Large Scale Analysis of Mobile Trajectories

- Survey and Analysis Stage -

Omar Iltaf
Supervised by Prof. Fabio Ciravegna
COM3610 Dissertation Project
3rd December 2018

This report is submitted in partial fulfilment of the requirement for the degree of BSc Computer Science by Omar Iltaf.

Declaration

"All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole."

Omar Iltaf

Abstract

Through the continuous recording of user movement data (e.g. Google Fit, Apple Healthkit), vast amounts of spatial trajectory data is available. This data holds valuable information allowing for insights into a users lifestyle, commitments and habits. Conducting an analysis on such data will help to understand what individual users prioritise in their daily lives and to compare this to a much larger group of users.

This project aims to use relevant and practised techniques from the area of trajectory data mining to be able to analyse large amounts of trajectory data received from users monitoring their daily routines. Trajectories most relevant to a users lifestyle will need to be identified and extracted from all other data. These relevant trajectories can then be used to determine specific insights into the users life. For example, their interests or activities they like to do. Lastly, a model to classify all users will be produced.

So far, extensive research has been performed into the field of trajectory data mining and the areas most relevant to this project will be discussed below. The requirements for this project have also been identified and stated below, along with an analysis of the best methods to use in solving the problems mentioned..

Contents

List of Figures

Chapter 1: Introduction

The mass-availability of sensors within mobile devices enables the continuous quantification of user movement (e.g. Google Fit, Apple Healthkit) as well as the analysis of user behaviour. This analysis on behaviour can be used to support users in the management of their everyday life. For example, the AI-powered virtual assistant: Google Assistant, tells users how long it will take to get home when the phone is accessed at a time when the user generally goes home from work.

The aim of this project is to create a tool able to analyse the spatial trajectory data received from a large set of users, then to extract from the data relevant insights on the routines of particular users and the user base as a whole. One such insight could be the places a user regularly visits, which may correspond to an interest/hobby. The final outcome will be a model to classify all users based on various factors such as their interests, location, or the type of activities they like to do.

The data for this project will be obtained from a mobile app monitoring the location data of its user as they go about their daily routine. In addition to the trajectory data indicating where the user has travelled, the app also provides detailed information such as: the mode of transport used (walking, bicycle, vehicle); the length of time spent in transit; the time of day when the journey began and ended; and the total distance travelled. This additional data allows for a detailed analysis to be conducted on a users lifestyle. For example one particular user may wake up and go running for an hour every morning at 6am, indicating they have a strong interest in fitness. Another user may display their interests by the type of places they frequent and the length of time spent there, for example an interest in films by a user who often goes to the cinema. These interests can be identified through the analysis of the trajectory data and will be used in the classification process. Data from thousands of users is being received, allowing for a large-scale analysis to be conducted.

The remaining chapters can be detailed as follows: Chapter 2 discusses the research and literature relevant to this project; Chapter 3 details the objectives of the project and how it will be evaluated; and Chapter 4 concludes with a summary of all progress made to date and a plan of work for the remainder of the project.

Chapter 2: Literature Survey

This chapter gathers together the research done in the area of Trajectory Data Mining, which relates closely to this project. It identifies and considers the possible techniques, methods and technologies to be used in performing the various tasks involved with the project. This includes: preparing the data before analysing it, taking into account how it was recorded in the first place; thinking about how best to store the data to aid in the analysis of it; identifying relevant trajectories to a users lifestyle; and dealing with anomalous trajectories.

2.1: Data Preprocessing

GPS tracked movement data is usually not completely reliable or precise, and may have significant noise affecting the data. Due to this, before it can be analysed, the data must first be preprocessed to correct or remove erroneous data and perhaps add additional information to allow the trajectories to possess more meaningful characteristics.

2.1.1: Noise Filtering

Due to the nature of how spatial trajectory data is usually recorded, where data is tracked through an app on a users mobile device, it is subject to produce noisy results. This could occur when travelling through areas which negatively impact positioning signals. Existing methods for noise filtering can be placed within three categories:

Mean/Median Filter
The Mean/Median filter approach involves finding an estimate of the true value for a given point by calculating the mean of all its previous points up to a given limit. The Mean filter proves a good choice when handling individual noise points. However, when multiple, consecutive noise points are preceding a given point, the calculated mean value for said point will have a larger error between it and the points true value. Mean/Median filters tend not to be the best choice when the distance between two consecutive points is greater than several hundred metres. (Zheng 2015)

Kalman and Particle Filters
The Kalman filter takes measurements from the noisy data over a period of time, and then produces variables more accurate than those based on measurements taken at a single point in time. This is done by estimating a joint probability distribution over the produced variables for each time frame. When assuming linear models and Gaussian noise, the Kalman filter efficiency improves. These assumptions are not used in the particle filter however, resulting in a more general and less-efficient algorithm.

Particle filters consist of four main steps. Firstly, the initialisation step generates a set number of particles from the initial distribution. Then a dynamic model is used to probabilistically simulate how the particles change over one time step. Thirdly, importance weights are calculated for all particles, and are normalised. A greater importance weight would indicate that the particle is better better supported by the measurement. The final step involves selecting a new set of particles which are proportional to the normalised importance weights. Then a weight sum is computed. (Zheng 2015)

Both the measurement of noise and the dynamics of a trajectory are modelled by Kalman and particle filters. Though it should be noted that their results highly depend on the reliable of the initial location. If the initial location of the trajectory is heavily affected by noise, the effectiveness of both filters decreases significantly. (Zheng 2015)

Heuristics-Based Outlier Detection

Hassan et al. (2011) details a heuristic approach to outlier detection in sensor node measurements inn which a combination of the the cluster-based, nearest neighbour algorithm and the measurements themselves are used. Their approach begins by specifying a width “w” for the clusters, after which clustering is performed on all sets of temperature readings produced by the sensors. Once the clusters have been obtained, the cluster with the least amount of values will be output as the outlier cluster. Finally, a confidence measure for each sensor is calculated using the knowledge of which sensors produced the readings present in the outlier cluster.

This method of heuristics-based outlier detection could also be used with trajectories, though some the width value should be chosen carefully as points along a trajectory may not be of equal distance apart. The general idea would be the same where each trajectory point could be checked to see if the number of its neighbours within a certain width is smaller than the proportion of points in the whole trajectory. (Zheng 2015)

2.1.2: Stay Point Detection

Li et al. (2008) describes a stay point as a geographic region where the user has stopped for some time. When compared to a raw GPS point, a stay point carries some semantic meaning, such as a users home, workplace, or restaurant they like to frequent.

Figure 2.1 below depicts two categories of stay points. Stay point 1 occurs at P3 where the user has remained stationary for a time period greater than an stay point identifying threshold. This could occur when entering a building and losing satellite signal for period of time before going back outside and continuing the trajectory. The second of category of stay point occurs when a user will move around a certain spatial region for some time before continuing the trajectory.
Whilst moving around in the set area, multiple GPS points are recorded (P5, P6, P7, P8). These points can be used to calculate a mean value for the coordinates of the spatial region that the user is moving around. An example of this could be when visiting places like a park or a shopping centre, where the place spans a large area.

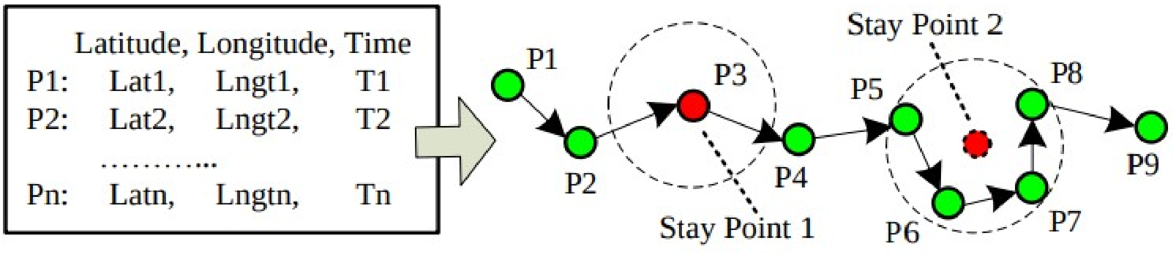

Figure 2.1 GPS Logs and Stay Points (Li et al. 2008)

There exists an algorithm to detect stay points present in a trajectory. The algorithm works by iteratively seeking the spatial region in which a user stays for a period over a threshold. For example, if a user spends more than 30 minutes at a place within a range of 200 metres, the place will be recorded as a stay point. It should be noted that for each extracted stay point, the algorithm also extracts temporal information, such as the arrival and leaving time at the stay point. (Yang 2009)

2.1.3: Trajectory Segmentation

In order to aid in the later processes of trajectory clustering and classification, a trajectory must be broken down into individual segments. The segmentation process will reduce the computational complexity of analysing the trajectories, as well as allowing to mine for deeper knowledge within the overall trajectory. (Zheng 2015)

Yoon and Shahabi (2008) describe three trajectory segmentation algorithms. The first is a top-down approach which takes an unsegmented trajectory as input and selects a characteristic timestamp. This will be the largest time-referenced distance between the observed location sampled at this time and the estimated time-referenced location. The characteristic timestamp is split into two subsequence for which the algorithm is recursively run until no observed location deviates from its estimated time referenced position by more than the given distance threshold. The second algorithm is a bottom-up approach which begins with the finest possible segments of a trajectory and attempts to merge two adjacent segments with minimum merge cost. The third is the sliding window algorithm which fixed the first timestamp of a trajectory as the first characteristic timestamp and attempts to place the next one as far along as possible whilst it still remains spatially and temporally homogeneous. Otherwise, the segment is separated from the trajectory and the process is repeated until the end of the trajectory is reached.

2.1.4: Map Matching

Map matching algorithms make use of inputs generated from positioning technologies and supplement them with data from high resolution spatial road network maps providing improved positioning output. The general purpose of of map matching algorithm is to identify the correct road segment on which a user has travelled, and to determine their location along the segment. (Quddus et al. 2007)

Lou et al. (2009) detail three approaches to matching GPS observations on to a digital map. They can be placed into three categories: local/incremental method; global method; and statistical method. Greenfield (2002) implemented an incremental method which uses distance similarity and orientation similarity to evaluate the candidate edges. Adding together the individual scores would produce the combined similarity measure. The global methods in map matching involve matching an entire trajectory to a road network. Yin and Wolfson (2004) developed a snapping method which finds the minimum weight path based on edit distance. There are other methods which instead use Fréchet distance. This distance measure is suitable for trajectory matching as it takes the continuity of curves into account. The third and final method for map matching uses statistical models. Pink and Hummel (2008) implemented such a method based on a Bayesian classifier and incorporates a Hidden Markov Model to model topological constraints on a road network. Lou et al. (2009) notes that statistical approaches are most effective in handling GPS measurement errors.

2.2: Data Management

The process of trajectory pattern mining can be very time consuming when dealing with large amounts of data as different samples of the trajectories and parts of it need to be accessed many times. As a way to combat this problem, effective data management techniques can be used to obtain the parts of the trajectory we need quickly. (Zheng 2015)

2.2.1: Data Model

Chakka et al. (2003) identify that trajectory data is continuously changing between any two successive location updates. Since most existing models for data representation are static in nature, this becomes a problem in representing the users location at all time. A commonly used model represents trajectory data as a straight line between two GPS point readings. A user who is moving has their GPS coordinates sampled at discrete times and a series of straight lines joint these points represents movement.

2.2.2: Query Types

Queries on moving data can be placed into two categories: those querying about future positions and those querying historical positions of moving objects. This project is most concerned with historical based queries so we can identify where users visit most often and patterns in their past trajectories. Pfoser et al. (2000) further divides historical queries into coordinate-based and trajectory-based. Coordinate based queries include queries within a given time interval or point in time, and nearest neighbour queries.

2.3: Trajectory Pattern Mining

There exists four major categories of patterns that can be found within either an individual trajectory or a group of trajectories. These include: moving together patters, trajectory clustering, sequential patterns, and periodic patters. (Zheng 2015)

2.3.1: Moving Together Patterns

Moving Together Patterns are described by Zheng (2015) as groups of objects that move together for a certain time period. Such patterns can have varying group shape/density, different numbers of objects within the group, and varying pattern duration. There are different types of these groups of objects, which are fairly intuitive, such as flocks, convoys, swarms, travelling companions and gatherings. Flocks can be thought of as a group of objects travelling together within a disc-like shape, the size of which can be specified and shape is maintained for a set number of consecutive timestamps. In order to avoid size and shape restrictions, convoys can be used to capture the generic pattern of any shape by using density-based clustering over a set number of consecutive time points. This is in place of using the disc-like shape of flocks. Li et al. (2010) introduced a more general trajectory pattern called swarms. Swarms allow for object clusters lasting for a set number of timestamps that does not necessarily have to be consecutive. Tang et al. (2012) discuss a more memory efficient pattern: the travelling companion, which uses a data structure called the travelling buddy to continuously find convoys/swarms from trajectories. The trajectories would be streamed into the system rather than being entirely loaded into memory as is the case for convoys and swarms. Lastly, the gathering pattern covers those events when objects (users) join together and then leave frequently. Examples of this would be celebrations or parades. Figure 2.2 below provides a visual representation for these patterns. (Zheng 2015)

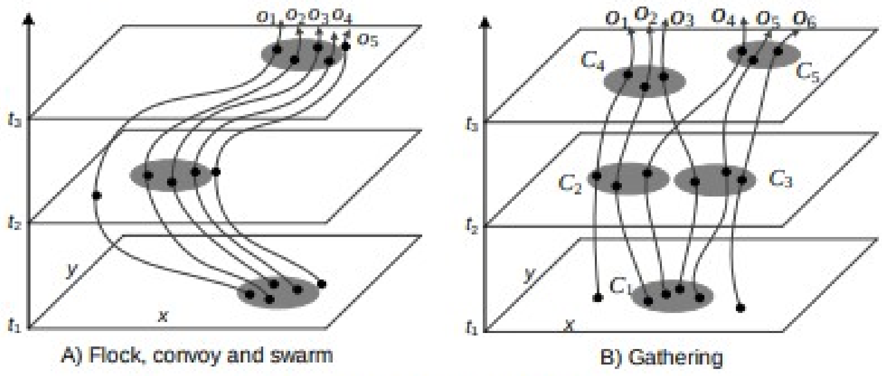

Figure 2.2 Examples of moving together patterns (Zheng 2015)

2.3.2: Trajectory Clustering

The basic principle of trajectory clustering algorithms is to group together similar trajectories as a whole to identify commonalities. However, as observed by Lee et al. (2007), the result of clustering trajectories as a whole could miss common sub-trajectories. These sub-trajectories can prove incredibly useful when conducting an analysis. For this project sub-trajectories would be vital in providing a complete analysis on user lifestyles and behaviour. Lee et al. (2007) proposes a new partition-and-group framework for clustering trajectories. This works by partitioning a trajectory into a set of line segments, and then groups similar line segments together into a cluster. Using this framework, TRACLUS algorithm is developed. While there are many available clustering algorithms, such as k-means, DBSCAN and OPTICS, they mainly deal with the clustering of point data. When dealing with trajectory data the TRACLUS clustering algorithm should prove more useful.

In Figure 2.3 five trajectories are shown with the rectangle identifying the area of commonality within the sub-trajectory. If clustered as a whole, TR1 to TR5 would each be clustered separately as all the trajectories diverge. However, when including the sub-trajectory we can identify common behaviour which is valuable information in analysis.

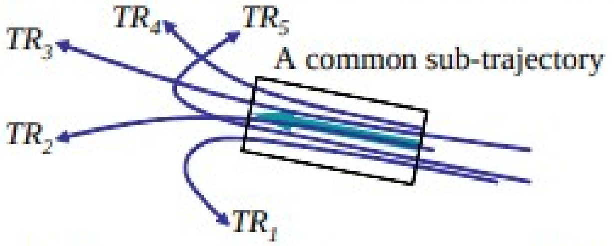

Figure 2.3 An example of a common sub-trajectory (Lee et al. 2007)

Brief outline of the TRACLUS Algorithm

Figure 2.4 below illustrates the overall procedure of trajectory clustering based on the partition-and-group framework. Each trajectory is first partitioned into a set of line segments, then all segments determined to be close to each other are grouped together into a cluster. Lastly, a representative trajectory is generated for each cluster.

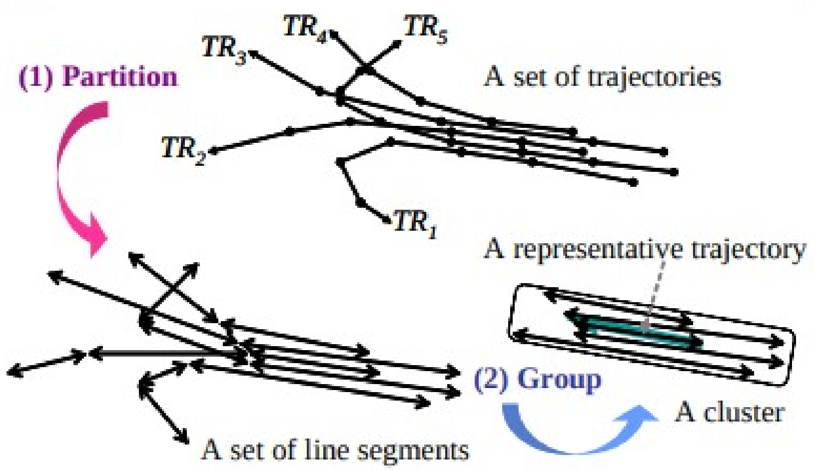

Figure 2.4 An example of trajectory clustering in the partition-and-group framework (Lee et al. 2007)

The distance function used in determining the closeness of line segments is composed of: the perpendicular distance, parallel distance, and the angle distance. The function can best be understood through an intuitive representation in Figure 2.5.

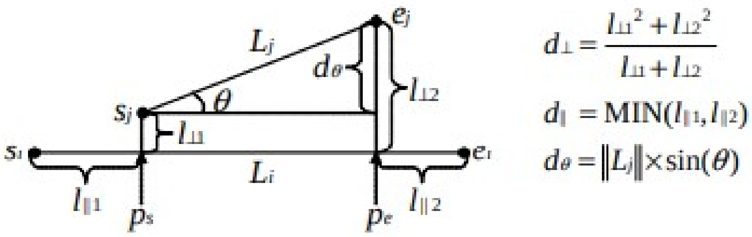

Figure 2.5 Three components of the distance function for line segments (Lee et al. 2007)

The algorithm takes in a set of trajectories, partitions them into segments, then groups the close segments using the values produced through the distance function. Then generates a representative trajectory based on each cluster. This highlights three main parts of the algorithm: partitioning, grouping, and generating the representative trajectory. Each of these aspects will be briefly discussed.

For the partitioning phase, characteristic points within the trajectory need to be identified. These are points where the behaviour of the trajectory changes rapidly. The trajectory will be partitioned at each characteristic point and each partition will be represented by a segment between two points. Optimal partitioning of a trajectory should be both: precise and concise. Precise meaning the difference between a trajectory and its partitions is as small as possible. Concise meaning that the number of partitions is as small as it can be. Next to discuss is the line segment clustering/grouping. The density-based clustering algorithm used in this part of the TRACLUS algorithm is quite similar to the well-known DBSCAN algorithm. The main difference being that not all density-connected sets should become clusters. This is to avoid situations where all line segments in a cluster are from the same initial trajectory. Such clusters would not explain the behaviour of a sufficient number of clusters. Lastly, a representative trajectory of a cluster should describe the overall movement of the individual partitions that make up the cluster. The representative trajectory is generated by performing a sweep across the line segments and counting how many are hit. If this number is equal to or greater than the minimum number of lines, then the average coordinates of those line segments are computed and inserted into the representative trajectory. (Lee et al. 2007)

2.4: Outlier Trajectory Detection

Trajectory outliers can either be whole trajectories themselves or segments of a larger trajectory. They would show a significant difference to the other trajectories. This can be identified by calculating a distance/similarity value representing either shape or travel time. If an trajectory/segment cannot be placed within any density-based cluster, it is likely an outlier. An example of this kind of outlier trajectory could be an unexpected detour in a users daily commute due to roadworks. Lee et al. (2008) detailed a partition-and-detect framework to find anomalous segments of trajectories. This is a similar idea to the partition-and-group framework for trajectory clustering discussed in section 2.3.2.

Trajectory outliers could also be events or observations that do not fit into an expected pattern. An example of this would be traffic congestion caused by a car accident. Traffic anomalies, could be detected by using a large number of trajectories. Such traffic anomalies could be caused by accidents, protests, or natural disasters. (Zheng 2015)

Chapter 3: Requirements and Analysis

This chapter highlights the aims and objectives for the project, analysing its individual parts in detail. The problems involved and potential solutions will be discussed, followed by a way of evaluating the completed project and determining if it has been successful.

3.1: Aims and Objectives

As mentioned in Chapter 1, the goal of this project is to create a tool to analyse a users trajectory data, thus discovering their habits and routines from relevant trajectories. As a result of a large-scale analysis over a large set of users, a model will be produced to cluster users based on certain factors. These could be user interests, locations frequented or the types of activity they like to do. This identifies a few of the main problems to solve in this project which will be discussed further below.

3.1.1: Preparing the data

Before beginning the analysis of a users trajectory data some work must be done to ensure we can produce the best results. For this we will need to do some preprocessing using one or some of the techniques detailed earlier in Chapter 2. The most important technique is likely noise filtering. This is due to the high likelihood of the data received being noisy as a result of how it was collected. Users whose trajectory data is being collected will have on their mobile device an app which tracks their movement. When travelling a user may lose phone signal for a certain period of time affecting the data collection process and thus leading to an inaccurate representation of their true trajectory.

3.1.2: Finding relevant trajectories

To find relevant trajectories for a single user would mean such trajectories display signs of a users common routine or habits. To get to this point from beginning with the initial untouched trajectory data we would use some of the pattern mining methods discussed in Chapter 2. The TRACLUS algorithm previously discussed works well for this as it does not just look at the whole trajectory, but also the individual sub-trajectories that make up a whole one. This leads to nuances in the data being detected and more knowledge can be extracted about a users lifestyle. Some outlier detection can also be performed to help improve the accuracy, so it more closely reflects the users habits.

Aside from discovering relevant trajectories, which show common movement patterns, we can also highlight relevant points along a trajectory at which a user spends some time. Such points are of course the stay points detailed in Chapter 2, which can likely prove useful in providing knowledge into a users lifestyle. Stay points could be the users place of work, their home or places they simply like to frequent. These places would be key in providing insights into their lifestyle. For example, do they regularly go to the pub after work or the gym. Do they do their grocery shopping on a weekday or weekend? Such information can be discovered by looking at stay points and extracting meaning from them, which is discussed next.

3.1.3: Extracting meaning from relevant trajectories

Extracting meaning from relevant trajectories will involve the process of converting the GPS trajectory coordinate data into detailed information about the users habits and lifestyle. Since this trajectory data should already contain the most common patterns within a users day-to-day life, it should be fairly simple to determine meaning. The task will involve identifying what exactly the GPS coordinates for trajectories and stay points represent (e.g. a park, or a restaurant). This can be done using one of the available map APIs and correlating these relevant coordinates with places of interest. One such API I looked at that is capable for this task is OpenStreetMap.

Additionally, the data being collected provides extra information aside from the GPS coordinates and time. The data will also display the mode of transport used during the majority of a certain trajectory (e.g. walking, running, bicycle, car/bus). This can be used to aid in the process of determining what a users lifestyle is like.

3.2: Evaluation

Once the project has been completed and user trajectory data can be analysed and then modelled, there remains the question as to whether it is at all accurate. For instance let us say the data for said user is analysed and the analysis determines that the user has an interest in fitness due to a commonality in their data being detected of them running in a park. How can we be completely certain that this person does in fact have an interest in fitness? The only way to do so would be to find this person and ask them directly, but of course that is not possible since the data is anonymised and to do so through other means would be unethical and likely illegal. Instead evaluating the results will have to be done my using my own trajectory data and seeing if the analysis can determine a truly accurate representation of my lifestyle.

Chapter 4: Progress, Conclusions and Project Plan

This report details all work done so far on the project. In Chapter 2, A survey into relevant research to this project was conducted, highlighting some of the processes and methods that would help in tackling this project. Those most relevant were discussed in Chapter 3 along with how they would be used to help in solving the various problems associated with the project. Additionally, I have been able to look at and use the mobile app collecting the data, as well as the actual trajectory data and have examined how it is formatted and presented. The remaining work for the project is presented below in a Gantt chart.

4.1: Project Plan

The Gantt chart below illustrates the structure and organisation of work for the remainder of the project.

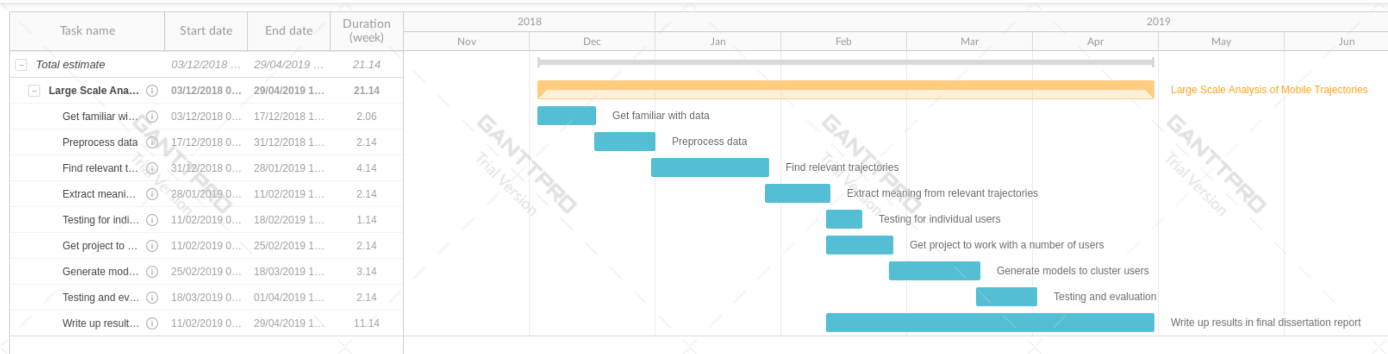

Figure 4.1 Gannt chart displaying work for remainder of project

References

Chakka, V.P., Everspaugh, A.C. and Patel, J.M., 2003. Indexing large trajectory data sets with SETI. Ann Arbor, 1001(48109-2122), p.12.

Greenfeld, J.S., 2002, January. Matching GPS observations to locations on a digital map. In 81th annual meeting of the transportation research board (Vol. 1, No. 3, pp. 164-173).

Hassan, A., Mokhtar, H.M. and Hegazy, O., 2011. A heuristic approach for sensor network outlier detection. Int J Res Rev Wirel Sens Netw (IJRRWSN), 1(4).

Lee, J.G., Han, J. and Whang, K.Y., 2007, June. Trajectory clustering: a partition-and-group framework. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (pp. 593-604). ACM.

Lee, J.G., Han, J. and Li, X., 2008, April. Trajectory outlier detection: A partition-and-detect framework. In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on (pp. 140-149). IEEE.

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. and Ma, W.Y., 2008, November. Mining user similarity based on location history. In Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems (p. 34). ACM.

Li, Z., Ding, B., Han, J. and Kays, R., 2010. Swarm: Mining relaxed temporal moving object clusters. Proceedings of the VLDB Endowment, 3(1-2), pp.723-734.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W. and Huang, Y., 2009, November. Map-matching for low-sampling-rate GPS trajectories. In Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 352-361). ACM.

OpenStreetMap: https://wiki.openstreetmap.org/wiki/Map_Features

Pfoser, D., Jensen, C.S. and Theodoridis, Y., 2000, September. Novel approaches to the indexing of moving object trajectories. In VLDB (pp. 395-406).

Pink, O. and Hummel, B., 2008, October. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on (pp. 862-867). IEEE.

Quddus, M.A., Ochieng, W.Y. and Noland, R.B., 2007. Current map-matching algorithms for transport applications: State-of-the art and future research directions. Transportation research part c: Emerging technologies, 15(5), pp.312-328.

Tang, L.A., Zheng, Y., Yuan, J., Han, J., Leung, A., Hung, C.C. and Peng, W.C., 2012, April. On discovery of traveling companions from streaming trajectories. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on (pp. 186-197). IEEE.

Ye, Y., Zheng, Y., Chen, Y., Feng, J. and Xie, X., 2009, May. Mining individual life pattern based on location history. In Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on (pp. 1-10). IEEE.

Yin, H. and Wolfson, O., 2004, June. A weight-based map matching method in moving objects databases. In Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on (pp. 437-438). IEEE.

Yoon, H. and Shahabi, C., 2008, December. Robust time-referenced segmentation of moving object trajectories. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 1121-1126). IEEE.

Zheng Y., (2015). Trajectory Data Mining: An Overview. ACM Transactions on Intelligent Systems and Technology. 6(3), 1-24