Predictability of Human Behavior using Mobility and Rich Social Data

Ana Martic

Kongens Lyngby 2013
IMM-M.Sc-2013-45

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc-2013-45

Summary

This thesis explores how predictable human mobility is, and whether knowing about mobility patterns of other people, who visit same places, can contribute to better prediction results. Human movements are periodical to some extends, which means that it is possible to create a model which can predict next place of a person in some moment based on the data about previous person's movements. In this thesis, an ensemble method is adopted, which gathers predictive power of multiple models, each capturing different human mobility features. The predictive models are build using GPS data collected for 136 experiment participants, during seven and a half months period. Prior to predictive modeling, data was carefully preprocessed and characteristics of human mobility are analyzed using multiple visualization techniques.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Informatics.

This thesis describes the tasks performed with the goal to create predictive models which can predict next place using past mobility data.

Lyngby, 17-June-2013

Ana Martic

Acknowledgements

I would like to thank my supervisors, Jakob Eg Larsen and Sune Lehmann, for their help and their guidence during the course of my work on this thesis.

I would also like to thank professors Morten Mørup and Mikkel Nørgaard Schmidt, and Ph.D. students Vedran Sekara and Piotr Sapiezynski, for their feedback and ideas which contributed my thesis.

Finally, I would like to thank my parents, sister and by boyfriend for their support during my master studies.

Contents

Chapter 1

Introduction

Every person follows a daily routine, imposed by their daily commitments and habits. Most people go to work every morning and are at home during the sleep time. Some spare time activities are also done on regular basis, such as visits to the same gym, or a favorite bar. Therefore, human mobility shows both temporal consistency, because certain places are visited periodically, and geographical consistency, as people are likely to return to places they have visited before.

This thesis tries to answer to which extent people follow patterns and to which extent their movement can be predicted when knowing the history of their previous movements. Authors of [SQBB10] conducted a research on 50.000 selected cell phone users and concluded that up to 93% of human movements can be predicted with the right prediction algorithm. The estimation of the limit of human predictability is based on empirically determined entropy of people's trajectories represented as a time series. Knowing people's mobility patterns can be used in various areas: traffic congestion control, network bandwidth provisioning [SDK+06], location-aware recommendations [ZZM+11], epidemics prevention etc.

This research is based on data collected during SensibleDTU experiment for the period from October 2012 to mid May 2013. All the experiment participants are DTU students on the first year of Bachelor studies. Students form a group which

is particularly hard to predict. Their daily schedule is more flexible than the one of employed people and they are more mobile during their "working" hours, since they have classes at various buildings. In addition, their daily patterns change throughout a year due to changes of courses timetable every semester, or changes of home address.

The tasks performed during this research include data preprocessing, data visualization and, finally, next place prediction. Data preprocessing consists of data cleanup and conversion between raw GPS location records to meaningful stay regions. The places visited by a user are detected using grid-based clustering algorithm proposed at [ZZXY10] and improved at [MGP10, DGP12] In the next step, various visualization techniques are applied to get insight into the dataset with the focus of discovering which factors influence human predictability. Knowledge from data visualizations is then used to improve the predictive model proposed at [DGP12] and apply it to this dataset. The prediction method proposed by [DGP12] combines conditional probability distributions of the output variable given the set of contextual variables which include: current location, hour, day of the week, weekend indicator, frequency and duration of visits to the current location. My contributions include changing the model to account for students' academic calendar, adding previous location and current location popularity as new contextual variables and finally an attempt to increase statistical power of the predictive model by including data from other similar users.

The thesis is organized as follows. Chapter "Related work" gives an overview of related research about predictability of human behavior. Chapter "Data collection" provides a brief description of how the mobility data was obtained. Chapter "Data preprocessing" summarized the steps taken to prepare data for further data mining. Chapter "Visualization" focuses on various characteristics of human mobility which can be inferred by visualizing the data set. Chapter "Next place prediction" describes predictive models which can predict the next place and includes analysis of the prediction results. Chapter "Conclusion" highlights main conclusions about human mobility predictability.

Chapter 2

Related work

Previous studies on human mobility differ by the sources of mobility data, the method used for discretization of geographical data, the predictive models they propose and the granularity of predicted location.

Before smart phones, containing GPS sensors, became widely available, cell phone tower id's were used for tracking location of cell phone users. When someone makes a phone call, his location is recorded based on the id of the nearest cell phone tower. Cell phone tower ids provide only coarse location estimation. Cell phone tower logs are used at [CML11, SQBB10].

Other popular data sources include check-ins from social networks such as Foursquare or Facebook [CML11, NS12, AN12], WiFi traces [SKJH03, SMM+11], and finally data from GPS sensors [SK12, DGP12, SMM+11]. Company Raytheon created a tool called Riot (Rapid Information Overlay Technology)[1] which tracks people on the Internet by combining location data obtained from check-ins on different social networks and pictures uploaded to the Internet.

Most prevalent approaches to location data discretizetion include grid-based clustering [CML11, SK12, MGP10] and density-based clustering, where DBSCAN [ZFL+04] and Density-Joinable Cluster [ZFL+04, GKdPC12] are used.

[1] http://www.guardian.co.uk/world/2013/feb/10/software-tracks-social-media-defence

In the presented work, both approaches are considered, and will be discussed in the following sections.

Cho et al. in [CML11] present prediction model based on geographic and temporal periodicity of human movements and existing social ties. A movement is considered to be influenced by social ties if a user is found in vicinity of his friend's home or if user's friend made a check-in at particular location prior to user's movement. They propose an ensemble method where they combine a spatiotemporal probabilistic model to predict human movements between "home" and "work" locations depending on time, and a social model to predict the movement to locations classified as "other", by spatiotemporal model. They conclude that long term travel is more influenced by social ties then the short term travel.

Sadilek and Krumm [SK12] propose a model which can predict someone's place at any time in future. The predicted place is one of the 10 most visited locations or 11th location which captures all the other places. Location is modeled as a triangular cell with 400 m long sides. The model is a matrix where rows contain days and columns contain visited location for every hour of a the day, day of the week and a holiday indicator. The proposed prediction algorithm is based on PCA analysis. PCA showed that for all subjects, 10 days (eigendays) are enough to reconstruct one's entire location history with more than 90% accuracy. Prediction is done by projecting the test vector to principal components (eigendays) and choosing the day with the highest weight. Similar study was previously described at [EP09], where features include location, modeled as one of the states "Home", "Work", "Elsewhere", "No Signal", and "Of State", for every hour of a day.

Authors in [AN12] fit two supervised regressors to the model built upon users' check-in data from Foursquare, to predict the ranking of the places within one city, where particular user might check-in within next 24 hours. For every place in a city, they calculate features categorized as user mobility features, global mobility features and temporal features. The features which showed the highest performance include: Categorical Preference (the number of visits to a particular category of places in the past) , Place Popularity (total number of check-ins at the venue) , Geographic Distance (the distance between current venue and all other places) and Place Hour (the number of past check-ins at the particular place during a particular hour of a day).

Song et al. [SKJH03] propose a 2-order Markov model with fallback to 1-order Markov model for on-campus next place prediction. State in Markov model is modeled as a location history containing two or one past location, in case the previous location is missing. Transitions in Markov model are possible locations that follow particular state, where the most probable transition is given as the

next place prediction. Markov based models are also used at [NS12, SDK+06, GKdPC12].

Authors of [YLWT11] proposed a novel approach where users are firstly clustered based on the similarity of their semantic trajectories, and next place prediction for a single user is done using geographic trajectories from all users in the same cluster and the given user's personal semantic trajectories. A geographic semantic information database is used in order to assign semantic labels to location points, and transform a geographic trajectory to a semantic trajectory. They use Prefix-Span algorithm to discover prediction rules, so that every trajectory with support higher than 50% is transformed to a prediction rule. Prediction is done by searching through the prefix tree, containing the prediction rules, for the path with the greatest support and the longest length which matches the current trajectory.

In this thesis, I try to reproduce and improve the next place prediction method presented at [DGP12]. Authors at [DGP12] propose an ensemble method, where conditional probability distributions over different set of input variables are combined into a single more powerful probabilistic model. This approach is chosen because the proposed predictive model considers multiple characteristics of human mobility and because it is easy to extent with new input variables.

Chapter 3

Data collection

Data collection started in October 2012, as a part of SensibleDTU project. Participants of the experiment are 136 first year students at DTU. As part of the experiment, students were handed Android smart phones and asked to use them as their primary phone.

Data is collected using an application based on modified version of open source framework called Funf Open Sensing Framework, which supports multiple probes. Probes used within SensibleDTU project include: Location, Bluetooth, Cell phone ids, Wi-Fi, Contact, SMS, Call log, Facebook, Screen, Battery etc. The application is deployed to phones and it collects data with predefined sampling rate. Data samples can be temporarily saved on the phone until there is a Wi-Fi access, when they are uploaded to the central SensibleDTU server. Data can be retrieved from SensibleDTU server by querying an API which returns response in JSON format.

This thesis only considers location dataset. Location data is sampled every 15 minutes for the next 30 seconds. It is provided by Android Location Services which use GPS sensor, Wi-Fi or cell tower ids as data sources. Location data provided by GPS has the highest accuracy, but it requires higher power cost and it is not available indoors. Android Location Services sometimes provide less precise location based on visible Wi-Fi networks/cell tower ids, by maintaining a mapping between known Wi-Fi hotspots/ cell tower id's and geographical

coordinates.

Every location object in location dataset contains information about latitude, longitude, timestamp, location accuracy, and various other information which was not considered in this work.

Chapter 4

Data preprocessing

4.1 Data cleanup

During data collection stage, various problems occurred affecting data quality. This section describes those problems and undertaken strategies to deal with missing data and invalid data.

4.1.1 Missing data

Some users joined the experiment late, while others left the experiment early. Figure 4.1 displays the date of the first and the last location point recorded for each user. The time from the first to the last location observation for a particular user is referred to as observation period, in further text. Users whose observation period is shorter than 80% of the overall observation period for all users (marked with horizontal lines on the plot at 4.1) are excluded from any further analysis, because results obtained using their data would be misleading.

The time interval between any two consecutive location points should be no longer than 15 minutes. However, it occurs that location points are not sampled as scheduled, due to one of the following reasons:

- Turned off phone
- Battery exhaustion
- Signal loss

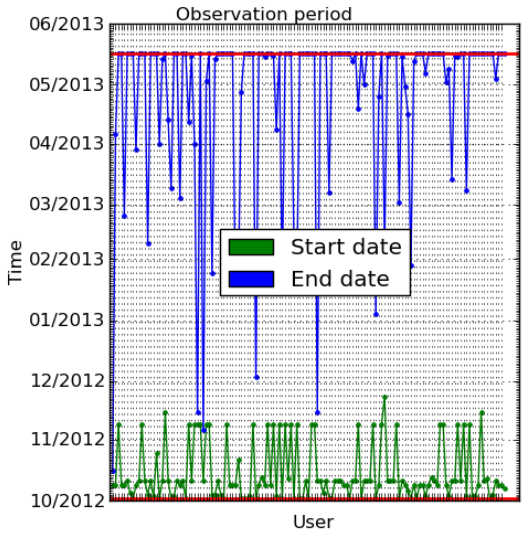


Figure 4.1: Observation period

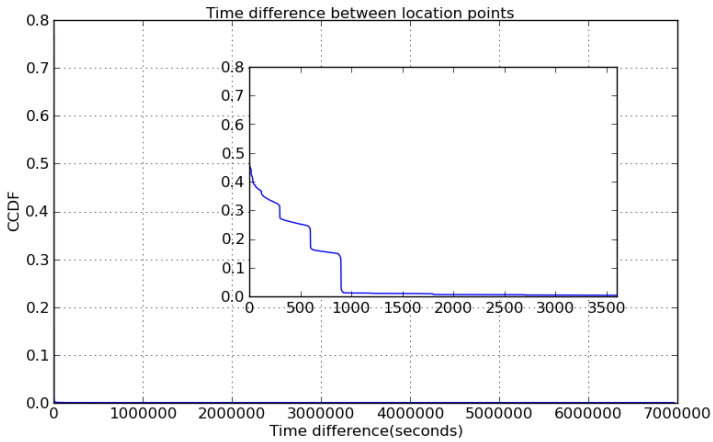


Figure 4.2: Complementary cumulative distribution function of sampling rate. Median = 1.0 second, Mean = 351.18 seconds, Standard deviation = 10547.23.

- Turned off GPS sensor

Figure 4.2 shows the CCDF of sampling rate for all users. It can be observed that the time difference between two location points is most commonly few seconds, 5 minutes, 10 minutes or 15 minutes and in rare cases (around 2 %), it is higher, with maximum value of around 7000000 seconds (81 days).

Therefore, when time interval between two consecutive location points is higher than 15 minutes, location of the user is considered unknown for the time period which is 15 minutes lower than the time between the two location points.

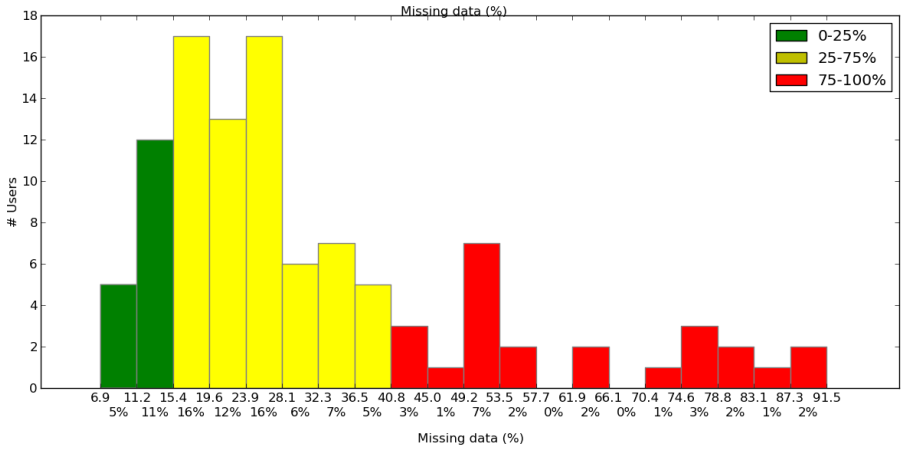


Figure 4.3: Distribution of percentage of missing data per user. For every bin, bin edges and a percentage of data it contains are displayed at the x axis.

Figure 4.3 shows the percentage of missing data per user, calculated as the ratio between the total time interval when user's location was unknown and the total length of the observation period for given user. The users which have more than 35% of missing data are removed from the data set. After this step, 75 out of 136 users remain.

For a comparison, a dataset, containing Bluetooth proximity data, described at [EP06], contains data for 85.3 % of time since the start of the experiment. In 14.7 % of cases, missing data occurs because users tend to turn off their phones during the night. In terms of next place predictability, it is important to know whether missing data is located in particular interval of the day or a week. Fiigure 4.4 shows the number of missing location points for every hour of the week, after users with over 35% of missing data were removed. It can be

observed that missing data mostly occurs during weekend, due to users probably forgetting to charge their phone for a long period of time.

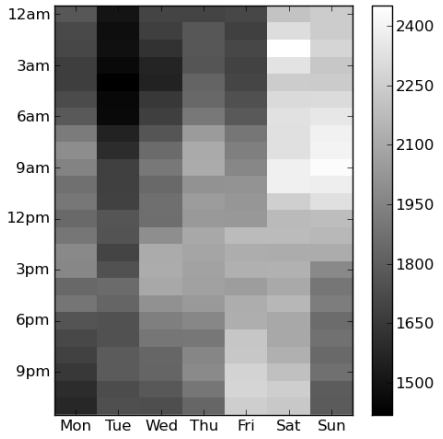


Figure 4.4: Number of missing location points per hour of the week

4.1.2 Invalid data

Location points come with an accuracy parameter, which is received from Android Location Services. If accuracy is less than 100m, the location point will be marked as invalid.

The author of [Cut13] proposed an algorithm for removal of isolated location points. By studying user trajectories, he noticed that location point jumps may occur. Namely, there are location points which are too far away to be the part of user's path, considering user's speed on that path, so it is likely that they appear due to errors of GPS. The proposed algorithm identifies a location point as isolated, if the speed between the location point and the previous location point is very high (> 1 m/s), and the speed between previous and next location point is very low (< 0.5 m/s). This algorithm is also adopted in this thesis. It checks all location point triplets which are sampled at the regular sampling rate and marks isolated points as invalid.

In this step, 11% of location points is marked as invalid. They are not removed because they are taken into account in other analysis where it is important to know whether location points are missing within some time interval.

4.2 Stay points and stay regions

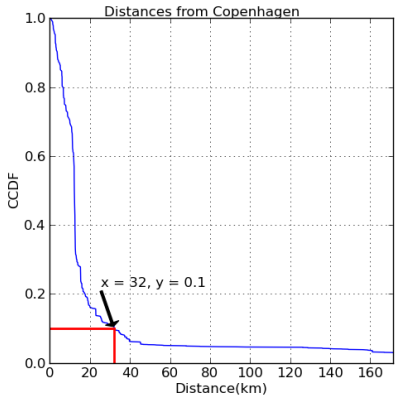
Visited places detection is performed according to the method proposed at [ZZXY10]. The methods consists of 2 steps: *a*) stay points detection; and *b*) stay regions detection using grid-based clustering; Zheng et al. define stay points as a group of consecutive location points which is constrained by maximum distance (maximum distance threshold), and minimum time (minimum time threshold) between the first and the last location point. In this thesis, maximum time threshold is used as well, to limit the time which passed between two consecutive location points, as suggested at [MGP10]. This is necessary from the perspective of determining duration of stay at certain location. It can happen that user's trace is lost for certain amount of time between two consecutive location points which are located close to each other. For example, someone can stop providing data samples while being at home, then move to some other location for some time and come back home and start sampling again. In this case it may seem that the user stayed at home for a very long time, which is not true. In this thesis, maximum distance threshold is set to 100 meters, minimum time threshold is set to 20 minutes and maximum time threshold is set to 30 minutes. This means that, it is considered that an individual stayed at some place if he spent more than 20 minutes within a radius of 100 m. Other sets of parameters were tested as well, but this one is selected based on prediction results.

Stay point detection algorithm sets start time, end time and coordinates for every stay point. Start and end time are set as timestamps of the first and the last location point, respectively. The coordinates are calculated as average of all location point coordinates within the stay point.

In the next step, stay points are clustered into stay regions using grid based clustering algorithm. The grid-based clustering algorithm requires dividing the world into uniform grid cells. Since most location points lay in the area close to Copenhagen (shown by CCDF of distances between every location point and Copenhagen at Figure 4.5a), the geographic area for the experiment is limited by the radius of 45km from Copenhagen (Figure 4.5b). The size of every world cell is set to 100x100 m and the size of every stay region is set to 300 x 300 m, as in [ZZXY10].

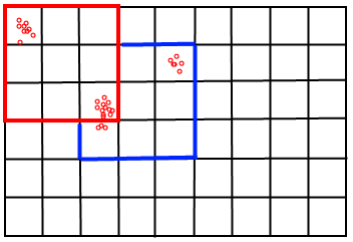
Every world cell is characterized by North and South geographical latitude and West and East geographical longitude. World cells are created by assigning coordinates of the North-Western corner of observation area to the first cell and calculating South and East coordinates using formula [1] which accepts the known

[1] http://www.movable-type.co.uk/scripts/latlong.html



(a) Complementary comulative distribution of distances from Copenhagen. Read lines show the point where 5% of the user have distance larger than x

(b) Resulting observation area

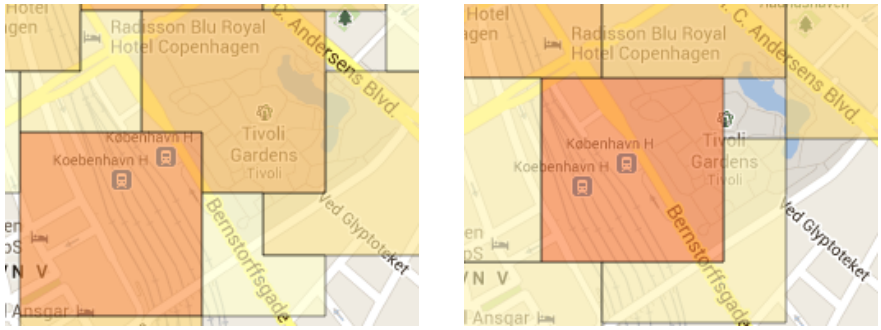


(c) Grid-based clustering. Algorithm starts with the cell in the 3^{th} row and 3^{th} column, which has the highest number of stay points. Stay region which covers the highest density is the one also containing the cell in the 1^{st} row and 1^{st} column. The cell in the 2^{nd} row and 5^{th} column has the highest density out of remaining cells. This cell is assigned to the region which does not form a full square, becuase two of its cells already belong to the first region.

Figure 4.5: Grid-based clustering

coordinates, distance (100 m) and bearing. The algorithm proceeds by creating adjacent cell in the same row, until the whole observation area is covered. The world grid is stored in a matrix where rows are populated in descending order of geographical latitude and columns are populated in ascending order of geographical longitude. Therefore, assigning a location point to a world cell is done in logarithmic time.

The grid-based clustering algorithm contains the following steps (see Figure



(a) Original grid-based clustering algorithm. Central Station and Tivoli Gardens are in separate regions.

(b) Implemented grid-based clustering algorithm.The most visited area of Central Station and Tivoli Gardens belongs to the same region.

Figure 4.6: Comparisson between two versions of grid-based clustering algorithm.

4.5c:

1. Creating world grid
2. Assigning stay points to world grid cells.
3. Selecting an unassigned cell with the highest density. If all the cells containing stay points are assigned, the algorithm finishes.
4. Creating a region with a unique ID
5. Choosing a square of dimensions of 3x3 cells which contains the cell and covers the highest possible density of unassigned stay points.
6. Assigning a newly created region to every cell, which does not belong to any other region, and to all stay points in the cell
7. Repeat from step 3.

In the original algorithm at [ZZXY10], step 5 was performed by creating a region of same size, which contains the highest density cell in the middle. The change in step 5 was proposed at [DGP12] in order to create more precise regions. With this approach, it takes less regions to cover the whole density, which has positive impact on the next place prediction accuracy. By observing stay regions on a map, I concluded that the original algorithm is better in separating semantically different locations (see Figure 4.6).

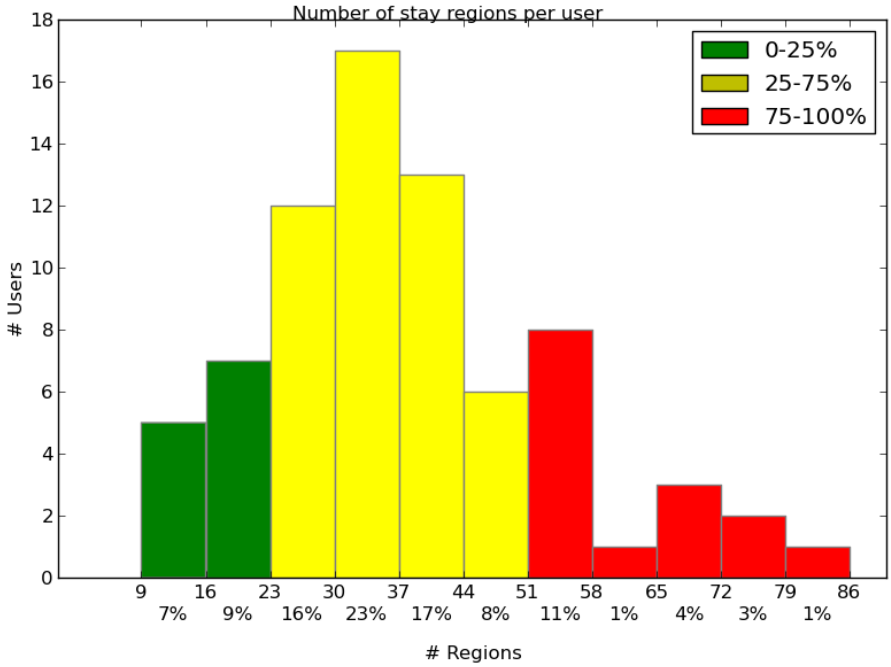


Figure 4.7: Number of regions per user

Figure 4.7 shows the number of detected stay regions per user for the entire period of 7.5 months.

After stay region detection, every stay point which lays within the observation area has its "region ID" assigned. All stay points outside the observation area will be removed. There might be consecutive stay points in one region, as maximum stay point area is smaller then stay region's area. In the next step, every two consecutive stay points with the same region IDs are merged into one if the time difference between them is lower than 30 minutes, or if there are location points every 30 minutes from the end of the first until the start of the second and if these location points lie within the same region , or any adjacent regions. This is done in order to disregard transitions within a single region. Namely, if someone stays at one place and moves to stay at another place in the same region, it is regarded as a single stay starting from the start time of the first stay until the end time of the second stay.

4.3 Comparison between grid-based clustering and DBSCAN

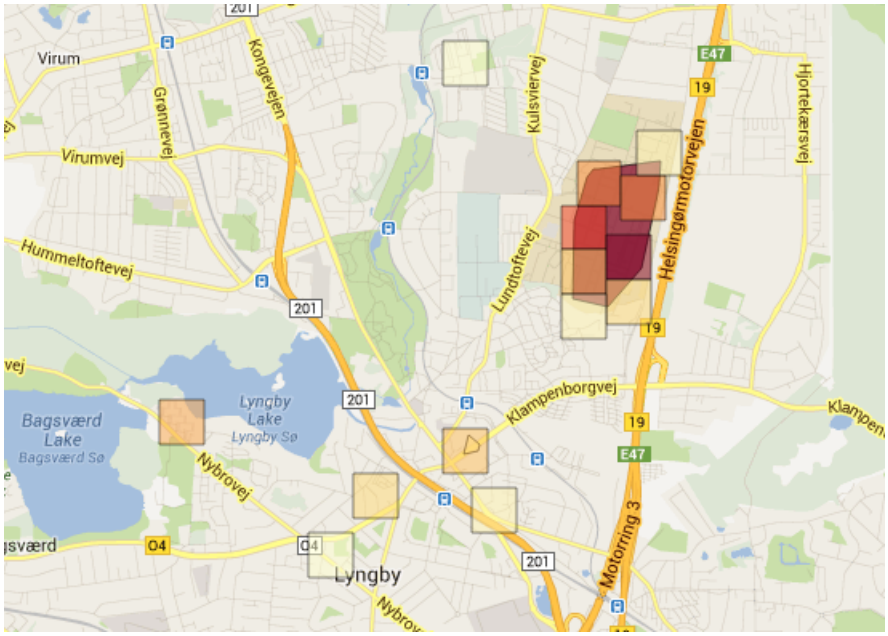


Figure 4.8: Stay regions and DBSCAN cluster. The shape of DBSCAN is approximated using convex hull algorithm

DBSCAN [2] is a representative of density-based clustering algorithms. It receives two parameters: MinPts, minimum number of points in the neighborhood, and Eps, maximum distance between neighboring points. The algorithm starts from the first stay point and it checks if there are any points in the point's Eps neighborhood. If the number of neighboring points is not less than MinPts, all previously not assigned neighboring points are added to the new cluster. Then, the cluster is expended to all other unassigned points which can be reached from the neighboring points with respect to Eps. Points which are not assigned to any cluster are considered outliers.

Figure 4.8 shows detected regions and DBSCAN clusters at the surrounding area of DTU campus. The color of the overlays signifies the importance of the area with respect to how much time a user spent there on a log scale. More important regions/clusters have darker color. Stay points detection algorithm was also run prior to DBSCAN clustering, using the same parameters as for

[2] http://en.wikipedia.org/wiki/DBSCAN

stay regions detection. The parameter MinPts is set to 2 and Eps is set to 200 m.

The differences between two clustering algorithms can be summarized as follows:

- Stay regions cover the entire density while DBSCAN leaves not frequently visited places out. In terms of predictability, it is reasonable to leave out the places which are visited once or twice during a long period of time. This can be achieved with grid-based clustering approach by filtering out the regions depending on the frequency of visits over some time period.

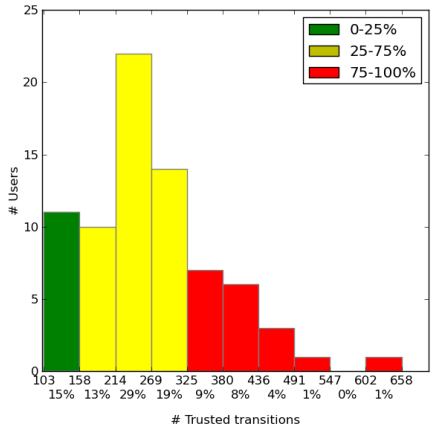
- All stay regions have equal size, while DBSCAN produces clusters of different sizes and shapes. Having places which are close to each other clustered together is a good idea if the prediction goal is just to get a coarse location of user. However, DBSCAN is not capable of separating semantically different locations. For example, if a user lives on campus, it is likely that his home and all university buildings would be detected as the same place, so the next place prediction would run only for transitions between one region, where user spends the most of his time, and few other regions, which are probably not visited on regular bases.

- DBSCAN algorithm requires storing a table containing distances between every pair of stay points.Therefore, its complexity is quadratic, while stay region detection algorithm has linear complexity with respect to the number of stay points.

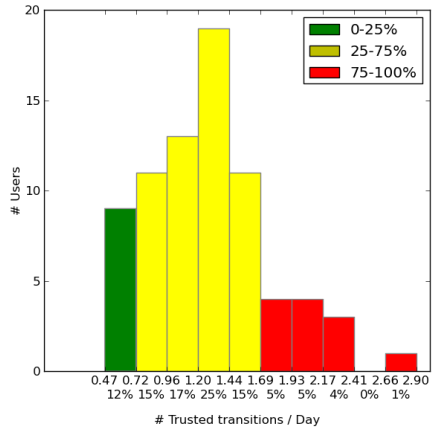
Having taken these issues under consideration, I decided to use grid-based clustering algorithm for place discovery.

4.4 Trusted observations

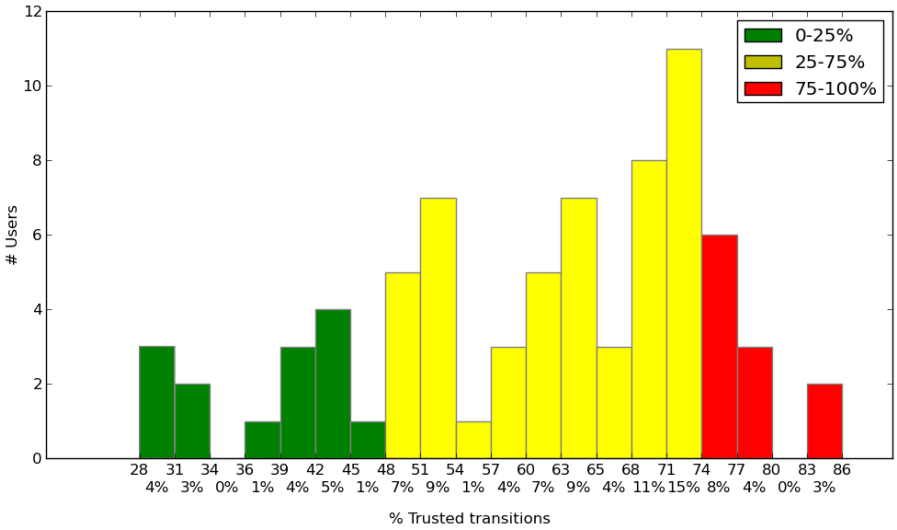
Due to missing data, some visits might not be recorded at all, or partially recorded, with incorrect start and end time. In data mining for predictive modeling, it is important to know whether some visit happened immediately after another visit and whether the stay duration of some visit is trustworthy. Trusted observations are, therefore, introduced, as means for additional data cleanup necessary for some models. The concept of trusted observations was previously used at [DGP12].



(a) Distribution of number of trusted transitions per user



(b) Distribution of average number of trusted transitions per day per user



(c) Distribution of percentage of trusted transitions per user

Figure 4.9: Trusted transitions per user

4.4.1 Trusted transitions

A transition between two places is trusted if location points are recorded during the transition time in a time interval, specified by some threshold. Do et al.

[DGP12] set the time interval threshold to 10 minutes, as the minimum stay duration is set to 20 minutes, so by knowing that data was recorded every 10 minutes, it is certain that another unobserved visit did not occur in between. In this thesis, the time interval threshold is set to 30 minutes because the sampling rate of location points is 15 minutes and because of the need to be more tolerable to errors in sampling in order to keep more data for predictive modeling. If transition time between two visits is lower than 30 minutes, or if location point is recorded every 30 minutes from the end time of first visit to the start time of the second visit, that transition is considered trusted.

Trusted transitions are converted to feature vectors and used in predictive model. On average, a user has 265 trusted transitions during the whole observation period (see Figure 4.9a), 60.57% of transitions are trusted transitions (see Figure 4.9c) and the average number of trusted transitions per day is 1.26 (see Figure 4.9b).

In order to test if the algorithms for stay points and trusted transitions detection correctly represents the recorded location points for a user the following visualizations are created:

- A matrix showing the number of location points in an hour of a day for a month long period (see Figure 4.12a).
- A matrix having days as rows and hours as columns where cells show start and end time of transitions and visits (see Figure 4.12b).
- A map which shows location points, stay points with start and end time, transitions between stay points for one day of recording (see Figure 4.13).

4.4.2 Trusted visits

A stay point is considered as a trusted visit if location points are recorded during a specific period of time before and after the visit. Trusted visits are introduced in order to know whether start time and end time of some visit are trustworthy. For example, if a user starts recording data at some location, after a long period without recording, the time of the first location point will be considered as a start time of a visit. Such visit will not be trusted, as user might have arrived to that location long before the first location point was recorded. Trusted visits are used in models when it is important to know the exact stay duration at some place.

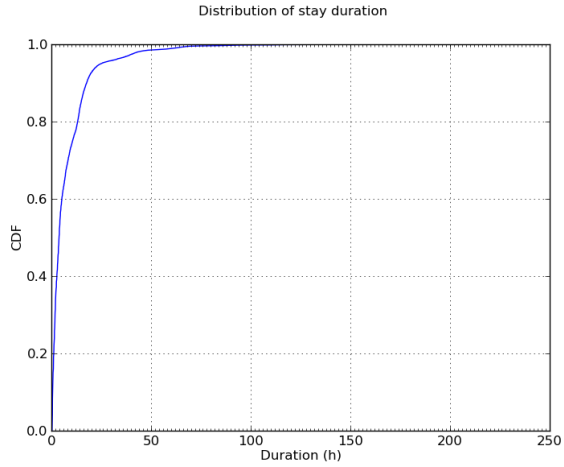
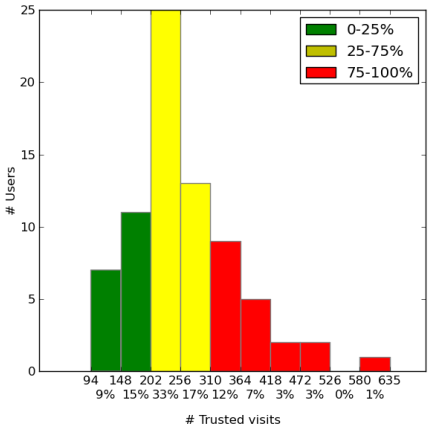


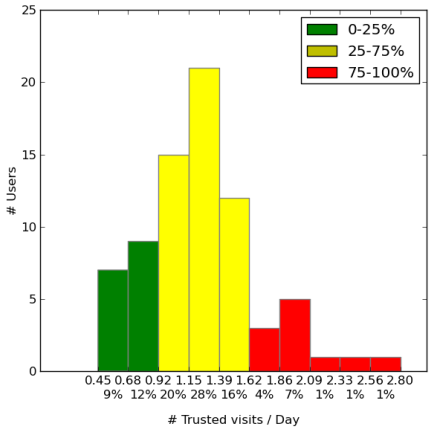
Figure 4.10: Distribution of stay durations. Mean: 7.92; Standard deviation: 11.85; Mode: 1; Median: 3.83; Min: 0.33; Max: 221.5.

Figure 4.10 shows the cumulative distribution function of a stay duration for trusted visits of all users. It can be observed that short stay durations are dominant, while, in rare cases, stays last for a few days. However, as explained in section 4.2, someone is considered to be staying at some location, as long as he does not move and stay at another location for at least 20 minutes.

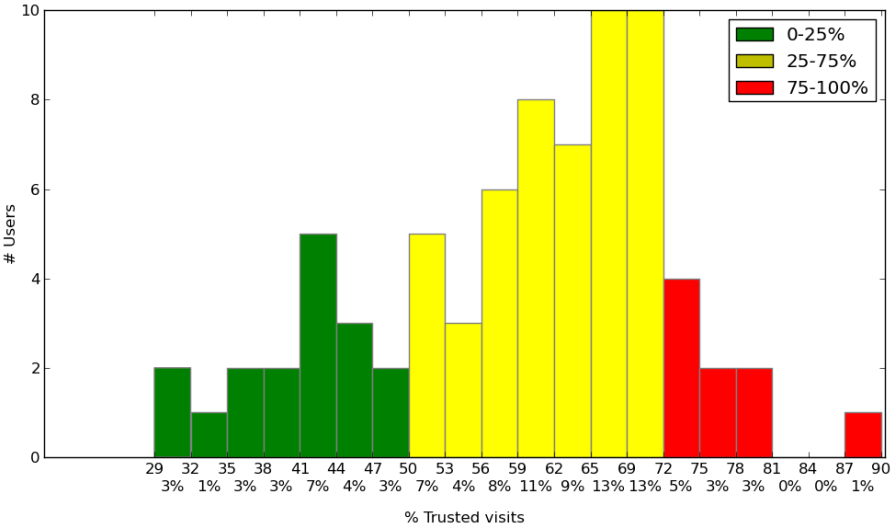
The time threshold for trusted visits is set to 30 minutes. On average, a user has 261 trusted visits during the whole observation period (see Figure 4.11a), 59.69% of stays are trusted visits (see Figure 4.11c) and the average number of trusted visits per day is 1.24 (see Figure 4.11b).



(a) Distribution of number of trusted visits per user

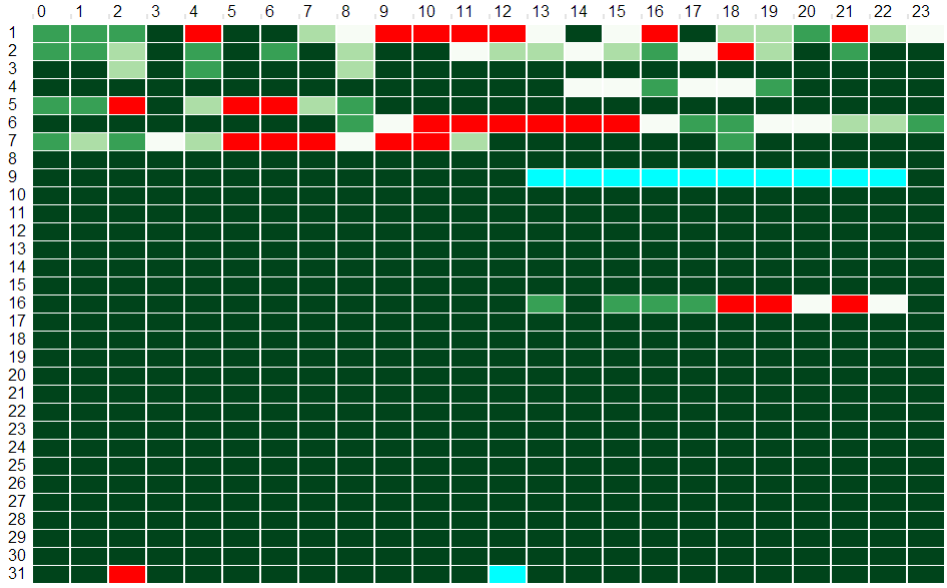


(b) Distribution of average number of trusted visits per day per user

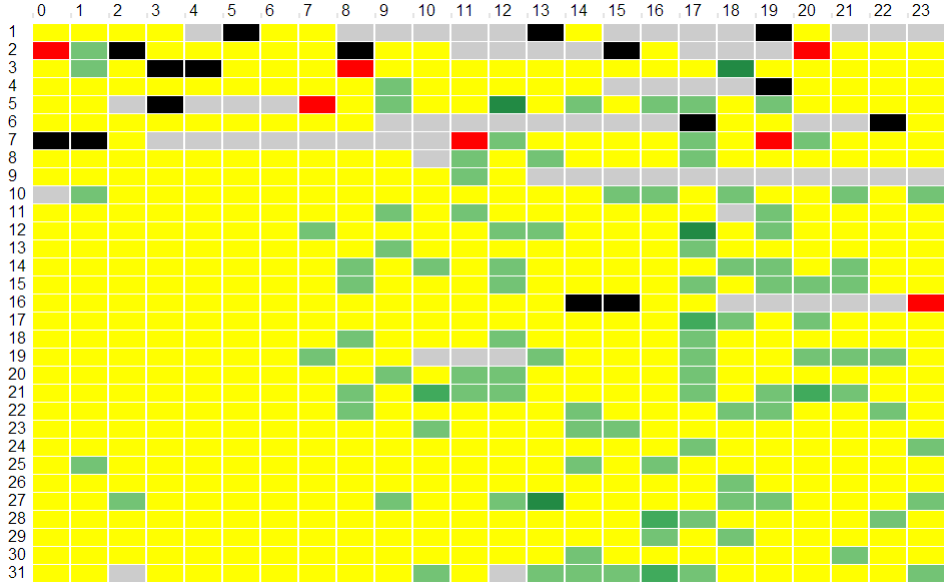


(c) Distribution of percentage of trusted visits per user

Figure 4.11: Trusted visits per user



(a) Number of location points per hour. Red - no location points. Blue - location points are outside of the observation area. Greens - number of location points depends on a shade of green. Dark green - at least one location point every quarter of an hour. White - there are location points only in one quarterly interval.



(b) Visits and transitions start and end time. Green - arrival to a new location when both previous and next locations are known. Red - arrival to a new location when previous location is unknown. Black - arrival to a new location when both previous and next locations are unknown. Yellow - stay at some location.

Figure 4.12: Matrix visualizations of user's behavior over a month

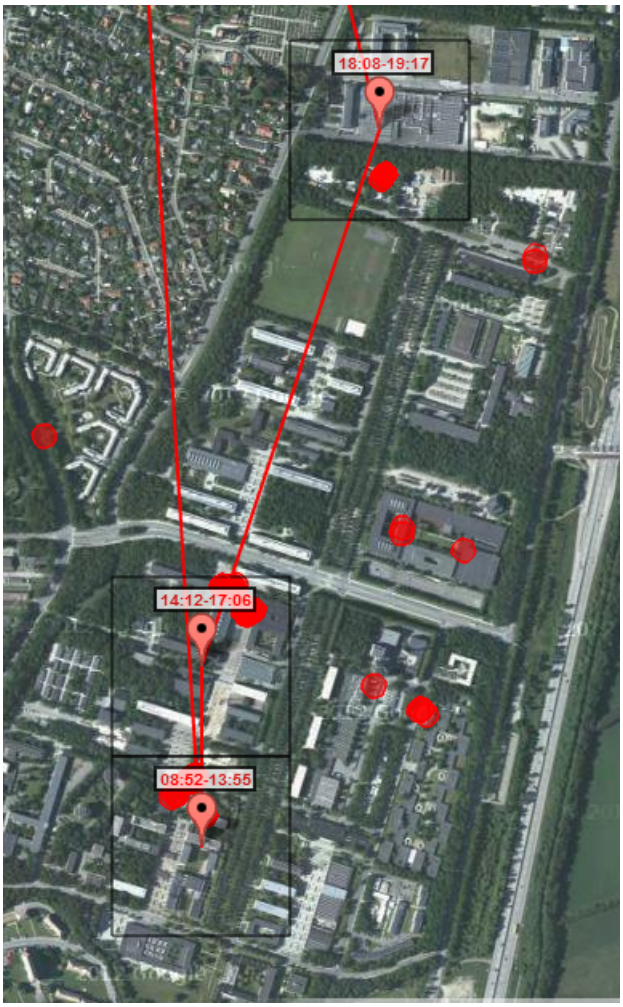


Figure 4.13: Visualization of location points, stay points, regions and trusted transitions on the map for 22^{th} day of a month. Stay points are marked with a marker with a label showing the start and the end of the visit; regions with squares; location points with circles. Map shows three visits where both previous and next locations are known.

Chapter 5

Visualization

This chapter contains visualizations of the data set, which reveal more characteristics of human mobility.

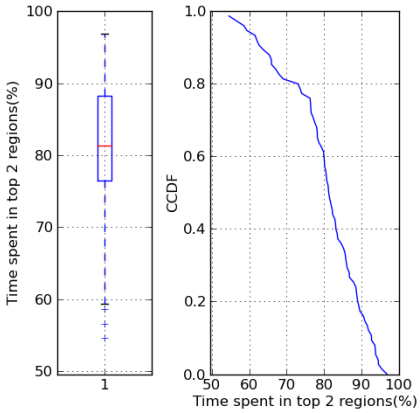
5.1 Time distribution

The goal of visualizations in this section is to show how many regions is enough to explain most of users' movements.

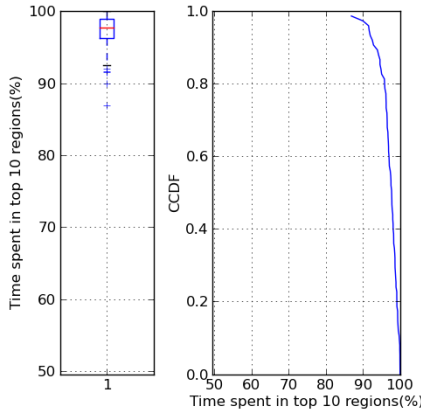
Figures 5.1a and 5.1b show the percentage of time spent in top 2 and top 10 regions, respectively, out of total time spent in all detected regions. On average, a user spends around 84% of his time in the top 2 regions, and 98% of time at top 10 regions. Figure 5.1c shows how many regions explain over 95% of user's mobility.

Figures 5.2 to 5.5 depicts how detected regions are distributed in space and how much time is spent at each region for three users. Title of the graph shows the number of detected regions and the number of regions which account for 95% of user's time. Users at Figures 5.2, 5.3 and 5.4 have the number of detected regions equal to minimum, median and maximum in the whole dataset, respectively. It

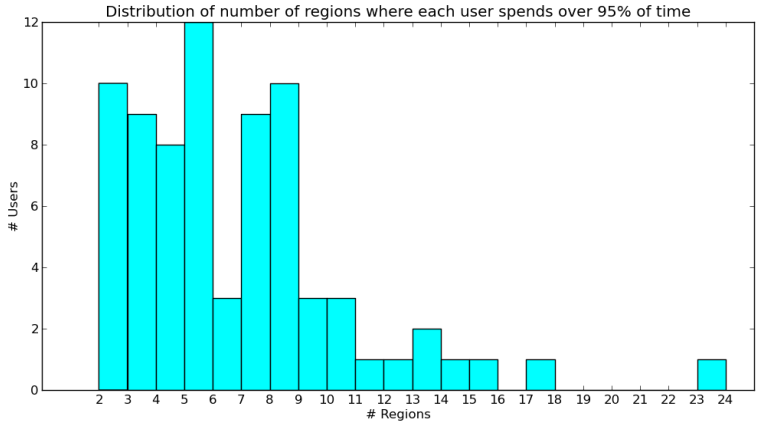
can be observed that there is a single location where users spend the most of their time - home location. This is the case for most of the users, while some have most of their time divided between two dominant regions (Figure 5.5).



(a) Time spent in top 2 regions. Mean: 80.36; Standard deviation: 10.20; Median: 81; Min: 55; Max: 97



(b) Time spent in top 10 regions. Mean: 97.75; Standard deviation: 2.53; Median: 98; Min: 86; Max: 100



(c) Distribution of number of regions where each user spends over 95% of time

Figure 5.1: Distribution of time spent at the most important regions

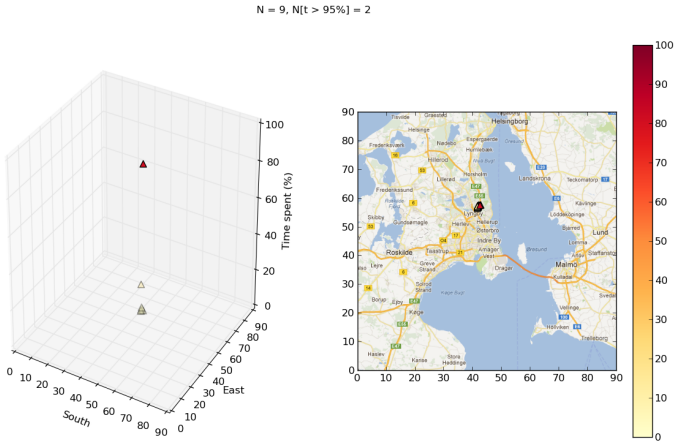


Figure 5.2: User A. Number of detected regions is 9. Number of regions which account for more than 95% is 2.

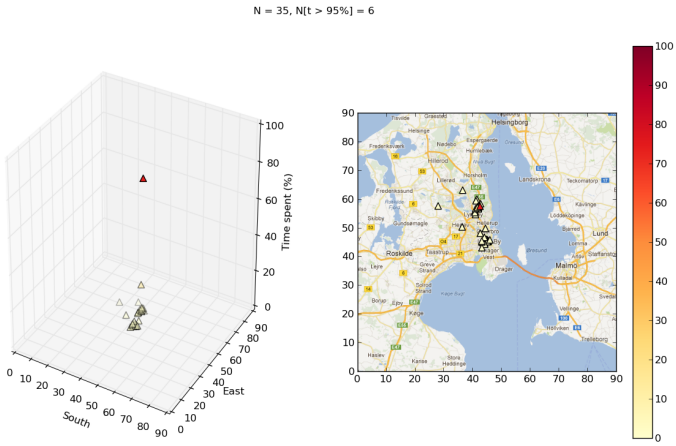


Figure 5.3: User B. Number of detected regions is 35. Number of regions which account for more than 95% is 6.

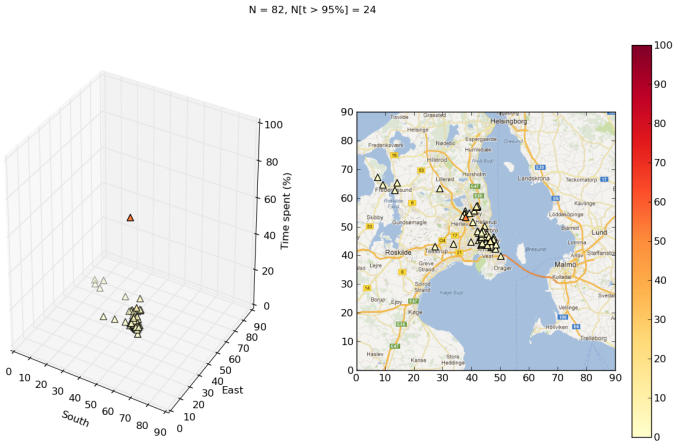


Figure 5.4: User C. Number of detected regions is 82. Number of regions which account for more than 95% is 24.

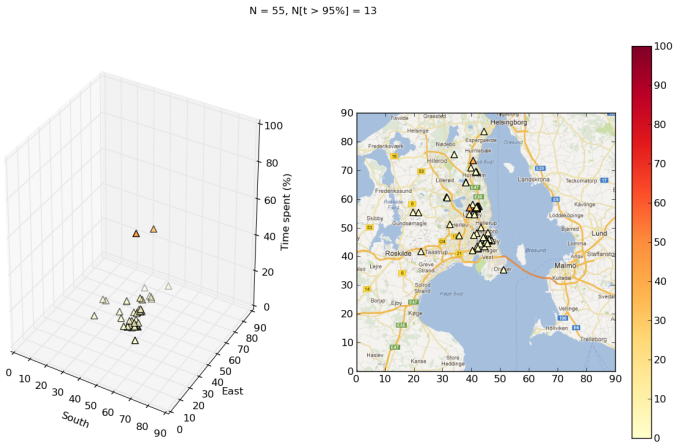


Figure 5.5: User D. Number of detected regions is 55. Number of regions which account for more than 95% is 13.

5.2 Categories of places

The purpose of visualizations in this section is to analyze whether place's popularity, frequency and duration of stays can indicate a particular semantic category a given place belongs to.Since dimensions of a stay region are 300x300m, it can contain multiple semantic places. Therefore, a category of a stay region cannot be used reliably in predictive modeling.

Figure 5.6 shows stay duration probability distributions at stay regions of different categories. The stay regions include two home places, two places at DTU, where students have lectures and do project work, Copenhagen Central Station and the area including a part of pedestrian street in Copenhagen.Width of each bin, at the main probability distribution plots, corresponds to a stay duration of 1h . Expectedly, long, over 12h stays, occur only at home places. Students most likely stay at DTU places for 2 or 4 hours, while stays at central station are rather short.

User's home place is set to a stay region where he spent the most of his time. Total stay duration is calculated for the period of last 12 weeks, because there is a higher chance of users moving to another place over a longer period of time. The resulting home places are compared with the home places determined as stay regions where users spent most of their time in the interval between 2^{am} and 6^{am}. The home places were not matching only for one user.

Figures 5.7 and 5.8 show characteristics of stay regions from four categories: "Homes", "Dorms", "DTU" and "Other". Parameter values for every stay region are offset for a random value in the interval between -0.2 and 0.2 on both axes, in order to split stay regions with equal values.

Stay regions containing dormitories are detected using text search feature from Google PlacesAPI[1]. For every stay region, a request is sent, to search for nearby locations matching the search word "kollegier". For each dormitory in the search result, it is verified whether it is located within the given stay region's bounds, in which case a "dorm" category is assigned to that stay region. Only dormitories where users live are labeled as "Dorms" at the plots, with "Dorms" category having precedence over other categories.

"DTU" stay regions are determined manually, based on a map showing all detected stay regions as overlays with attached info box containing stay region's ID.

[1] https://developers.google.com/places/documentation/

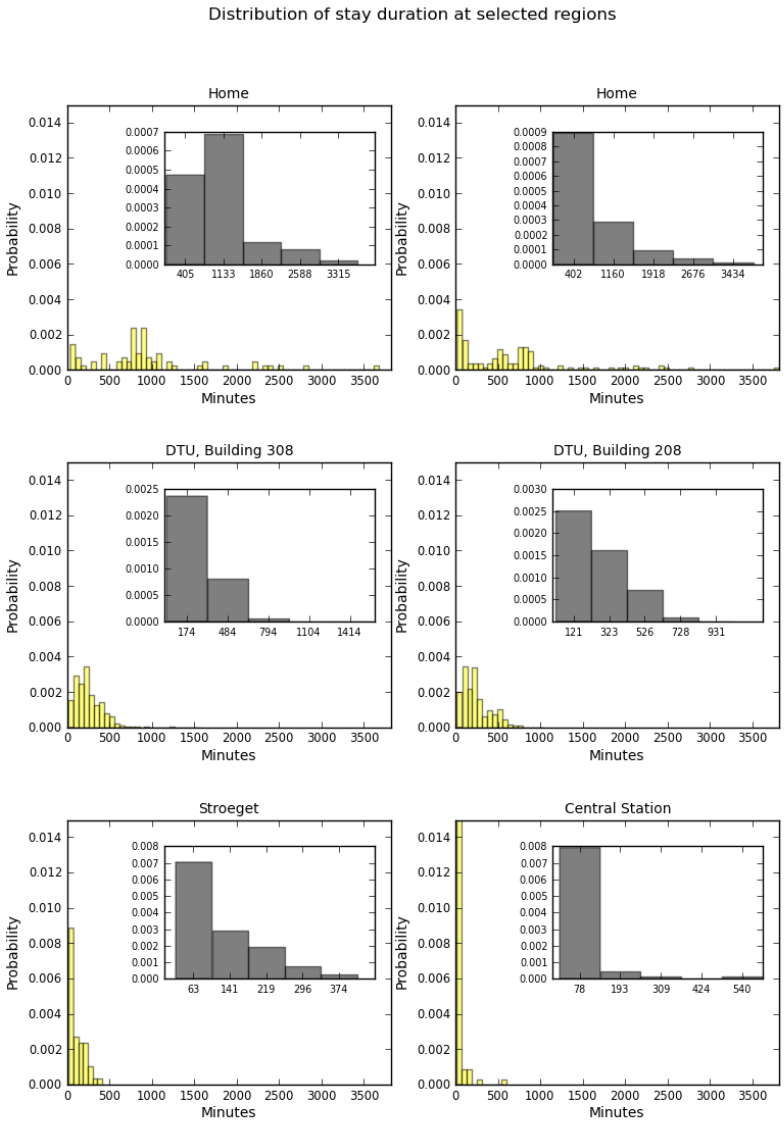


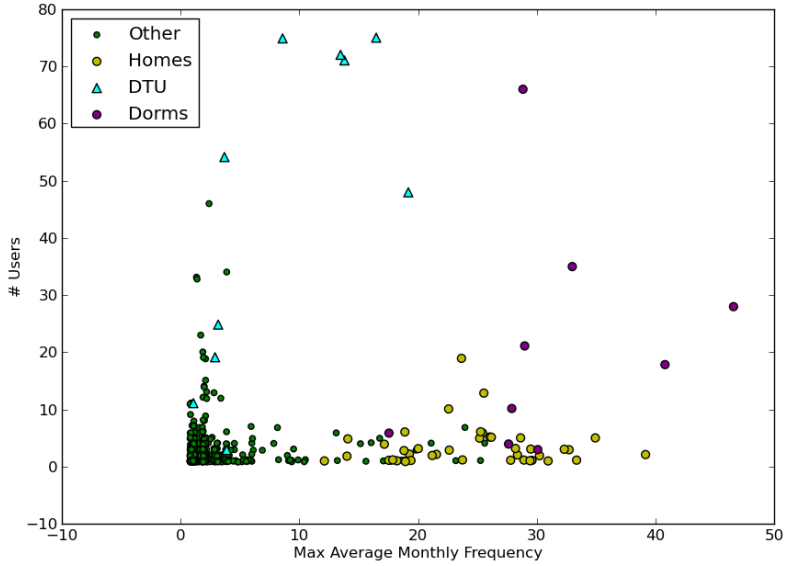
Figure 5.6: Stay duration distribution probabilities at selected stay regions.

Figure 5.7 shows the correlation between average monthly frequency of visits,

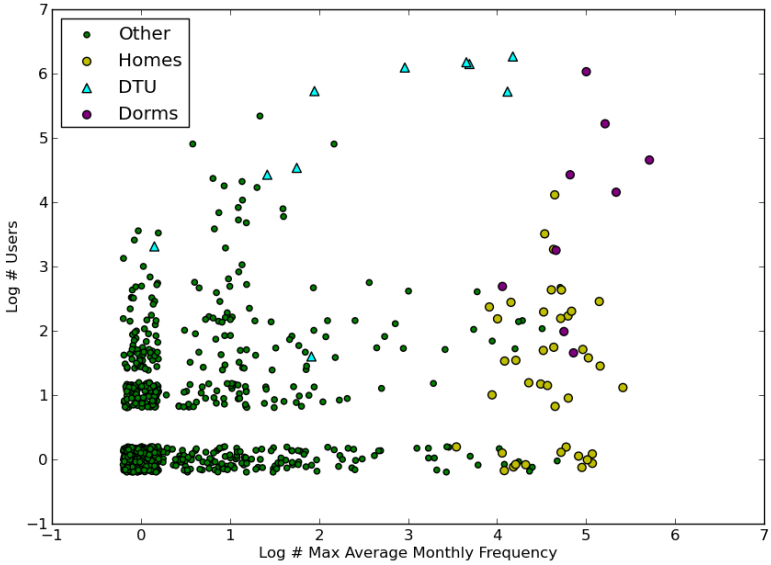
popularity and semantic category of a stay region.The average monthly frequency of visits to some stay region is calculated for every user independently, and a maximum value is set as parameter of a stay region. The maximum value is considered as a better representative, than median and mean, as they depend on the number of users who visited a particular stay region, which varies a lot between different stay region categories. Stay region popularity is determined by the number of users who visited the given stay region.

The figure shows that average monthly frequency is good in isolating stay regions containing users' homes, including dormitories. As expected, some dormitories have higher popularity than other home places. That is also the case for few homes, which are probably located at the same stay region as other places that are visited by a lot of people. The most popular places are four stay regions at DTU. It is expected that DTU stay regions are less frequently visited than homes, however that is not always the case, probably due to missing data affecting the average monthly frequency.

Figure 5.8 shows the correlation between average stay duration, popularity and semantic category of a region. Stay duration is not as good in separating home locations as average monthly frequency, nor as good in separating DTU regions as popularity.

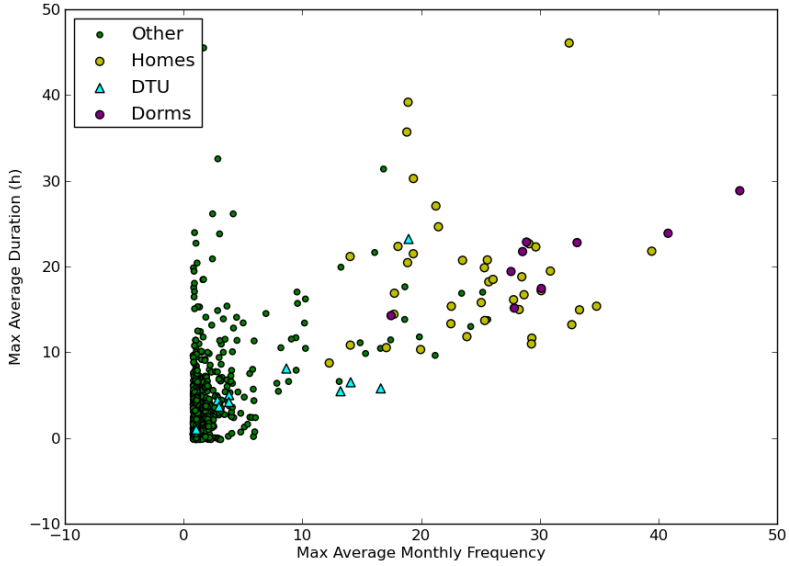


(a) Linear scale

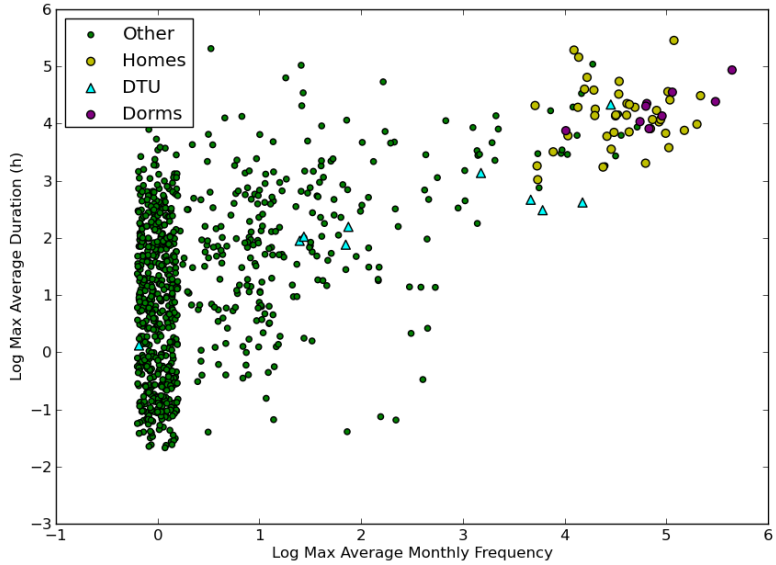


(b) Log scale.

Figure 5.7: Correlation between average monthly frequency of visits, popularity and semantic category of regions



(a) Linear scale.



(b) Log scale.

Figure 5.8: Correlation between average monthly frequency of visits, average stay duration and semantic category of regions

5.3 Changes of behavior over time

It is essential to consider changes of users' behavior over time while selecting the time duration of training data for next place prediction. Students' daily routine can be affected by changes of residence address, lectures schedule, changes of seasons etc. If the training set accounts for too long period of time, there is a chance that data collected at the beginning of the period does not represent user's current behavior patterns correctly.

Figure 5.9 illustrates how long time of recording it takes to detect most of the stay regions that user visits. Two plots at Figure 5.9 show cumulative distributions and average values of number of new stay regions detected per user for a week long period. The average number of discovered regions drops steeply during the first 6 weeks and, afterwards, it varies within the range from 0 to 1.5. Cumulative distributions for weeks from 6th of January 2012 and on show that no new regions were detected for more than 50% of users. In the weeks from 1st of April and from 6th of May 2013 more than 90% of users had less than 2 new stay regions detected.

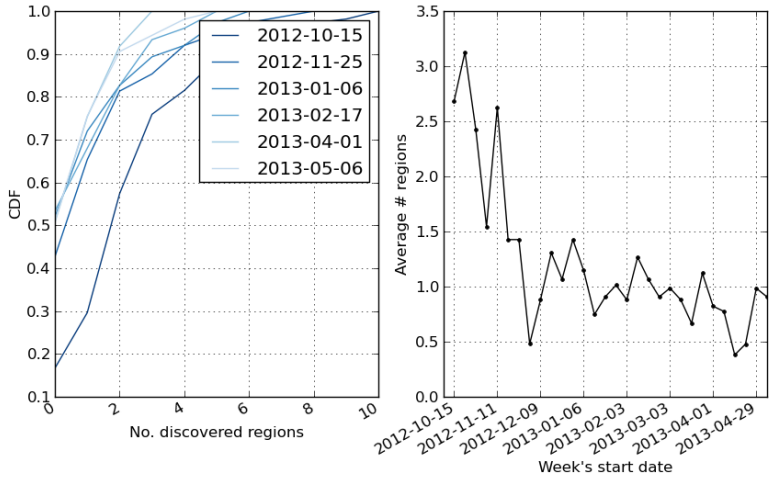


Figure 5.9: Number of stay regions detected per user during one week period

Figure 5.10 shows Jaccard similarity coefficient between a particular week and previous weeks with respect to stay regions visited during those weeks. The Jaccard similarity coefficient is calculated as the size of intersection divided

by the size of union of stay regions visited during particular two weeks. This measurement gives only coarse idea of similarity between two periods. as it does not consider frequency and time of the visits. However it clearly shows that stay regions change over time.

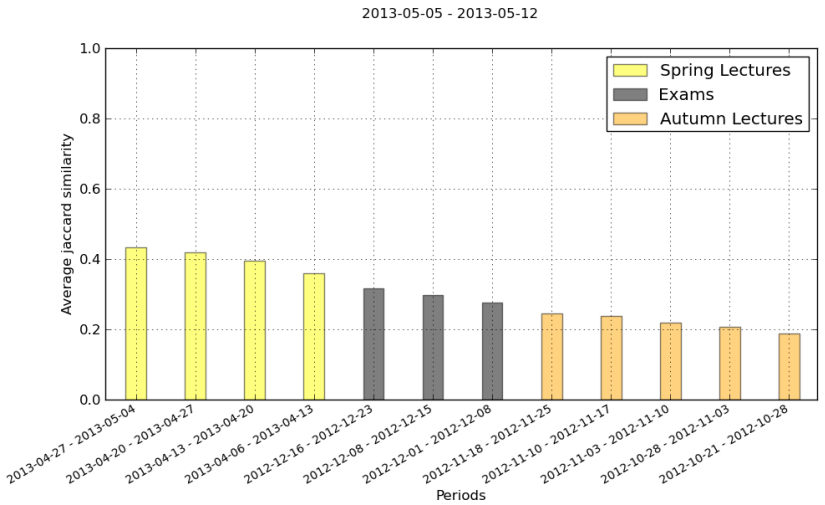


Figure 5.10: Similarity between the week from 5.5.2013 to 12.5.2013 and previous weeks

5.4 Co-location

Since all the experiment participants are first-year DTU students, it is safe to assume that they interact with each other or at least have similar schedule and spend time at the same places. The aim of this section is to provide an overview of when the experiment participants co-locate and how consistently that happens. If an individual regularly spends time at the same place as a group of other people, the data recorded for all of them together can be combined in the next place predictive model.

Figure 5.11 shows when and with how many other participants a particular participant co-locates with per 2 hour interval, during April 2013. It is considered that two participants co-locate during a particular time interval, if they are staying at the same stay region at any point of time during that time interval. The participant whose data is visualized at Figure 5.11 probably lives at the same place as few other participants, as even during the night he co-locates with

others. However, there are co-locations with more people on weekdays from 8-16 o'clock.

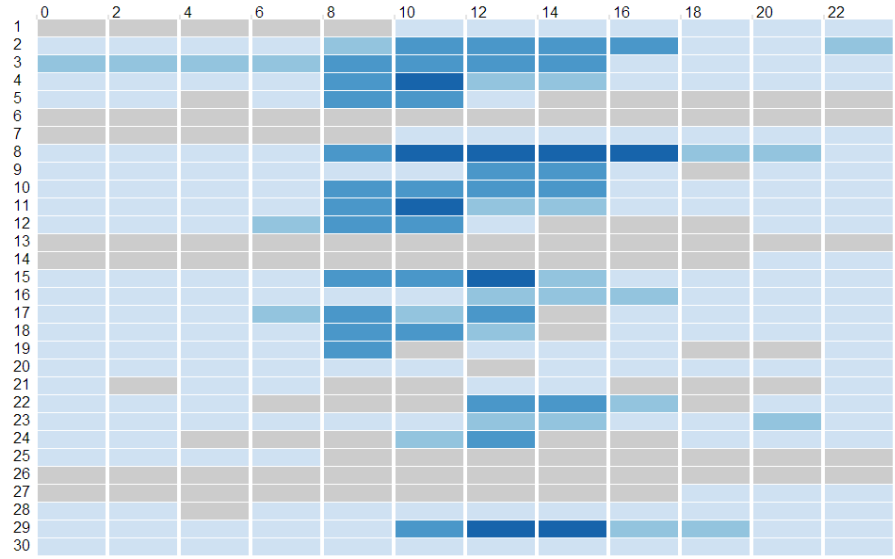


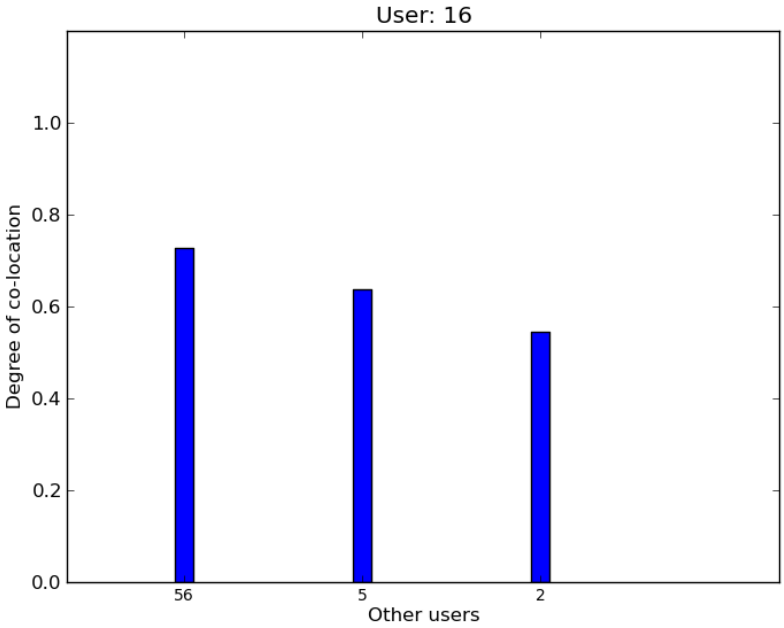
Figure 5.11: Co-locations for a particular participant during April 2013. Matrix cells represent the number of participants which have the same location as the observed participant, during a particular time interval of a particular day. Rows in the matrix represent days in a month and columns represent 2-hour intervals in a day.The number of participants is discretized into 4 bins using the following bin edges: 3, 10, 30.The color scheme includes shades of blue and a gray color, so that the darkest color corresponds with the bin containing the highest numbers and gray color corresponds to 0 co-locations.

During weekends , there are barely any co-locations, except Sunday afternoon (Weekends in April are 6^{th} and 7^{th}, 13^{th} and 14^{th}, 20^{th} and 21^{th}, and 27^{th} and 28^{th}.).

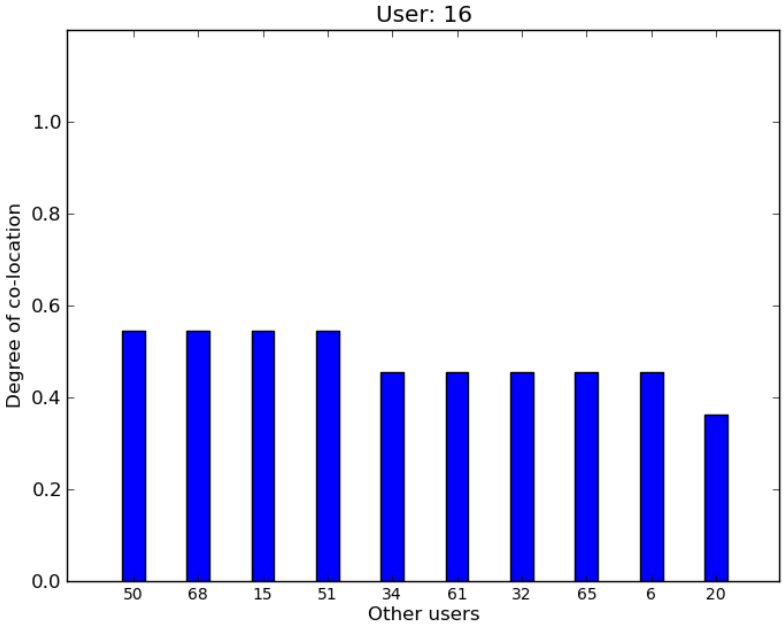
As shown at Figure 5.11, there is some consistency with respect to when the co-location with other participants occur. However, it is not visible with whom the observed participant co-locates, nor whether the co-locations with the same participants repeat with certain temporal consistency. Degree of co-location is introduced as a measurement for temporal consistency of co-location between 2 participants in certain time interval. Figures 5.12 and 5.13 show the degree of

co-location calculated based on data for all Fridays in April for the participant whose data was shown at 5.11. A day is divided into four 6-hour intervals and the degree of co-location is calculated for the participant in question and all other participants he meets in particular interval (The plots at 5.12 and 5.13 only show 10 participants with whom the participant in question has the highest degree of co-location). The degree of co-location during the 6-hour interval for two participants is calculated as the number of 2-hour intervals within the 6-hour interval for all Fridays in April when the participants are observed at the same location divided by the total number the intervals when the location of the first participant is known.

Figure 5.12a shows that in more then 50% of cases, a participant, marked as user 16, is at the same stay region as participants marked as 56, 5 and 2 on Fridays between midnight and 6 o'clock. According to this, and the degree of co-location values for the same interval on other days of the week , it seems that the participant is question is living at the same location as above mentioned participants. On Fridays between 6 and 12 o'clock, participant 16 shows high degree of-co location with multiple other participants, which probably means that he has morning classes at DTU on Fridays. Participant 16 does not show high degree of co-location with other participants in remaining 6-hour intervals on Fridays.

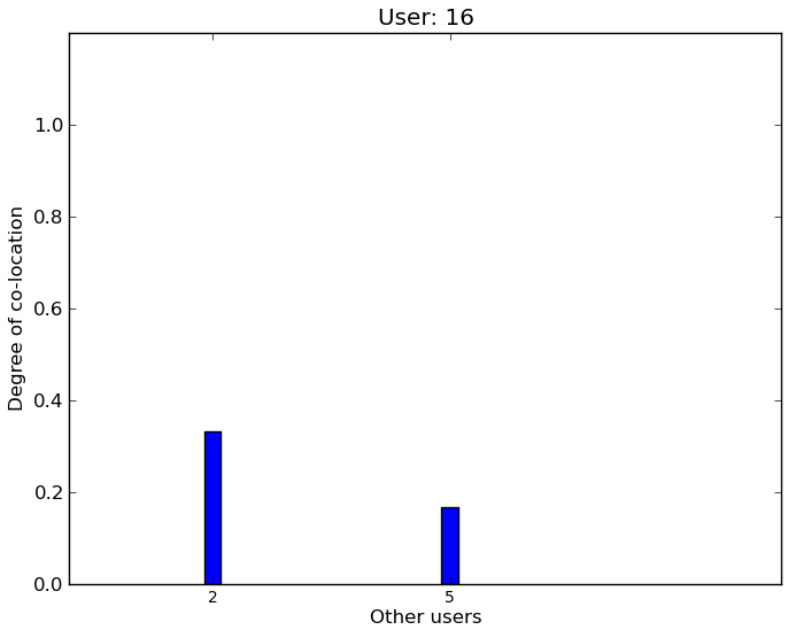


(a) 0-6 o'clock

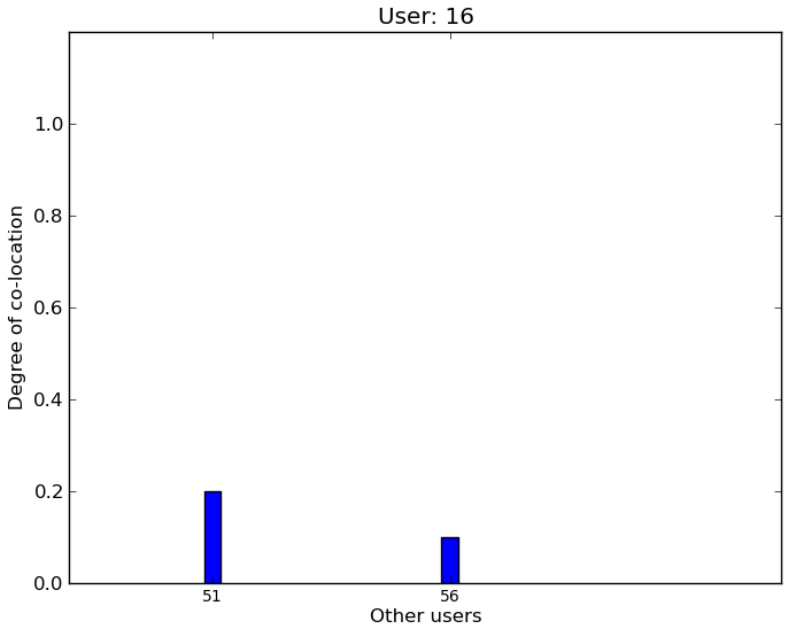


(b) 6-12 o'clock

Figure 5.12: Degree of co-location for Friday.



(a) 12-18 o'clock.



(b) 18-24h

Figure 5.13: Degree of co-location for Friday.

Chapter 6

Next place prediction

6.1 Conditional Contextual Models

A contextual conditional model estimates the probabilities that an experiment participant will visit one of previously visited destinations, given the current context. The current context is defined by a single or a combination of the following contextual variables: current location, hour, day of a week, weekend indicator, previous location, frequency and duration of visits, and popularity of current location. Accordingly, the model is equivalent to a conditional probability distribution where output variable is potential next place and input variables are given by a particular subset of contextual variables.

This model is originally proposed at [DGP12]. In this thesis, two new contextual variables are considered, which include popularity and previous location. The overview of variables is given in Table 6.1. Conditional probability distribution can be estimated only if both input and output variables are discrete. The variables are discretized using different bin edges comparing to [DGP12]. Bin edges for frequency, popularity and duration are decided based on the plots 5.7a and 5.8a at section 5.2 and by adjusting their values based on the resulting prediction performance.

The model which depends only on LOC variable is equivalent to the 1-order

Markov model, and the model depending only on PREV variable is similar to 2-order Markov model with fall-back, proposed at [SKJH03].

Name	Description
LOC	Current stay region ID
H	Hour of the day when a particular transition occurs. The variable is discretized into 12 levels, each containing a 2-hour interval.
D	Day of a week when a particular transition occurs. The variable ranges from 0 to 6, for days from Monday to Sunday.
W	Weekend indicator. A variable takes 0 or 1 values, which indicate whether a particular transition occurred on weekend.
DUR	Average duration of trusted visits to current stay region. The variable is discretized into 3 bins using 4 and 10 hours as bin edges .
FREQ	Average monthly frequency of visits to current stay region. The variable is discretized into 3 bins using 5 and 12 as bin edges.
POP	Popularity of current stay region expressed with number of users who visited it. The variable is discretized into 4 bins using 3, 10 and 50 as bin values.
PREV	IDs of previous and current stay regions. If the previous location is not available, the model falls back to the model containing only current location ID.

Table 6.1: Contextual Variables

Each trusted transition corresponds to a record in the conditional contextual model. A trusted transition contains a pair of visits, which occur one after another. The end time of the first visit is regarded as trusted transition time. Values of FREQ and DUR variables are calculated based on all user's visits and trusted visits, respectively, occurring before the the current trusted transition. Value of PREV variable is calculated based on the previous trusted transition. If the stay region of the first visit, of the current trusted transition matches the stay region of the second visit, of the previous trusted transition, then the stay region of the first visit, of the previous trusted transition, is considered as previous location, otherwise,value of PREV is considered unknown. Values of all other contextual variables are calculated based on information about the

first visit of the trusted transition, while the output variable is set to the stay region ID of the second visit.

The model can be formally expressed as follows. Let the following be the variables used in the model formula:

- i is trusted transition index in particular user's trusted transitions history.
- u is a particular user
- $x_k(u,i)$ is a vector consisting of values for particular contextual variables, calculated based on user's past mobility at the time of trusted transition i.
- $y(u,i)$ contains the ID of destination stay region of trusted transition i.
- $Y(u,i)$ is a set containing distinct IDs of destination stay regions of trusted transitions occurring before trusted transition i and a $\{NewPlace\}$ destination.
- α - regularization factor

Probability of potential next destination y for user u and trusted transition i is calculated as follows, having that the context given by $x_k(u,i)$ occurred in the past ($\sum_{j=1}^{i-1} 1[x_k(u,j) = x_k(u,i)] > 0$):

$$p_k(y|x_k(u,i)) = \frac{\sum_{j=1}^{i-1} 1[x_k(u,j) = x_k(u,i) \wedge y(u,j) = y] + \alpha}{\sum_{j=1}^{i-1} 1[x_k(u,j) = x_k(u,i)] + \alpha|Y(u,i)|}, \tag{6.1}$$

where $p_k(y|x_k(u,i))$ is an abbreviation for conditional probability $p_k(Y = y|X = x_k(u,i))$.

If the context did not occur before, the probability is calculated as by the formula:

$$p_k(y|x_k(u,i)) = \begin{cases} \frac{2\alpha}{|Y(u,i)|}, & y = NewPlace. \\ \frac{\alpha}{|Y(u,i)|}, & \text{otherwise.} \end{cases} \tag{6.2}$$

Let the vector $P_k(y|x_k(u,i)) = \{p_k(y|x_k(u,t)) \mid y \in Y(u,i)\}$ be the vector of estimated probabilities for each potential next destination for the trusted transition i of user u. Then the prediction is given by the formula:

$$y_{\text{predicted}}(u,i) = argmax_y P_k(y|x_k(u,i)) \tag{6.3}$$

If the most probable destination is equal to the current location, the next most probable destination is given as the next place prediction.

According to equations (6.2) and (6.3), if particular combination of values for contextual variables, given by the vector $x(u, i)$, does not exist in the training set, NewPlace will be predicted as the next destination. In this case, probabilities for all potential next destinations, including NewPlace, will be low. This is done in order to lower the impact of probabilities from the models which cannot give a prediction based on a particular context, to the combined Model. The combined model is explained in the next section.

Multiple conditional contextual models are tested for each user, by using the first half of particular user's trusted transitions for the training set and second half for the test set. In each step, a training set is updated with the trusted transition, which was just tested. If the result of next place prediction is NewPlace, the result is treated as correct if the actual next place was not visited in the past. Accuracy of particular model for particular user is calculated as the number of correct predictions divided by the total number of predictions. The average prediction accuracy for each considered model is provided in the Table 6.2.

Name	Acc.	Name	Acc.
LOC	0.454	FREQ + H	0.495
DUR	0.453	FREQ + H + W	0.488
FREQ	0.449	FREQ + H + D	0.455
H	0.495	FREQ + DUR	0.457
D	0.447	FREQ + DUR + H	0.491
W	0.444	FREQ + DUR + H + W	0.484
LOC + H	0.457	FREQ + DUR + H + D	0.449
D + H	0.478	FREQ + POP	0.448
W+ H	0.493	FREQ + POP + H	0.495
LOC + H + D	0.424	FREQ + POP + H + W	0.488
LOC + H + W	0.453	FREQ + POP + H +D	0.455
DUR + H	0.495	PREV	0.451
DUR + H + W	0.490	PREV + H	0.455
DUR + H + D	0.465		

Table 6.2: Average prediction accuracy per conditional contextual model

The models with the highest performance are models depending on H, DUR+H, FREQ+H and FREQ+POP+H. Therefore, the most important contextual variables for next place prediction are hour of the day and contextual variables determining the type of current location, but not the current location itself.

Additionally, a weekend indicator plays more important role than the day of a week variable. The reason for the above might be the lack of data samples, manifesting in the lack of occurrences satisfying a condition given by particular values for variables of higher granularity.

6.2 Combined Model

Do at al. [DGP12] proposed an ensemble method with the purpose of increasing prediction performance over the conditional contextual models. The ensemble method consists of learning weights for each individual model and combining weighted probabilities, given by individual models, into a single probabilistic model. They introduced the combined model to resolve two conflicting needs: a need to do more informed predictions, relying on multiple contextual variables, and a need to estimate the conditional probability distribution accurately, which requires having enough data samples satisfying the condition given by the contextual variables. If the context was given using all contextual variables, the model would have poor prediction performance due to lack of data to estimate the conditional probability distribution accurately. The proposed model is similar to Naive Bayes model, however Naive Bayes model combines conditional probability distributions on equal grounds, where each conditional probability distribution depends on a single variable. Naive Bayes predictor relies on an assumption that conditional variables are mutually independent, which is not the case with the proposed combined model.

Combined model provides a probability distribution over a set of potential next places, where each probability is calculated as a weighted combination of probabilities given by multiple conditional contextual models for a particular user and a particular transition. Probabilities for combined model are calculated by the following formula:

$$p(y|x(u,i)) = \frac{\prod_{k=1}^{K} p_k(y|x_k(u,i))^{w_k}}{Z(x(u,i))} \tag{6.4}$$

Symbols introduced by the formula are the following:

- K is the number of conditional contextual models considered. In this thesis, 27 contextual conditional models are considered (see Table 6.2).
- w_k weight of k^{th} conditional contextual model.

- $p_k(y|x_k(u,i))$ - conditional probability given by a particular conditional contextual model.
- $Z(x(u,i))$ - normalization constant. $Z(x(u,i)) = \sum_{y' \in Y(u,i)} \prod_{k=1}^{K} p_k(y'|x_k(u,i))^{w_k}$

Learning weights is set as an optimization problem with the goal to maximize the difference between probability of the actual next place and the probability of any other candidate for the next place, over the whole data set including all transitions for all users. Accordingly, for every user u and for every trusted transition i, the following in-equation should be valid:

$$\prod_{k=1}^{K} p_k(y_{\text{actual}}|x_k(u,i))^{w_k} > \prod_{k=1}^{K} p_k(y|x_k(u,i))^{w_k}, \ \forall y \in Y(u,i) \wedge y \neq y_{\text{actual}} \tag{6.5}$$

The optimization problem is solved using Stochastic Gradient Descent method [1]. Stochastic Gradient descent estimates parameter w by minimizing an objective function of the following form:

$$Q(w) = \sum_{i=1}^{n} Q_i(w), \tag{6.6}$$

where $Q_i(w)$ is a value of loss function of i^{th} data sample. Stochastic Gradient Descent iteratively minimizes given objective function by subtracting the value of a gradient of a loss function, multiplied by a small step size α, from the parameter w for each data sample:

$$w = w - \alpha \nabla Q_i(w), \ \forall i \in 1, ..., n \tag{6.7}$$

The objective function for the given optimization problem is formulated as follows. A natural logarithms are applied to both sides of the in-equation at (6.5), which results in the following in-equation:

$$\sum_{k=1}^{K} w_k \ln p_k(y_{\text{actual}}|x_k(u,i)) > \sum_{k=1}^{K} w_k \ln p_k(y|x_k(u,i)), \ \forall y \in Y(u,i) \wedge y \neq y_{\text{actual}} \tag{6.8}$$

If $W = \{w_k \mid k \in \{1, ..., K\}\}$ and $P_{u,i,y} = \{p_k(y|x_k(u,i)) \mid k \in \{1, ..., K\}\}$ are two vectors, then the in-equation (6.8) can be rewritten using the inner product of the two vectors:

$$\langle \ln P_{u,i,y_{\text{actual}}}, W \rangle > \langle \ln P_{u,i,y}, W \rangle \ , \ \forall y \in Y(u,i) \wedge y \neq y_{\text{actual}} \tag{6.9}$$

[1] http://en.wikipedia.org/wiki/Stochastic_gradient_descent

Accordingly, Do et al. [DGP12] define the objective function as follows:

$$Q(W) = \frac{\lambda}{2}||W||^2 + \sum_{u,i}\sum_{y} \underbrace{\max(0,\ 1 - \langle(\ln P_{u,i,y_{\text{actual}}} - \ln P_{u,i,y}),\ W\rangle)}_{\text{hinge loss}^2} \quad (6.10)$$

In this thesis, additional constraint is specified: $w_k \geq 0, \forall k \in 1, ..., K$, which I assume is implied at Do et al. work [DGP12]. Initially, all weights in the the weight vector W are set to 1. In every iteration of Stochastic Gradient Descent, weights are adjusted and the value of the objective function in recalculated. If the value of the objective function is greater in the current iteration than in the previous, the algorithm terminates.

The estimated weights are reused in the combined models for all users. Therefore, the training set for learning weights contains the first half of trusted transitions of every user. Furthermore, the training set was again divided on 2 halves. The first half is used to train the conditional contextual models, so that the second half can be used for estimating the probabilities for each conditional contextual model , which are used as constants in the objective function. Combined model is tested using the remaining have of trusted transitions per user, so same data is not used in the training set and the test set. Do at al. [DGP12] use leave one user out cross validation to learn weights; they estimate combination weights based on all trusted transitions of all users but one, and then test the combined model using the trusted transitions of the remaining user.

Figure 6.1 shows weights of each conditional contextual model, where weight is greater than zero.

The performance of the combined model is estimated using average accuracy, based on the data for 75 users. The average accuracy is equal to 0.552, which is higher than average accuracy of any individual conditional contextual model, where maximum average accuracy is 0.495.

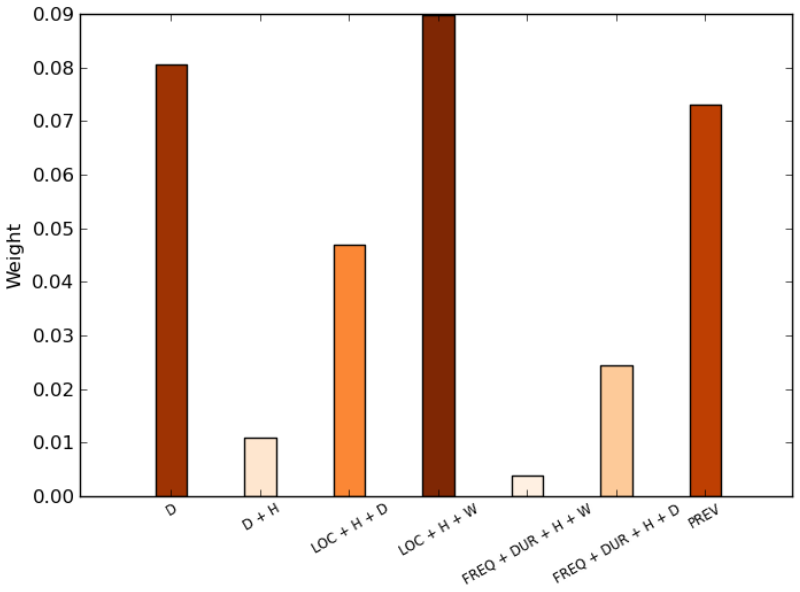


Figure 6.1: Weights for conditional contextual models

6.3 Baseline Models

As previously discussed, at Section 5.1, the time a user spends at different stay regions is not balanced. On average, users spend 83.5% in top 2 regions. This is also reflected on trusted transitions, where, on average, for more than 50% of trusted transitions, destination stay region is one of top 2 most frequently visited regions. Figure 6.2 shows the average fraction of trusted transitions to 10 most visited regions and the fraction of trusted transitions to all remaining stay regions.

Since the data set is imbalanced, the accuracy of predictive models is not a good estimation of the predictive performance, if it is not compared to the performance of baseline models which give the most frequent value of the output variable as a prediction. In this thesis, the performance of combined model and individual conditional contextual models is compared with the performance of three baseline models:

a) Most frequent - always gives the most frequently visited stay region as next place prediction.

b) Longest stay - predicts the stay region where a user spent the highest amount of time in in total.

c) Longest stay per hour of a week - predicts the stay region where a user spent most of the time per particular hour of particular day in a week.

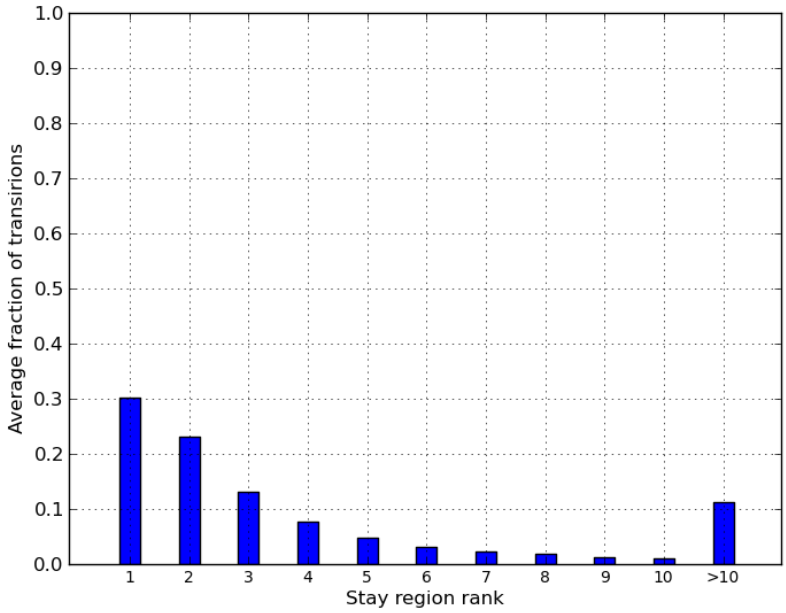


Figure 6.2: Average fraction of trusted transitions per region rank. The stay regions are ranked by frequency of visits in descending order.

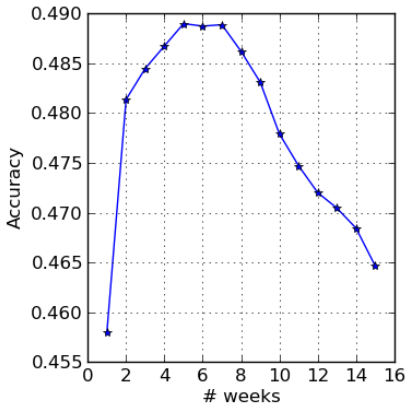
In all baseline models, most frequently visited region, or region with longest stay in total, is determined based on the past trusted transitions which occurred in a fixed-length period before the current trusted transition. This is done in order to acknowledge the changes occurring in user's mobility patterns over time. For example, most visited stay region during one month is not equal to the most visited region during last four months. Figures 6.3a, 6.3b and 6.3c show how the average accuracy of the baseline models changes depending on the number of weeks used for calculating most frequently visited region,

region with longest stay in total, and region with longest stay per hour of the week predictors, respectively. Accordingly, the length of training data is set to 5,6,13, for most frequent, longest stay and longest stay per hour of a week, respectively. All baseline models are tested using the second half of each user's trusted transitions, as it is done for other models. The average accuracy per baseline model is displayed at Table 6.3.

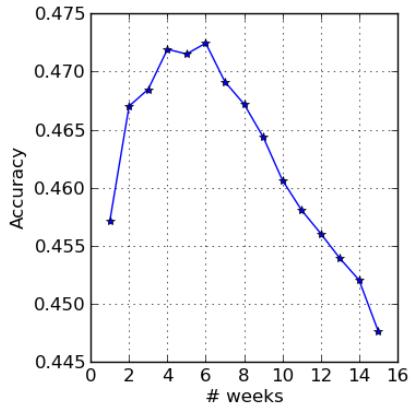
Name	Avg. Accuracy
Most Frequent	0.489
Longest Stay	0.481
Longest Stay per Hour of a Week	0.498

Table 6.3: Average prediction accuracy per conditional contextual model

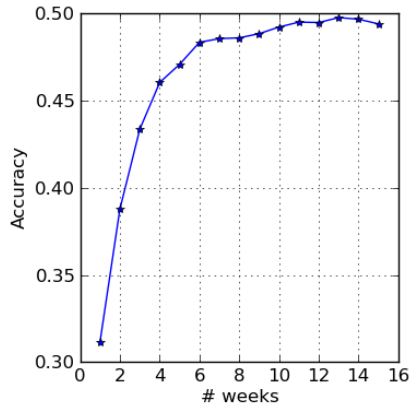
None of the baseline models outperforms the combined model. However, longest stay per hour of a week outperforms each conditional contextual model.



(a) Most Frequent.



(b) Longest Stay.



(c) Longest Stay per Hour of a Week

Figure 6.3: Correlation between accuracy and training set length for baseline models.

6.4 Improvements to next place prediction model

6.4.1 Academic calendar aware predictive model

This section provides a summery of the approaches considered towards encompassing the fact that human mobility changes over time into the conditional

contextual models. Do et al. proposed a weighted conditional contextual model where higher weight is assigned to more recent trusted transitions in the training set. In weighted conditional contextual model, the observations are firstly ordered in a reverse order of the time when they are occurring, so that observations occurring sooner to the current time have lower indices in the ordered list. Then, when estimating the conditional probability distribution, every observation is weighted using the inverse value of the observation's index in the ordered list. This approach was tested, but it did no bring any improvements to the prediction results.

Since it is known that the students had two lectures periods during the course of the experiment, I assumed that the predictive performance could be improved by predicting next place for trusted transitions which happened in certain lectures period, using the trusted transitions from the same period. This resulted in no improvement of prediction performance due to small number of data samples.

Name	Acc.	Name	Acc.
LOC	0.475	FREQ + H	0.511
DUR	0.475	FREQ + H + W	0.504
FREQ	0.473	FREQ + H + D	0.467
H	0.511	FREQ + DUR	0.479
D	0.472	FREQ + DUR + H	0.507
W	0.470	FREQ + DUR + H + W	0.497
LOC + H	0.469	FREQ + DUR + H + D	0.470
D + H	0.497	FREQ + POP	0.473
W+ H	0.506	FREQ + POP + H	0.511
LOC + H + D	0.433	FREQ + POP + H + W	0.504
LOC + H + W	0.465	FREQ + POP + H +D	0.467
DUR + H	0.510	PREV	0.472
DUR + H + W	0.504	PREV + H	0.466
DUR + H + D	0.481		
Combined model: 0.563			

Table 6.4: Average prediction accuracy per model

To improve this idea, I considered using all available past trusted transitions in the training set, provided that the trusted transitions occurring in the same period as the current trusted transitions had higher weight than trusted transitions occurring in other periods. According to the academic calendar at DTU, the whole experiment time is divided into three bins using dates of the end of first semester and the start of second semester as bin edges. The period in between the two semesters consists of winter exam period and winter break or

three weeks course period. The largest predictive performance improvement is achieved if the weight is set to 3. The model was tested using 2, 2.5, 3, 3.5 and 4 as weight values. The results at Table 6.4 show improvement in performance for each conditional contextual model and for the combined model. The conditional contextual models with the highest performance are H, DUR+H, FREQ+H, FREQ+POP+H, which is the same as for the flat models.

6.4.2 Co-location aware predictive model

It is expected that, as experiment goes forward and more data is collected, the prediction results will improve. However, by testing the predictive models from April until June, I noticed no improvement or only limited improvement of predictive performance.

An alternative way to increase the number of data samples is to join the data from other users into the predictive model build for a single user. This can only be done if there is certain similarity in users' mobility patterns. The similarity of users' mobility patterns have temporal characteristics, as shown in the Section 5.4: a particular user might meet a particular group of people only during certain part of a day, or he can stop meeting certain group of people he used to meet on daily basis. In this thesis, temporal similarity of mobility patterns of every two users is measured by the degree of co-location per 6-hour interval of a particular day of a week during 30 days period.

For every user's trusted transition for which the next place prediction is done, conditional probability distribution is calculated based on previous observations for that particular user and previous observations of all other users who co-located with the particular user during the day of a week of the trusted transition in any 6-hour interval of that particular day of a week. Degree of co-location between user and other users is calculated for 30 days period, and only the observations of other users' mobility during last 30 days period are included, provided that they occurred within a particular 6-hour interval, where degree of co-location between the two users was higher than 0.5.

The results of academic calendar and co-location aware conditional contextual models and the combined model are given at the Table 6.5. The results of the conditional contextual models which depend on hour of a day show slight increase in performance over the performance of only academic calendar aware models. For example, average accuracy for co-location and academic calendar aware models for H and H+W increased from 0.511 to 0.527, and from 0.506 to 0.521, respectively. However, I noticed slight fall in performance of the models depending on a day of a week, which might be due to the way how data from

other users was sampled. In total, the performance of the combined model is 0.558, which is lower than the performance of only academic calendar aware combined model.

Name	Acc.	Name	Acc.
LOC	0.474	FREQ + H	0.517
DUR	0.482	FREQ + H + W	0.509
FREQ	0.474	FREQ + H + D	0.464
H	0.527	FREQ + DUR	0.483
D	0.461	FREQ + DUR + H	0.509
W	0.473	FREQ + DUR + H + W	0.500
LOC + H	0.472	FREQ + DUR + H + D	0.460
D + H	0.482	FREQ + POP	0.474
W+ H	0.521	FREQ + POP + H	0.517
LOC + H + D	0.423	FREQ + POP + H + W	0.509
LOC + H + W	0.466	FREQ + POP + H +D	0.464
DUR + H	0.519	PREV	0.473
DUR + H + W	0.514	PREV + H	0.467
DUR + H + D	0.478		
Combined model: 0.558			

Table 6.5: Average prediction accuracy per model

6.5 Analysis of next place prediction results

The best performing model is academic calendar aware combined model whose average accuracy is 0.563. The distribution of accuracy per user is shown at Figure 6.4. The accuracy per user varies from 0.35 to 0.82. In order to better understand what influences on how predictable some user is, I investigated the correlation between accuracy and the number of detected regions per user (Figure 6.7) and I compared the accuracy between best performing conditional contextual models, baseline models and combined model for each user (Figure 6.9). By viewing the plots such as the one at Figure 6.9 for every user, I noticed that, in most cases, the performance of all models is in tight relation with how much time a particular user spends at the most visited stay region. This relation can be observed at Figure **??** showing the comparison between accuracy of most frequent and combined model for every user.

Another measurement of model performance is average distance between the actual and predicted stay regions. The distances probability distribution (see Figure 6.6) is left skewed with over 60% of distances bellow 1 km.

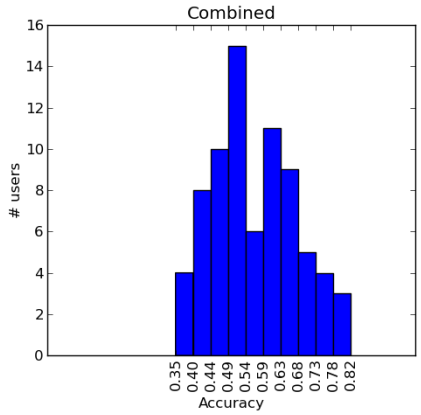


Figure 6.4: Distribution of accuracy per user for academic calendar aware combined model

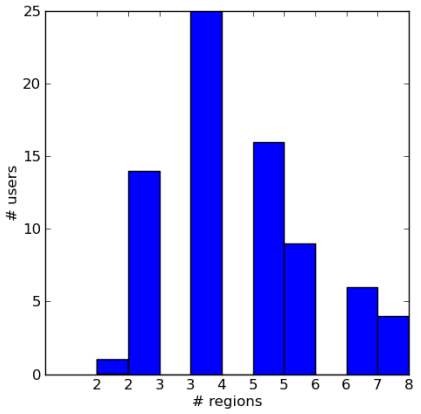


Figure 6.5: Distribution of number of stay region visited more than 5 times a month by one user.

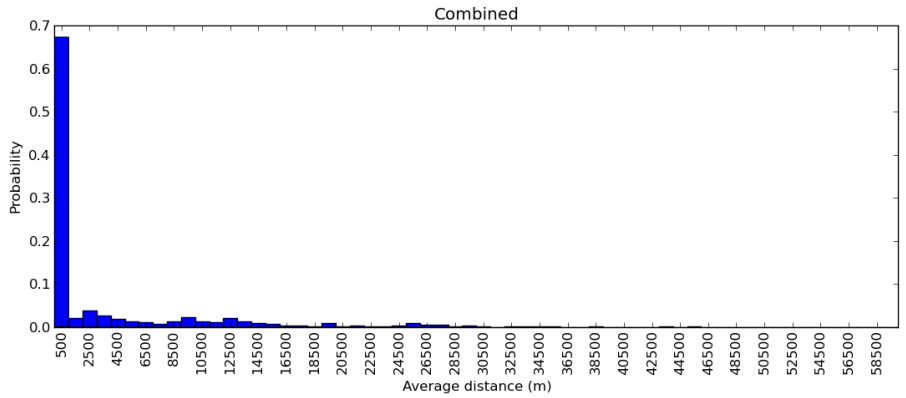


Figure 6.6: Probability distribution of distances between actual and predicted place for all trusted transitions.

Other picks of the probability distribution, at 2.5, 9.5 and 12.5 km might indicate the geographical distribution of stay regions, users visits. Average distance error for academic calendar aware model is equal to 3.83 km.

The same prediction method shows better results, when applied to another data set at [DGP12]. The average accuracy of the conditional contextual models at[DGP12] ranges from 0.392, for weighted conditional contextual model

depending on day of a week, to 0.604 for weighted conditional contextual model depending on current location and hour of a day. As a consequence, their combined model also has higher average accuracy, equal to 0.64.

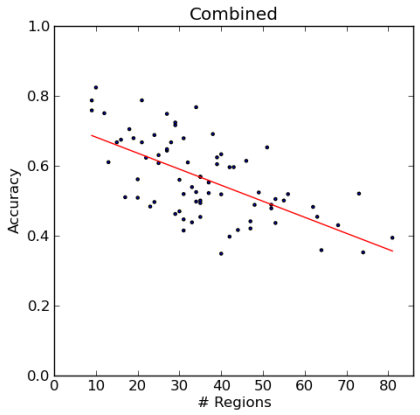


Figure 6.7: Correlation between the number of detected stay regions and accuracy of academic calendar aware combined model per user.

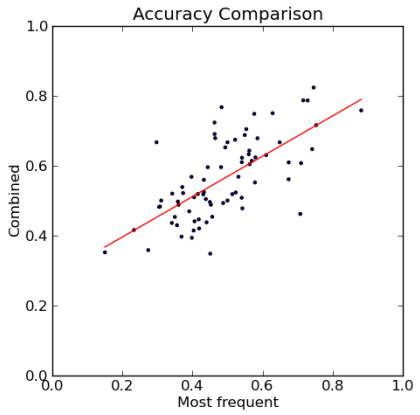


Figure 6.8: Correlation between the accuracy of most frequent and academic calendar aware combined model per user.

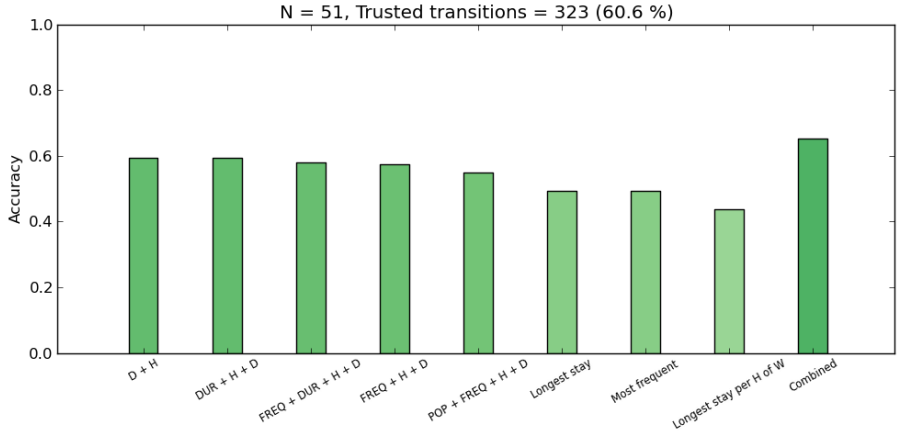


Figure 6.9: Accuracy of 5 best performing conditional contextual models, baseline models and combined model for a particular user.

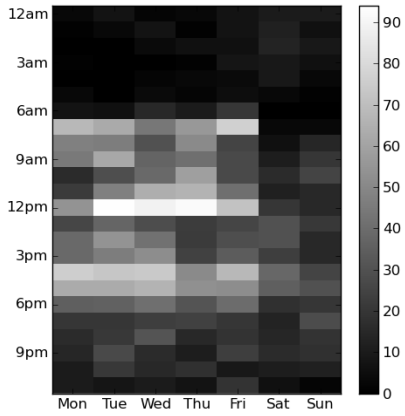
Their dataset contains data for 17 months and 153 users, consisting of students, professionals, few retired people and housewives, while data set used in this thesis accounts for 7,5 months and 75 users, all of which are students. They use the same plot as 6.2, to show the number of transitions per stay region, showing that almost 70% of trusted transitions has one of the two most frequent stay regions as destination. In this data set, two most frequent regions are destination for something above 50% of trusted transitions. This might mean that participants of this experiment are more mobile. However, I cannot be certain about such conclusion due to a big difference in the time lengths of the two data sets.

One of the problems regarding next place predicting is the lack of data samples to accurately estimate conditional property distributions. In order to simulate the situation when that is not longer an issue, I trained and tested the predictive models using only transitions between stay regions where monthly frequency of visits is greater than 5. Figure 6.5 shows the distribution of the number of stay regions per user where monthly frequency of visits is greater than 5. The average accuracy of the calendar aware combined model in such setting is 0.737. The average accuracies of baseline models are 0.630, 0.610 and 0.606 for the most frequent, longest stay and longest stay per hour of a week models, respectively.

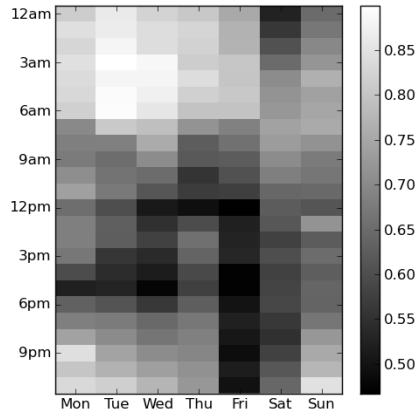
I assumed that the prediction performance would improve with better data quality. To verify that, I tested the prediction models only for 44 users for whom more than 60% of transitions were trusted. However, this resulted in lower average accuracy of the academic calendar aware combined model, 0.548, comparing to the following average accuracies 0.489, 0.481 and 0.498 of the most frequent, longest stay and longest stay per hour of a week baseline models.

Figures 6.10a and 6.10b show particular details about users' weekly mobility patterns. Figure 6.10a shows number of transitions per hour of a week, and Figure 6.10b shows regularity per hour of a week.In the context of human mobility predictability, regularity is a term previously used at [SQBB10], and shows the probability that user will be at the most probable place for particular hour of a week. Both visualizations are able to show DTU students daily patterns. Number of transitions is low by night and increases starting from 7^{am} on a weekday, then it picks at noon, during the lunch time, and then again at the afternoon, when students leave the university. It appears that there is a low number of trusted transitions after school and during weekend. However, this might be misleading because the number of transitions is calculated based on mobility of all users, and users' mobility patterns are very similar during school hours, but not in the afternoon. Similarly, the regularity is highest during the night and lowest during the lunch time and the time when lectures end. Furthermore, regularity image shows that users are least regular during Friday afternoon. Do et al. [DGP12] compared number of transitions and accuracy per hour of a

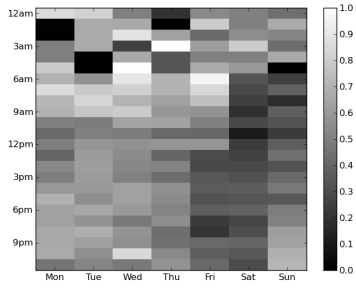
week. I tried to do the same, however it is impossible to see daily patterns at the accuracy image due to low number of transitions at certain hours. Instead, the accuracy image only shows that the accuracy is lower during weekend and Friday evening. Figure 6.10d shows that the average accuracy is higher at the beginning of the week and it achieves its lowest value on Saturday. This might be correlated to how missing data is distributed, which was shown at Figure 4.4 at Section 4.1.1.



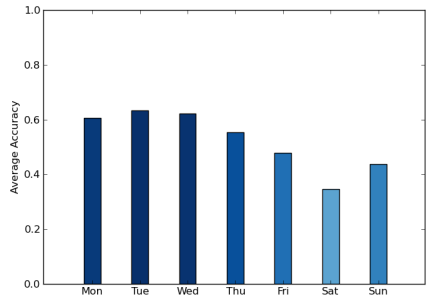
(a) Number of transitions per hour of a week.



(b) Regularity per hour of a week.



(c) Accuracy per hour of a week.



(d) Accuracy per day of a week.

Figure 6.10: Users' weekly mobility patterns.

Chapter 7

Conclusion

The work presented in this thesis includes thorough analysis of human mobility patterns with the goal to predict next place based on previous mobility. At the first stage, data was prepared for data mining, which consisted of handling of a large amount of missing data and turning raw GPS data to final set of stay regions. At the second stage, various data visualizations are done to reveal important characteristics of human mobility patterns, such as unbalanced distribution of time spent at different stay regions, characteristics of stay region categories, changes over time and existence of patterns in when, and with whom users co-locate. Finally, predictive modeling is implemented by adopting the framework proposed at [DGP12] to this data set. The framework consists of joining multiple conditional probability distributions, capturing various mobility patterns, in a single combined model, with the goal to maximize predictive performance. A conditional probability distribution depends on a subset of contextual variables including current location, hour of a day, day of a week, weekend indicator, duration and frequency of visits, previous and current location combined, and popularity of current location. The best performing conditional contextual model is the one depending only on hour of a day. The framework could be further expended by considering other relevant information such as weather history. Currently, the conditional contextual models do not depend on any characteristics of potential next destinations, which could be included to the models by assigning weights to model records, depending on characteristics of candidate destinations, such as general popularity of certain place, the

distance between current place and the candidate place, number of friends who are already there etc.

I introduced two alterations to the predictive framework: academic calendar aware models and co-location aware models. The academic calendar aware models give more relevance to the mobility patterns which occurred in the same period of a year as the current visit. With this approach, the predictive performance of all contextual conditional models is improved, which results in improved performance of the combined model as well. Co-location aware models join data of other users to particular user's predictive model based on when and how often a particular pair of users co-locates. Co-location aware models bring improvement to certain contextual conditional models which depend on hour of a day, however they do not bring improvement to the combined model. This might be due to the way data of other users is sampled, which requires further investigation.

The main reason of rather low accuracy of the predictive models is lack of data samples eg. users visit certain places seldomly or change mobility patterns, so it is hard to learn under which circumstances visits to those places occur. In order to partially overcome this problem, predicting of the semantic category of the next place could be considered, rather than predicting the exact next place. This could still have practical usage in advertising eg. if it is known that someone is going shopping, discounts can be offered to that person. Another option would be to reformulate the prediction task into predicting whether a user will visit a particular place among the most important places, or some other place. This is not considered in this thesis, because as seen at various visualizations, users most commonly have one or two most important places, and all other places have much lower relevance. I assumed that predicting transitions between home and work is not challenging, as it could be achieved by simple visualizations of someones schedule.

Bibliography

[AN12] Neal Lathia Cecilia Mascolo Anastasios Noulas, Salvatore Scellato. Mining user mobility features for next place prediction in location-based services. In *IEEE Internationcal Conference on Data Mining (ICDM 2012)*, 2012.

[CML11] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.

[Cut13] Andrea Cuttone. Sensiblejournal: A mobile personal informatics system for visualizing mobility and social interactions. 2013.

[DGP12] Trinh Minh Tri Do and Daniel Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 163–172. ACM, 2012.

[EP06] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[EP09] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[GKdPC12] S. Gambs, M.O. Killijian, and M.N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[MGP10] Raul Montoliu and Daniel Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[NS12] Tommy Nguyen and Boleslaw K. Szymanski. Using location-based social networks to validate human mobility and relationships models. *CoRR*, abs/1208.3653, 2012.

[SDK+06] Libo Song, Udayan Deshpande, Ulas C. Kozat, David Kotz, and Ravi Jain. Predictability of wlan mobility and its effects on bandwidth provisioning. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain*. IEEE, 2006.

[SK12] Adam Sadilek and John Krumm. Far out: predicting long-term human mobility. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 814–820, 2012.

[SKJH03] Libo Song, David Kotz, Ravi Jain, and Xiaoning He. Evaluating location predictors with extensive wi-fi mobility data. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7:64–65, October 2003.

[SMM+11] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Proceedings of the 9th international conference on Pervasive computing*, pages 152–169, 2011.

[SQBB10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327:1018–1021, 2010.

[YLWT11] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.

[ZFL+04] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. In *Geographic Information Systems: Proceedings of the 12 th annual ACM international workshop on Geographic information systems*, volume 12, pages 266–273, 2004.

[ZZM+11] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Trans. Web*, feb 2011.

[ZZXY10] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038, 2010.