Transaction on the Web. Volume 4, Issue 1, January, 2010. pp. 1-36

[4] http://www.openstreetmap.org/?minlon=-179.9695933&minlat=1.044024&maxlon=179.9969416&maxlat=64.751993

[5] Andras Garzo, Andras A. Benczur, Csaba Istvan Sidlo, Daniel Tahara, and Erik Francis Wyatt. Real-time streaming mobility analytics. In Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013, pages 697–702, 2013.

[6] Vincent Etter, Mohamed Kafsi, Ehsan Kazemi, Matthias Grossglauser, Patrick Thiran
Where to go from here? Mobility prediction from instantaneous information.
School of Computer and Communication Sciences, EPFL, CH-1015 Lausanne, Switzerland, July 2013

[7] Jingjing Wang. Periodicity based next place prediction, 2012.

[8] Anastasios Noulas, Salvatore Scellato, Neal Lathia, Cecilia Mascolo, "Mining User Mobility Features for Next Place Prediction in Location-Based Services", ICDM, 2012, 2012 IEEE 12th International Conference on Data Mining (ICDM 2012), 2012 IEEE 12th International Conference on Data Mining (ICDM 2012) 2012, pp. 1038-1043, doi:10.1109/ICDM.2012.113.