Personalized Travel Route Recommendation Based on GPS Trajectories

Cui, Ge

doctoral thesis

University of Calgary graduate students retain copyright ownership and moral rights for their thesis. You may use this material in any way that is permitted by the Copyright Act or through licensing that has been assigned to the document. For uses that are not allowable under copyright legislation or licensing, you are required to seek permission.

Downloaded from PRISM: https://prism.ucalgary.ca
UNIVERSITY OF CALGARY

Personalized Travel Route Recommendation Based on GPS Trajectories

by

Ge Cui

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

JUNE, 2018

© Ge Cui 2018
Abstract

Travelling is a critical component of daily life. With new technology, personalized travel route recommendations are possible and have become a new research area. A personalized travel route recommendation refers to plan an optimal travel route between two geographical locations, based on the road networks and users’ travel preferences. In this thesis, it first proposes a segment-based map matching method to locate GPS trajectories onto the road network, and then extract users’ travel behaviours from their historical routes. Next, users’ travel behaviour frequencies are estimated by using collaborative filtering technique. This thesis defines two types of travel behaviours, appearance behaviour and transition behaviour, from users’ historical GPS trajectories and propose three personalized travel route recommendation methods, including CTRR, CTRR+ and Map2R, to consider users’ personal travel preferences based on their historical GPS trajectories. A route with the maximum probability of a user’s travel behaviour is then generated. CTRR only considers user’s appearance behaviour and calculates the maximum probability route based on naïve Bayes model. Besides, CTRR is extended to CTRR+ by integrating distance with the user appearance behaviour probability. In MaP2R, it considers both appearance behaviour and transition behaviour, and calculate the maximum probability route based on Markov model. This thesis also conducts some case studies based on a real GPS trajectory dataset from Beijing, China. The experimental results show that the proposed CTRR methods achieve better results for travel route recommendations compared with the shortest distance path method, and both CTRR+ and MaP2R can enhance the performance of CTRR, respectively.
Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Xin Wang for the financial and academic support for my research in the University of Calgary. Her great patience, continued guidance, kind encouragement and invaluable advice help me keep going forward during my Ph.D. program and lead me to the world of data mining. This thesis would not be completed without her guidance.

I would like to thank my supervisory committee Dr. Steve Liang, Dr. Ruisheng Wang, Dr. Rangelova Elena from Department of Geomatics Engineering, Dr. Zhangxing Chen from Department of Chemical and Petroleum Engineering of University of Calgary, Dr. Ming Zhong from Wuhan Univeristy of Technology. I am deeply grateful for their insightful suggestions on my thesis. I would also like to thank Dr. Andrew Hunter and Dr. Stevanus Tjandra for serving in my candidacy exam.

Many thanks to the group members in the university, Dequan Zhou, Mohammdreza Rahimi, Abul Kalam Azad, Yuanchen Li, Helen Pinto, Xiaodong Sun, Wexin Yang, Bingjie Wei, Rodrigo Silva. I enjoy the discussions with you. My thanks also expressed to my dear friends in Calgary and in China, Hang Liu, Junbo Shi, Tao Zeng, Lingling Duan, Ting Lei, Bing Yang, Jun Cui, Tian Lan, Wen Xie, Ran Liu, Shanzi Wang, Yue Dou, Zhengyin Sun, Jun Li. Thank you for making my life exciting and joyful.

Academic guidance is also appreciated when I studied in Wuhan University, China. I would like to thank my supervisor Dr. Yaolin Liu who supervises me during my graduate study. I would like to thank Dr. Yanfang Liu and Dr. Jiping Liu for giving me the guidance when I studied in Wuhan University.
I would like to thank Mitacs offering me Globalink Research Wards to support my internship in Shenzhen Institutes of Advanced Technology. I am grateful for the guidance and help from Dr. Jun Luo during the internship. I would like to thank Mitacs Accelerate Program to giving me the opportunity to work with Office of Safety, City of Edmonton. This collaboration broadens my research view. I really appreciate the valuable suggestions from Dr. Daewon Kwon in City of Edmonton. I would like to thank the financial support from University Research Travel Grant, Schulich Student Activity Fund and Department FGS Scholarship in University of Calgary.

I would like to thank my family for their caring and support, my uncles Wenzhuan Cui, Ming Cui, Guoquan Ouyang, Anmin Zhang, my aunts Jihua Chen, Dongmei Song, Xiaowang Cui, Qun Yan, my cousins Shan Cui, Xiang Cui and Haoyu Ouyang.

Most importantly, I would like to give a sincere thank to my parents, Jin Cui and Wenting Zhang. Their endless love, patience, understanding and support is my foundation for my dream. Thank you for everything in my life. I love you!
Table of Content

Abstract .......................................................................................................................... i
Acknowledgements ...................................................................................................... ii
Table of Content ......................................................................................................... iv
List of Tables ................................................................................................................ vi
List of Figures .............................................................................................................. vii
List of Symbols ............................................................................................................. ix
List of Abbreviations .................................................................................................. xii

Chapter 1: Introduction ............................................................................................... 1
  1.1 Background ........................................................................................................... 1
  1.2 Challenges ............................................................................................................ 3
  1.3 Objectives ............................................................................................................ 9
  1.4 Contributions ...................................................................................................... 9
  1.5 Organization of the Thesis .................................................................................. 13

Chapter 2: Related Works .......................................................................................... 16
  2.1 Map Matching ..................................................................................................... 16
      2.1.1 Point-based map matching ....................................................................... 17
      2.1.2 Segment-based map matching ................................................................ 21
  2.2 GPS Trajectory Segmentation ........................................................................... 22
      2.2.1 Trajectory segmentation based on spatial shape .................................... 23
      2.2.2 Trajectory segmentation based on semantic point ............................... 24
      2.2.3 Trajectory segmentation based on time interval .................................... 25
  2.3 Recommendation methods .................................................................................. 25
      2.3.1 Content-based method .......................................................................... 26
      2.3.2 Collaborative filtering method ............................................................... 29
      2.3.3 Summary for recommendation method .................................................. 33
  2.4 Mobility Recommendation based on GPS Trajectories ..................................... 34
  2.5 Route Recommendation .................................................................................... 36
      2.5.1 Traditional route recommendation ......................................................... 36
      2.5.2 Popular route recommendation using trajectories .................................. 36
      2.5.3 Personalized route recommendation using GPS trajectories ............... 39
  2.6 Summary .............................................................................................................. 43

Chapter 3: Segment-based hidden Markov model for map matching ...................... 45
  3.1 Map Matching Problem ...................................................................................... 45
  3.2 Definitions ........................................................................................................... 50
  3.3 Segment-based Hidden Markov Model ............................................................... 52
      3.3.1 Overview of SHMM .............................................................................. 52
      3.3.2 Data preprocessing ............................................................................... 54
      3.3.3 Trajectory segmentation and candidate road segment sequence search ... 55
      3.3.4 LCS-HP Distance measurement ............................................................. 61
      3.3.5 Map matching in hidden Markov model ............................................... 64
List of Tables

Table 4.1 Frequencies of appearance behaviours ......................................................... 86
Table 4.2 Performances of CTRR with respect to different time interval sizes ............ 113
Table 5.1 Frequencies of transition behaviours ............................................................. 120
List of Figures

Figure 1.1 Examples from Geolife data showing routes taken by two different users and the shortest path ................................................................. 2
Figure 1.2 Example of map matching for GPS trajectory ...................................................... 5
Figure 1.3 GPS trajectories of one user ........................................................................ 8
Figure 2.1 Illustration of HMM ......................................................................................... 19
Figure 3.1 Example of point-based map matching result based on hidden Markov model ..................................................................................... 47
Figure 3.2 Road segment sequence .................................................................................. 51
Figure 3.3 The framework of SHMM ................................................................................. 53
Figure 3.4 Examples of GPS reading outliers. (a) Outlier far from road network. (b) Outlier with unreasonable speed ........................................................................ 55
Figure 3.5 Pseudocode of TSHH ....................................................................................... 57
Figure 3.6 Trajectory segmentation based on heading homogeneity ................................. 58
Figure 3.7 Example of candidate RSS search .................................................................... 60
Figure 3.8 Pseudocode of candidate road segment search ................................................. 61
Figure 3.9 Segment-based map matching using HMM ...................................................... 65
Figure 3.10 Transition probability in SHMM ..................................................................... 69
Figure 3.11 Distribution of length of GPS trajectories ...................................................... 70
Figure 3.12 Accuracy evaluation of the experiments ......................................................... 71
Figure 3.13 a real GPS trajectory .................................................................................... 74
Figure 3.14 Map matching result of the GPS trajectory by SHMM .................................... 75
Figure 3.15 The impact of sampling rates on accuracy ...................................................... 76
Figure 3.16 Comparison of running time between SHMM and PHMM ............................. 77
Figure 3.17 Running time of SHMM vs. No. of GPS points .............................................. 78
Figure 3.18 Running time of SHMM vs No. of GPS sub-trajectories ............................... 78
Figure 4.1 Example of two users travel in the road networks. .............................................. 85

Figure 4.2 Framework of the CTRR method................................................................. 87

Figure 4.3 Histogram of sampling rates between GPS readings. ............................... 91

Figure 4.4 User-appearance behaviour matrix.......................................................... 93

Figure 4.5 Pseudo-code of estimating users’ appearance behaviour frequency. .............. 95

Figure 4.6 Pseudo-code of calculating user’s log-inversed appearance behaviour probability on a road segment................................................................. 100

Figure 4.7 Histogram of the length of the trips............................................................ 104

Figure 4.8 Performance evaluation of three methods in precision (a) and recall (b)..... 107

Figure 4.9 Route recommendation results of three methods and the real route. .......... 109

Figure 4.10 Precision (a) and recall (b) for cold start users........................................ 111

Figure 4.11 Example of route recommendation result for a cold start user............... 112

Figure 4.12 The performances of CTRR+ versus power parameter α.......................... 114

Figure 5.1 Dependency relationship between appearance behaviours ....................... 119

Figure 5.2 An example showing the relationship between the road work and the behaviour graph. ................................................................. 127

Figure 5.3 The effect of number of latent factors in MF vs. (a) RMSE and (b) training time. .................................................................................................................. 129

Figure 5.4 Performance Comparison among SDR, CTRR and MaP2R for four groups of trips with different lengths (a) precision of number of road segments; (b) recall of number of road segments; (c) precision of length of road segments; (d) recall of length of road segments. ................................................................. 131

Figure 5.5 Comparison among MaP2R, Google Map and Baidu Map............... 135
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>Road network</td>
</tr>
<tr>
<td>$V$</td>
<td>A set of vertices representing junctions in road network</td>
</tr>
<tr>
<td>$E$</td>
<td>A set of directed edges representing the road segments</td>
</tr>
<tr>
<td>$v$</td>
<td>Terminal point of road segment</td>
</tr>
<tr>
<td>$e$</td>
<td>Road segment</td>
</tr>
<tr>
<td>$rss$</td>
<td>Road segment sequence</td>
</tr>
<tr>
<td>$p$</td>
<td>GPS reading</td>
</tr>
<tr>
<td>$t$</td>
<td>Timestamp of GPS reading</td>
</tr>
<tr>
<td>$lat$</td>
<td>Latitude of GPS reading</td>
</tr>
<tr>
<td>$lng$</td>
<td>Longitude of GPS reading</td>
</tr>
<tr>
<td>$trj$</td>
<td>GPS trajectory</td>
</tr>
<tr>
<td>$strj$</td>
<td>GPS sub-trajectory</td>
</tr>
<tr>
<td>$\theta_{i,j}$</td>
<td>Heading between two GPS reading $p_i$ and $p_j$</td>
</tr>
<tr>
<td>$D(\theta_{i,j}, \theta_{k,l})$</td>
<td>The angle difference between two headings $\theta_{i,j}$ and $\theta_{k,l}$</td>
</tr>
<tr>
<td>$\theta_{strj}$</td>
<td>Heading of GPS sub-trajectory $strj$</td>
</tr>
<tr>
<td>$r$</td>
<td>Search radius</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Speed constraint of moving vehicles</td>
</tr>
</tbody>
</table>
\( \gamma \)  
Heading change threshold for trajectory segmentation with heading homogeneity

\( \beta \)  
Heading change threshold for candidate RSS search

\( \nu(\text{rss}) \)  
Probability of the sequence of the last hidden states where the last state is \( \text{rss} \)

\( \Pr(\text{strj}|\text{rss}) \)  
Emission probability of \( \text{strj} \) is located on \( \text{rss} \)

\( \Pr(\text{rss}_j|\text{rss}_i, \text{strj}_{i-1}, \text{strj}_t) \)  
Transition probability of moving from \( \text{rss}_i \) of \( \text{strj}_{i-1} \) to \( \text{rss}_j \) of \( \text{strj}_t \)

\( \text{Fre}_\text{dis} \)  
Fréchet distance

\( \sigma_z \)  
Parameter in emission probability

\( \rho \)  
Parameter in transition probability

\( rt \)  
Route in road network

\( b \)  
Appearance behaviour

\( tb \)  
Transition behaviour

\( u \)  
User

\( frq \)  
Frequency of travel behaviours

\( s \)  
Sampling rate

\( p_u \)  
User latent factor vector

\( q_b \)  
Travel behaviour latent factor vector

\( UB \)  
User-travel behaviour matrix

\( \alpha \)  
Power parameter in CTRR+
$a$  Smoothing value for travel behaviour probability estimation

$c$  Smoothing value for transition behaviour probability estimation

$tb_{i\rightarrow j}$  Transition behaviour from appearance behaviour $b_i$ to $b_j$

$\varphi$  Threshold for time-interval based GPS trajectory segmentation

$\varepsilon_d$  The distance threshold between GPS point and road segment

$\varepsilon_h$  The heading threshold between GPS point and road segment
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>CTRR</td>
<td>Collaborative Travel Route Recommendation</td>
</tr>
<tr>
<td>MaP2R</td>
<td>Maximum Probability Route Recommendation</td>
</tr>
<tr>
<td>CF</td>
<td>Collaborative Filtering</td>
</tr>
<tr>
<td>MF</td>
<td>Matrix Factorization</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Descent</td>
</tr>
<tr>
<td>RWR</td>
<td>Random Walk with Restart</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>PRR</td>
<td>Personalized Route Recommendation</td>
</tr>
<tr>
<td>RSS</td>
<td>Road Segment Sequence</td>
</tr>
<tr>
<td>TSHH</td>
<td>Trajectory Segmentation with Heading Homogeneity</td>
</tr>
<tr>
<td>STRJ</td>
<td>GPS Sub-trajectory</td>
</tr>
<tr>
<td>AN</td>
<td>Accuracy by Number of Road Segment</td>
</tr>
<tr>
<td>AL</td>
<td>Accuracy by Length of Road Segment</td>
</tr>
<tr>
<td>SDR</td>
<td>Shortest Distance Path</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Squared Error</td>
</tr>
<tr>
<td>VSM</td>
<td>Vector Space Model</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>Term Frequency-Inverse Document Frequency</td>
</tr>
<tr>
<td>LCS</td>
<td>Longest Common Subsequence</td>
</tr>
<tr>
<td>LCS-HP</td>
<td>Longest Common Subsequence with Heading Penalty</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

1.1 Background

People travel between different locations for working, trading goods, recreation and other social activities, so travelling is a critical component in daily life, and route recommendation is one of the most popular services for travel planning.

The current route recommendation services generally consider a certain metric such as the shortest distance or least traveling time, and provide the shortest or quickest path between locations to users; however, the recommended path would not often be chosen in real travel (Letchner et al. 2006, Delling and Wagner 2009). Moreover, different users may prefer to travel on different routes, but current route recommendation services, such as Google Maps (www.google.com/maps, 2018), Baidu Maps (map.baidu.com, 2018) and Here Maps (wego.here.com, 2018) do not take account of users’ personal travel preferences.

Figure 1.1 shows the travel routes of users extracted from a real world dataset, i.e. Geolife dataset (Zheng et al. 2008a, 2009) and routes provided using the shortest distance. As shown in the figure, two users (user #9 and user #10) travelling from the same origin to the same destination at a similar time (between 7 and 8 am) took different routes (shown in solid gray line and black line): neither of the routes was the shortest distance path (in dash line). One reason that users do not follow routes recommended by existing route planning services is that they do not take into account individual user’s travel preferences.
Figure 1.1 Examples from Geolife data showing routes taken by two different users and the shortest distance path
Personalized travel route recommendation is an important research topic and has not been fully explored. Personalized travel route recommendation refers to the planning of an optimal travel route between two geographical locations based on the road networks and users’ travel preferences. However, users’ specific travel preferences are influenced by many factors (Szczerba et al. 2000, Delling et al. 2009, Niaraki and Kim 2009), such as distance, traffic volume, travelling time, weather, fuel consumption, safety, and many other implicit and hidden factors. Thus, it is difficult to determine a comprehensive user travel preference metric to develop personalized travel route recommendation.

With advances in Global Positioning System (GPS) technology and the popularity of the mobile devices, massive amounts of human movement data in GPS trajectories have been collected and are available for research, which provides an alternative way to make personalized travel route recommendation (Letchner et al. 2006, Papinski et al. 2009, He et al. 2012). Users’ historical GPS trajectories contain information about travelling time and location, which makes it possible to understand users’ travel behaviours.

1.2 Challenges

There are, however, a few challenges when using GPS trajectories for personalized travel route recommendation.

First, GPS readings are recorded over time, and an issue of uncertainty may arise from GPS readings lost for a long time due to “urban canyon” (Marmasse and Schmandt 2000), device malfunction (McCluskey et al. 2012) or human interruption (e.g. driver turning it off purposely) (Du and Aultman-Hall 2007), which could lead to imprecise extraction of user’s travel behaviours from GPS trajectories. Therefore, a GPS trajectory
often needs to be divided into several sub-trajectories to avoid the uncertainty. The division of these GPS trajectories into sub-trajectories and describe users’ travel behaviours at different times of day is an issue that must be addressed.

Besides, another uncertainty issue of GPS trajectories is resulted from that the spatial position of GPS trajectories is usually imprecise due to the measurement error (Van Diggelen 1998) and low sampling rates of GPS receivers (Greenfeld 2002). Thus, it requires to recover users’ historical travel routes by locating users’ GPS trajectory onto the road network so that their travel behaviours can be understood. The process of locating GPS trajectories onto the road segments in the road network is called map matching, which is shown in Figure 1.2.

In Figure 1.2, a user’s GPS trajectory (shown in black circles) is displayed in the road network (shown in gray lines). Due to the measurement error, GPS points in the trajectory deviate from their real locations, which leads to the uncertainty that the trajectory can be matched to several routes in the road network. Map matching is to locate the GPS trajectory onto its real route (shown in red line) in the road network based on the characteristics of GPS trajectory and the road network, such as geometric feature, topological feature and speed information.

In the era of information technology, the vast amount of GPS trajectory data is generated per second due to the properties of high “volume” and “velocity” of big data, which requires map matching algorithm can deal with GPS trajectories efficiently. Moreover, some global map matching methods (Newson and Krumm 2009, Lou et al. 2009) may be time consuming for the trajectories with large GPS readings, because they
Figure 1.2 Example of map matching for GPS trajectory
need to achieve a global optimal matching result for the entire trajectory, which will result in huge increase in time cost. Hence, to improve the efficiency of map matching algorithm is significant.

Furthermore, the performance of map matching usually decreases, as the measurement error of GPS point increases. Most current map matching algorithms locate GPS point onto the road segment with the largest similarity which is mainly measured based on the spatial proximity between them. Thus, GPS points are often located onto the incorrect road segment when the measurement error increases. Hence, to improve the effectiveness of map matching algorithm is also worth studying.

One potential improvement for map matching is to divide GPS trajectory into GPS sub-trajectories with homogeneous attribute, and then GPS sub-trajectory can be snapped to the sequence of road segment with the advantage that the attribute of sub-trajectory can reduce the uncertainty between them. If GPS trajectory is divided into GPS sub-trajectories for map matching, it is significant to explore an effective method to measure the distance between GPS sub-trajectories and the sequence of road segments. Moreover, the distance metric should consider the attribute of the sub-trajectories.

After map matching, users’ GPS trajectory can be represented as historical travel routes in the road network. There are a few challenges to make a personalized route recommendation based on users’ historical travel routes.

As a user generally travels on only a few routine routes daily, each user’s GPS trajectories only cover limited road segments of the road network. For instance, Figure 1.3 displays all GPS trajectory data of one user, and it can be observed that the GPS trajectories for the user only cover a small part of the city. Hence, a user’s travel behaviour probability
estimation on the road segments that he/she has never travelled is also critical for route recommendation, especially for users whose travel behaviours are not available and cases when users intend to travel to unfamiliar locations.

Additionally, the temporal factor also plays an important role in the route recommendation. Similar to the spatial coverage of historical GPS trajectories, user’s GPS trajectories are mainly concentrated in the limited time intervals such as the rush hours from home to work, and there exists no GPS trajectory in certain time intervals for some users. Therefore, how to make use of the temporal information from the historical GPS trajectory to estimate users’ travel behaviours at different time intervals is also worth researching.

Distance is always an important and available information for route recommendation. Users may not travel on the shortest distance route in most occasions, but they may prefer shorter distance route when the other information of travel routes is unknown. Thus, personalized travel route recommendation needs to integrate distance with user’s travel behaviours learned from GPS trajectories, in order to determine routes with the highest likeliness for travel. This integration has not been explored by existing route recommendation methods.

Last, it is significant to build a model to recommend user an effective personalized route based on the user’s travel behaviours learned from GPS trajectories. Many researches have revealed that Markov property plays an important role in movement models (Letchner et al. 2006, Chen et al. 2011, Dai et al. 2015), and ignorance of the dependency relationship between travel behaviours may produce ineffective recommendation in the route planning.
Figure 1.3 GPS trajectories of one user
How to consider the dependency relationship between travel behaviours in route recommendation should be studied.

1.3 Objectives

The major objective of this thesis is to provide users a personalized travel route recommendation based on their historical GPS trajectories. To be more specific, given the user, the query time, and the origin and destination location in the road network, it aims to provide the travel route from the origin to the destination which the user may prefer to travel on with the maximum probability.

To achieve the major objective of this thesis and address the aforementioned challenges, the specific objectives should be done as follows:

(1) Locate GPS trajectories onto road network with an efficient and effective map matching method.

(2) Understand users’ travel preference from GPS trajectories, and estimate the user’s travel preference on the time and road segment which they have never travelled on before.

(3) Provide user a personalized travel route which he/she prefer to travel on with the maximum probability based on the historical GPS trajectories.

1.4 Contributions

In this thesis, three personalized travel route recommendation methods are proposed, called collaborative travel route recommendation (CTRR) and the extended versions of CTRR, called CTRR+, and the maximum probability route recommendation
(MaP2R). This thesis only considers that users travel in a single transportation mode, and the purpose of a travel is not considered. All algorithms recommend personalized travel routes in the single transportation mode based on users’ historical GPS trajectories.

In this thesis, a concept, called travel behaviour, is first proposed to describe user’s travel preferences in time intervals and road segments. There are two types of travel behaviours, called appearance behaviour and transition behaviour, respectively. Appearance behaviour describes when and which road segment a user prefers to travel on; transition behaviour represents the dependency relationship between appearance behaviours, transition from one appearance behaviour to another one. The travel behaviours for each user are then extracted from his/her historical GPS trajectories after map matching. The relevant frequencies of the travel behaviours can then be transformed to travel behaviour probability.

To locate users’ GPS trajectories onto the road network efficiently and effectively, the segment-based hidden Markov model is proposed, called SHMM. SHMM divides a GPS trajectory into several GPS sub-trajectories with homogeneous heading, and then locates GPS sub-trajectories onto the road network in the hidden Markov model. SHMM processes a sequence of homogeneous GPS points together, which could reduce the computation time and increases the accuracy.

In SHMM, the distance between GPS sub-trajectory and candidate road segment sequence is calculated based on the proposed distance measurement method, called longest common subsequence with heading penalty (LCS-HP). LCS-HP considers the heading information when measuring the distance between GPS sub-trajectories and the candidate
road segment sequences, and gives penalty to the road segment sequences which have
dissimilar heading with the target GPS sub-trajectory.

Collaborative filtering (CF), which relies only on past user feedback, is one of the
most widely used recommendation methods. CF can analyze collected information to
provide personalized recommendations to suit a user’s taste (Koren et al. 2009). Matrix
factorization (MF) is one of the most widely used CF approach and is used in this thesis to
infer a user’s implicit travel preference (i.e. a user’s travel behaviour frequencies) by
aggregating the travel behaviours of users that have similar travel behaviour patterns.
Moreover, as the temporal factor is considered in travel behaviour, users’ travel behaviours
in the matrix are very sparse. To address it, the correlation between travel behaviours in
the nearby time interval is considered, and the frequency of user’s travel behaviour is
estimated by smoothing the frequencies of the user’s travel behaviour in the nearby time
intervals.

For route recommendation based on the estimated travel behaviour probability for
each user, the naïve Bayes model and the Markov model are utilized, respectively. To be
more specific, CTRR only considers users’ appearance behaviours based on the naïve
Bayes model while MaP2R considers both appearance behaviours and transition
behaviours based on the Markov model. As for the users whose travel behaviours are not
available (i.e. cold start users), CTRR+ estimates their travel behaviour probabilities based
on other users. Moreover, CTRR+ extends CTRR by integrating distance with travel
behaviour probability.

In summary, this thesis has the following contributions:
1. The concept of travel behaviour is proposed as the basic element to describe users’ travel preference from GPS trajectories. Two types of travel behaviours, appearance behaviour and transition behaviour, are studied in this thesis. To better extract and represent the travel behaviours from the GPS trajectory data, GPS trajectories are partitioned into GPS sub-trajectories based on time interval threshold for avoiding uncertainty. An entropy-based histogram thresholding method is proposed to determine the time interval threshold based on the frequencies of GPS sampling rates.

2. A segment-based hidden Markov model for map matching, called SHMM, is proposed to locate GPS trajectories onto the road network. SHMM divides a GPS trajectory into GPS sub-trajectories where GPS readings have similar heading, and matches GPS sub-trajectories onto the road segment sequences in the hidden Markov model. SHMM processes a sequence of similar GPS points together, rather than an individual GPS points, which can improve the efficiency. Besides, SHMM compares the similarity between GPS sub-trajectories and route segment sequences based on heading and geometric information, which can reduce the negative impact of the measurement error of individual GPS point.

3. This thesis proposes a method, called LCS-HP, to measure the distance between GPS sub-trajectories and road segment sequences. LCS-HP introduces heading information for the similarity measurement between trajectories. Compared with Fréchet distance, LCS-HP is robust to the measurement error of GPS points. Compared with LCS and edit distance with real penalty (EDR), LCS-HP introduces
penalty for dissimilar headings, which is more reasonable for the distance measurement between GPS trajectories and road segment sequences.

4. This thesis addresses the sparseness of user trajectory coverage on the road network by applying matrix factorization to estimate the frequencies of users’ missing travel behaviours, and utilizing Laplace smoothing method to estimate the probability of users’ missing travel behaviours. Additionally, a temporal decay function is used to describe the temporal correlations among travel behaviours.

5. Naïve Bayes model is utilized to generate a route with the maximized implicit appearance behaviour probability along the route. This thesis proposes CTRR method to calculate the maximum probability route for each user based on their historical appearance behaviours. CTRR+ is proposed to improve CTRR by integrating distance with the appearance behaviour probability, which can correct the limitation of CTRR and improve the performance of travel route recommendation.

6. By considering both appearance behaviour and transition behaviour, MaP2R method is proposed to calculate the maximum probability route based on Markov model. Compared with CTRR, MaP2R extracts more travel information from users’ historical GPS trajectories, and improve the performance of travel route recommendation.

1.5 Organization of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 reviews related work. This chapter first reviews two significant pre-processing steps for GPS trajectory, GPS
trajectory map matching and GPS trajectory segmentation. Current map matching methods are discussed and compared. Besides, different trajectory segmentation methods are introduced based on the distinct motivations. Then, some recommendation methods are introduced. Next, this chapter reviews some relevant research topics about the human mobility by using GPS trajectories. Last, the route recommendation methods are reviewed, including the traditional route recommendation, popular route discovery and personalized route recommendation.

In Chapter 3, the segment-based hidden Markov model, called SHMM, is proposed for GPS trajectory map matching. SHMM first divides GPS trajectories into homogeneous GPS sub-trajectories with similar heading, and then searches the candidate route segment sequences for each GPS sub-trajectory. Last, a hidden Markov model is built with GPS sub-trajectories and the corresponding route segment sequences. Based on the topology of the road network and the similarity measurement between GPS sub-trajectories and candidate road segment sequences, the global optimal matching result between GPS sub-trajectories and road segment sequences with the largest probability is calculated in hidden Markov model. The experimental result shows that the performance of SHMM is better than the baseline methods.

In Chapter 4, a collaborative travel route recommendation method, called CTRR, is proposed to provide user a personalized route recommendation. First, GPS trajectories are processed with the proposed entropy-based thresholding method and map matching method. Second, users’ appearance behaviours are extracted from the GPS trajectories, and the probabilities of users’ appearance behaviours are estimated using matrix factorization and Laplace smoothing method. Last, the maximum probability route which user may
prefer is calculated based on the naïve Bayes model. CTRR is extended to CTRR+ by integrating distance, and the experimental results show that CTRR and CTRR+ outperform the baseline method, respectively.

In Chapter 5, another personalized route recommendation method, called MaP2R, is proposed. Different from CTRR, MaP2R extracts two types of travel behaviours, appearance behaviour and transition behaviour, from user’s historical GPS trajectories, and calculate the maximum probability route based on the Markov model in a built behaviour graph. The experiment results exhibit that MaP2R has a better performance than the shortest distance path method and CTRR.

In Chapter 6, the conclusions are made for the thesis and suggestions are given for future research.
Chapter 2: Related Works

This chapter reviews the literature about map matching for GPS trajectories (Chapter 2.1), GPS trajectory segmentation (Chapter 2.2), collaborative filtering (Chapter 2.3), mobility recommendation based on GPS trajectories (Chapter 2.4) and route recommendation method (Chapter 2.5). Chapter 2.1 discusses the current map matching methods for GPS trajectories. Chapter 2.2 reviews the different trajectory segmentation methods based on distinct motivations. Chapter 2.3 discusses several recommendation methods. Chapter 2.4 introduces human mobility recommendation related to GPS trajectories and Chapter 2.5 discusses the route recommendation methods. Chapter 2.6 gives a summary of the literature review.

2.1 Map Matching

Map matching is the process to locate GPS trajectories onto the road network, which is an important pre-processing step for the applications related to GPS trajectories. In the last several decades, many map matching algorithms have been proposed. Quddus et al. (2007) conducted a literature review of the map matching algorithms, and divided them into geometric method, topological method (topological analysis of spatial road network data), probability theory and other advanced methods such as Kalman filter (Kim et al. 2000, Obradovic et al. 2006), fuzzy logic (Syed and Cannon 2004, Quddus et al. 2006), and state space model (Gustafsson et al. 2002). Besides, they pointed out that the determination of a vehicle location on a road depends to a large extent on both the quality of the spatial road map and the used map matching algorithm.
Map matching methods can be divided into two categories, the point-based map matching and the segment-based map matching, based on the strategy to process GPS trajectories.

### 2.1.1 Point-based map matching

Most current methods are point-based map matching methods which process individual GPS points in a GPS trajectory separately. Point-based map matching methods can be divided into two groups: local method and global method. The local method (also called incremental method) considers the local optimum matching between GPS points and road segments, while the global method pursues a global optimum matching for all GPS points.

#### 2.1.1.1 Local map matching method

One important local map matching method is proposed by Greenfeld (2002), which is an incremental map matching algorithm based on the geometry and topology. Firstly, it searched candidate road segments for the individual GPS reading based on the topology of the road networks. Specifically, the candidate road segments of the current GPS reading are those connected to the road segment matched to the previous GPS reading. Then, it utilized a similarity measure to locate the GPS points to the road segment with a weighting system of balancing spatial proximity, intersection and direction which are calculated between the candidate road segment and the line segment composed by two adjacent GPS readings.
Brakatsoulas et al. (2005) adopted the weighting system of Greenfeld’s work, and proposed a look-ahead policy to make a local matching decision by exploring a sequence of road segments rather than a single road segment. Compared with only considering the measure similarity between the target GPS reading and its candidate road segment, the look-ahead policy also considers the similarity between the next few GPS readings of the target GPS reading and the adjacent road segments of the candidate road segment. However, the above two map matching methods only simply utilize the geometric features and the topology of road network, which may identify the road segment incorrectly in the road network when the measurement error of some individual GPS reading increases. Besides, an incorrect match on some road segments may result in a continuous mismatch by these map matching methods.

2.1.1.2 Global map matching method

Some advanced global methods have been proposed for the point-based map matching strategy. Lou et al. (2009) came up with the ST-Matching algorithm by integrating the geometry and speed information for GPS trajectory with low sampling rate. Firstly, this algorithm calculated the spatial analysis function by the multiplication of observation probability and transmission probability. The observation probability was measured based on the geometry of GPS points and road segments, and the transmission probability was measured base on the topology of the road network. Secondly, the temporal analysis function was developed based on the average speed of two candidate points of the GPS points on the road segments and the typical speed value of each road segment. Last,
it constructed a candidate graph and identified the best matching path sequence in the graph.

Yuan et al. (2010) proposed a method, called IVMM, based on ST-Matching algorithm. IVMM adopted the spatial analysis function and the temporal analysis function in ST-Matching. Moreover, it considered the interactive influence between GPS points, and assumed that the influence made by the close GPS point was more important than the influence made by the GPS point far away. However, both ST-Matching algorithm and IVMM required extra road network information such as speed information of road segments, which is sometimes unavailable.

Newson and Krumm (2009) proposed a map matching method based on hidden Markov model (HMM). Hidden Markov model is a probabilistic model for sequential data, which computes the probability distribution over possible sequences and chooses the sequence with the maximum probability. Figure 2.1 gives an illustration about HMM.

![Figure 2.1 Illustration of HMM](image)

In this figure, HMM consists of a sequence of observations and hidden states in discrete time steps $t \in \{1, ..., T\}$. $O_t$ and $S_t$ stands for the observation and the hidden state at time step $t$, respectively. It assumes that the observation is generated by some process whose state is hidden from the observation. Besides, the state of this hidden process
satisfies Markov model which means the state $S_t$ is only dependent on the previous state $S_{t-1}$, but independent of any state prior to $S_{t-1}$. The probability distribution over the sequence can be defined as the initial probability for the states at the starting of the model, the transition probability between states $Pr(S_t|S_{t-1})$ and the emission probability $Pr(O_t|S_t)$. This corresponding joint distribution has the form based on the Markov property (Murphy 2012, p.606)

$$Pr(S_{1:T}, O_{1:T}) = Pr(S_1)Pr(O_{1:T}|S_{1:T}) = \left[Pr(S_1) \prod_{t=2}^{T} Pr(S_t|S_{t-1}) \prod_{t=1}^{T} Pr(O_t|S_t)\right]$$

In HMM, the sequence of the hidden states with the maximum probability in the discrete time steps can be estimated based on the sequence of observations, transition probability and emission probability. Based on this characteristic of HMM, it can be utilized to solve map matching problem. GPS points are regarded as observations in the discrete time steps in order, and road segments are taken as the hidden states. With the given transition probability and emission probability, the sequence of road segments with the maximum probability can be calculated out based on the observed GPS points.

Newson and Krumm (2009) calculated the emission probability based on the spatial proximity between GPS points and the candidate road segments. They assumed that the emission probability between GPS point and its real road segment obeyed Gaussian distribution with respective to the spatial distance between them. The larger distance means the smaller emission probability. Besides, they assumed that the network distance between the GPS point $o_t$ and the next GPS point $o_{t+1}$ in the road network should be the same with the great circle distance between $o_t$ and $o_{t+1}$. By observing the distribution of the difference of the two distance values, they assumed that the distance difference followed
an exponential distribution and calculated the transition probability based on the exponential distribution with respective to distance difference. Last, Viterbi algorithm was used to calculate the sequence of the road segments with the maximum probability.

Compared with the local methods using the geometric and topological analysis (Greenfeld 2002, Brakatsoulas et al. 2005), the approach proposed by Newson and Krumm (2009) integrates geometric and topological features into the hidden Markov model which considers the best global optimum matching between GPS points and road segments rather than local optimum. Thus, the method using hidden Markov model has achieved a better performance, and has been considered as the classic map matching approach (Goh et al. 2012, Oran and Jaillet 2013, Li et al. 2015). However, the deficiency of this method is that the similarity measurement between individual GPS point and the road segment based on spatial proximity is unreliable when the measurement error of GPS points increases, which may generate unreasonable detours in the road network.

2.1.2 Segment-based map matching

Very few segment-based methods have been studied for map matching. For instance, Chawathe (2007) proposed to match a sequence of $k$ continuous GPS points to road segments with the integrated score value from high to low in order, but the determination of the parameter $k$ is not discussed. Moreover, it does not provide a specific method for calculating the score value of sequences. Thus, it does not fundamentally solve the problem of the local/incremental algorithms based on the topology of road network, especially when the sequence with highest score (for example, based on distance) is matched onto the incorrect road segments.
Zhu et al. (2017) proposed a trajectory segmentation map matching approach for high-resolution GPS trajectory. They calculated the shortest distance path between the first and last GPS point of a trajectory, and compared the similarity between the shortest distance path and the GPS trajectory. If the similarity value was less than a threshold, the trajectory would be divided into sub-trajectories and repeat the similarity comparison. When the map matching is finished, some sub-trajectories might be skipped without matching if no similar path could be found in the road network. The proposed method could improve the map matching efficiency, as it only compares the similarity between the GPS trajectory and the shortest distance path between the start and end points of the GPS trajectory, instead of searching candidate routes of the GPS trajectory. However, only considering the shortest distance path may be unreasonable because the correct path for a GPS trajectory or sub-trajectory is usually not the shortest distance path. Besides, a fixed similarity threshold cannot stand for the whole GPS trajectory when the measurement error level varies for some GPS readings. Moreover, how to deal with the skipped sub-segments is not discussed.

2.2 GPS Trajectory Segmentation

Travellers’ historical GPS trajectories are important indicators of their travelling behaviours, and GPS trajectory segmentation is a significant step for a further GPS trajectory application. Zheng (2015) gave a comprehensive overview on trajectory data mining, and provided a framework for trajectory data mining. The paper includes three different methods for trajectory segmentation in the trajectory preprocessing step. The first trajectory segmentation method is based on the time interval. If the time interval between
two consecutive sampling points is larger than a given threshold, a trajectory will be divided into two parts at the two points. The second method is based on the spatial shape of trajectories. The key points maintaining the shape of a trajectory are detected, and the trajectory is split at the key points. The third method is based on the semantic meaning of points. A trajectory is partitioned by semantic points such as stay points or transportation transit points. These methods can be used as primary guidelines for trajectory segmentation regarding to different applications.

### 2.2.1 Trajectory segmentation based on spatial shape

Many segmentation methods have been proposed based on the spatial shape and they partition GPS trajectory at characteristic points by maintaining the spatial structure of GPS trajectory. For instance, Douglas-Peucker algorithm (Douglas and Peucker 1973) is a classic method to partition trajectories to sub-trajectories with similar shape. It keeps the first and last GPS point of the trajectory and detects the GPS point which is the furthest from the line segment composed by the first and last GPS point. If the distance between furthest GPS point and the line segment is smaller than a given threshold, the segmentation is ended. Otherwise the GPS trajectory will be split at the furthest GPS point into two GPS sub-trajectories. Douglas-Peucker algorithm will continue for all GPS sub-trajectories until no segmentation happens in GPS sub-trajectories. Lee et al. (2007) partitioned and simplified GPS trajectories by using the minimum description length (MDL) principle to make a trade-off between preciseness and conciseness. Preciseness means the difference between trajectory and sub-trajectories should be as small as possible; conciseness means the number of sub-trajectories should be as small as possible. Moreover, the degree of
preciseness and conciseness are represented by $L(H)$ and $L(D|H)$ respectively, and the optimal segmentation is to minimize $L(H) + L(D|H)$. Compared with the two methods above (Douglas and Peucker 1973, Lee et al. 2007), Yoon and Shahabi (2008) proposed a trajectory segmentation method which takes into account both spatial and temporal structure in the trajectory data, which partitions a GPS trajectory into spatially and temporally homogeneous segments.

2.2.2 Trajectory segmentation based on semantic point

Different methods have been proposed for trajectory segmentation based on the semantic meaning of points. For instance, one semantic point based trajectory segmentation is to partition the trajectory into sub-trajectories with different transportation modes. Different strategies are proposed to partition GPS trajectories into sub-trajectories with distinct transportation modes. Top-down strategy is widely used to segment trajectory into trips. This strategy first detects the semantic points of a trajectory; then, it divides the trajectory into segments at these points; last, machine learning methods are applied to identify the transportation mode of each segment. For instance, Spaccapietra et al. (2008) proposed a stop-and-move-on-trajectory method which assumes people would stop at a location for an activity and then continue to move to the next location for another activity. Zheng et al. (2008b) proposed a walking-based segmentation method which assumes that people need to walk for transit between different transportation modes. However, the current top-down approaches have some deficiencies such as subjective and uncertainty in activity transitions in space and time. Das and Winter (2016a) provided a bottom-up segmentation strategy to overcome the deficiencies. They divided a trajectory into a set of
atomic kernels, and applied machine learning methods to identify the transportation mode on each atomic kernel; next, the adjacent atomic kernels with the same transportation mode were merged; last, the predicted trips were refined based on the general transit feed specification and other spatial information.

2.2.3 Trajectory segmentation based on time interval

The time interval method is another type of method for trajectory segmentation. If GPS signals in a trajectory are lost for a long time, this part of the trajectory would be uncertain in travel route. The uncertainty would affect many trajectory-based applications such as route planning. To avoid the uncertainty, time interval method suggests dividing the trajectories based on a time interval threshold. Previous studies have used different time interval thresholds, such as 120s (Bothe and Maat 2009) or 180s (Newson and Krum 2009), but none of them discussed how to determine the threshold. Moreover, the threshold should be data dependent or context-dependent, rather than a fixed value for various applications.

2.3 Recommendation methods

Recommender system is a hot research topic which analyzes the collected information to provide personalized recommendations to suit a user’s taste (Koren et al. 2009). The techniques in the recommender system can be divided into two groups, content-based recommendation and collaborative filtering recommendation. Content-based recommendation tries to recommend items similar to those a given user has liked in the past, while collaborative filtering recommendation identifies users whose tastes are similar
to those of the given user and recommends items they have liked (Balabanović and Shoham 1997).

### 2.3.1 Content-based method

Content-based method recommends an item to a user based on a description of the item and a profile of the user’s interests. Lops et al. (2011) stated that the content-based recommendation could be divided into three steps. First, structured relevant information is extracted from information source, and is used to represent data items from original information space to the target feature space. Second, a model of user interests is inferred starting from items liked or disliked in the past. Last, items are recommended to users by matching the user profile against the description of items.

#### 2.3.1.1 Item representation

Items should be represented with a few features or attributes and managed as structured data in the content-based recommender systems. Vector space model (VSM), originated from text retrieval, is widely used for item representation. VSM extracts the key features of items from original information and represents them as a vector. The values in the vector stand for the weights of the features. The weights can be determined by domain experts or some weighting methods. For instance, Movie2Go (Mukherjee et al. 2001) was a recommendation system for movies, and characterized movies with different features such as actors, genres, etc.
TF-IDF (term frequency-inverse document frequency) weighting is the common used weighting method associated with VSM. TF-IDF is also originated from text retrieval. The TF-IDF weight \( w(t, d) \) can be represented as follows,

\[
w(t, d) = \frac{tf(t, d) \times \log \frac{N}{n}}{\sqrt{\sum_{t_i} tf(t_i, d)^2 \times (\log \frac{N}{n})^2}}
\]

where \( tf(t, d) \) is the frequency of the term \( t \) in the document \( d \). \( N \) is the total number of the documents in the corpus, and \( n \) is the number of documents where the term \( t \) occurs at least once.

Salton and Buckley (1988) proposed TF-IDF weighting scheme is based on empirical observation: 1) term frequency factor should be used in the weighting system to measure the frequency of occurrence of terms in the document; 2) inverse document frequency prevents the terms with high frequency are retrieved prevalent in the whole collection of documents; 3) normalization implies the best terms should have high term frequencies but low overall term frequencies in the document.

In the content-based recommender system, items can be treated as documents and the features of the item can be taken as the terms in the document. For instance, Cantador et al. (2010) developed a social tagging system for recommendation. They considered tags of items as features and applied TF-IDF weighting method for calculating the weights of the tags.

2.3.1.2 User model

Learning user model is to predict whether the user would like or dislike an item, which could be achieved based on either explicit feedback or implicit feedback. The
explicit feedback is that user rates the items explicitly. For instance, like/dislike as binary rating scale, ratings as numerical rating scale and text comment. Implicit feedback indicates the relevance of a user on an item, such as bookmarking, browsing and discarding. Explicit feedback has little noise, but users make explicit feedback for only a small number of items. Contrarily, implicit feedback does not need direct user involvement, but it is difficult to understand whether the user real attitude to the items.

Based on users’ feedback, many methods have been proposed for user model. To predict if the user likes/dislikes a new item, nearest neighbor model compares the similarity between the new item and the items the user has rated. The rating value of the nearest neighbor will be assigned to the new item. Cosine similarity function is one of the widely used functions to compare the similarity between two neighbors in nearest neighbor model. The cosine similarity between two items $item_a$ and $item_b$ is calculated as,

$$Sim = \frac{\sum_i A_i \cdot B_i}{\sqrt{\sum_i A_i^2} \sqrt{\sum_i B_i^2}}$$

where $A$ and $B$ are vectors of features of $item_a$ and $item_b$, respectively. $A_i$ represents the $i$-th element in $A$.

Naïve Bayes model is another popular model to predict user’s rating on a new item. In naïve Bayes model, user’s ratings can be represented as $R = \{r_1, ..., r_m\}$, and a certain item $item_a$ can be represented with a feature vector $A = (a_1, ..., a_n)$. The naïve Bayes model intends to find the rating value $r$ on $item_a$ with the maximum probability $P(r|A)$, namely,

$$r = \arg \max_{r_i \in R} P(r_i|a_1, ..., a_n)$$

Based on Bayes rule,
\[ P(r_i | a_1, \ldots, a_n) = \frac{P(a_1, \ldots, a_n | r_i) P(r_i)}{P(a_1, \ldots, a_n)} \]

As \( P(a_1, \ldots, a_n) \) is equal for all rating values, thus it can be removed,

\[ r = \arg \max_{r_i \in R} P(a_1, \ldots, a_n | r_i) P(r_i) \]

Naïve Bayes model assumes that the features are conditionally independent from each other. Thus, the estimation of the rating value \( r \) can be represented as,

\[ r = \arg \max_{r_i \in R} \prod_{j=1}^{n} P(a_j | r_i) \]

Where \( P(a_j | r_i) \) is the probability that the item contains the feature \( a_j \) when the user gives the rating \( r_i \) on the item, and \( P(r_i) \) is the probability that the user gives the rating \( r_i \) in the history.

### 2.3.2 Collaborative filtering method

Out of the recommender systems, collaborative filtering (CF) is one of the most widely used recommendation techniques. Similar to the content-based method, the feedback can be explicit or implicit. Explicit feedback is user’s rating on items, such as on 1-5 scale; implicit information could be the history of purchases or click-throughs (Su and Khoshgoftaar 2009). Different from the content-based method, collaborative filtering method recommend items which similar users liked.

Generally speaking, collaborative filtering methods can be divided into two groups, memory-based method and model-based method. Memory-based CF method utilizes the entire user-item information to make a recommendation. User/item-based CF methods are memory-based methods. They calculate the similarities between users/items and take
similarities as weights to produce a prediction by taking the weighted average of all the ratings of users/items on the target user/item. However, the performance of memory-based CF methods will decrease greatly when they deal with the large-scale dataset. Model-based CF methods will learn the model and its parameters to predict the rating values, such as matrix factorization (Koren et al. 2009), random walk with restart (Konstas et al. 2009), aspect model (Hofmann 2004). Compared with memory-based CF methods, model-based CF methods usually have better performance in both efficiency and effectiveness, especially for the large-scale dataset. Several CF methods are discussed below.

2.3.2.1 User-based collaborative filtering

User-based collaborative filtering method (UF) calculates the similarity between the active user and other users, and takes the similarity value as the weights of other user’s influence on the active user. Next, UF predict the rating value of the active user on the new item by calculating the weighted sum of other users’ rating values on the new item.

There are two common methods for similarity measurement, one is cosine similarity mentioned in Section 2.3.1.2, and another one is Pearson correlation similarity. The Pearson correlation between two users \( u \) and \( v \) is calculated as follows,

\[
sim(u, v) = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}
\]

Where the \( i \in I \) summations are over the items that both the users \( u \) and \( v \) have rated and \( \bar{r}_u \) is the average rating of the co-rated items of the \( u \)th user.
After calculating the similarity between users, the predicted rating \( r_{a,i} \) of the user \( a \) on the certain item \( i \) is calculated as the weighted summation of other users’ ratings on the certain item \( i \),

\[
r_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot sim(u, a)}{\sum_{u \in U} sim(u, a)}
\]

Where \( \bar{r}_a \) is the average ratings of all items by user \( a \), \( \bar{r}_u \) is the average rating of all items by user \( u \). \( sim(u, a) \) is the similarity between users \( a \) and \( u \).

2.3.2.2 Item-based collaborative filtering

Item-based collaborative filtering predicts user’s rating on certain item by measuring the similarities between items. The Pearson similarity between two items \( i \) and \( j \) is calculated as,

\[
sim(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2 \cdot \sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}
\]

where \( \bar{r}_i \) and \( \bar{r}_j \) are the average ratings by all users on the items \( i \) and \( j \), respectively. Item-based collaborative filtering method predicts the rating value of the item \( a \) by the active user \( u \) by weighted summation of the rating values of all the items by the active user,

\[
r_{u,a} = \frac{\sum_{i \in I} r_{u,i} \cdot sim(a, i)}{\sum_{i \in I} sim(a, i)}
\]

Where \( r_{u,i} \) is the rating value of the user \( u \) on the item \( i \), and \( sim(a, i) \) is the similarity value between item \( a \) and \( i \).

It could be noticed that the nearest neighbor method in the content-based recommendation method is very similar to the item-based collaborative filtering method, as they both calculate the similarity between items. However, the difference between them lies at that
NN compares the similarity between items based on the features of items, while the item-based collaborative filtering method calculates the similarity between two items based on the ratings of users on the items in the history.

2.3.2.3 Matrix factorization

Matrix Factorization (MF) (Lee and Seung 2001, Koren et al. 2009) is one of the most important model-based CF methods in recommender system. MF is a latent factor model and assumes that each user and item is associated with a vector of latent factors. MF first constructs a user-item matrix based on users’ ratings, and then decomposes the user-item matrix into two low rank matrices representing latent factors. The product of corresponding user factors and item factors computes the prediction.

MF predicts user’s rating on certain item by \( \hat{r}_{u,i} = q_i^T p_u \), where \( p_u \) is the vector of latent factors associated with the user \( u \), and \( q_i \) is the vector of latent factors associated with the item \( i \). The parameters \( p_u \) and \( q_i \) are learned by minimizing the following objective function,

\[
L = \sum_{(u,i) \in S} (r_{u,i} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)
\]

where \( S \) is the set of the \( (u,i) \) pairs for which \( r_{u,i} \) is known. \( \|q_i\|^2 \) and \( \|p_u\|^2 \) are regularization items. \( \lambda \) controls the parameter of regularization and is determined by cross validation.

Stochastic gradient method (SGD) can be used to minimize the objective function. SGD modifies the parameters by a magnitude in the opposite direction of the gradient with the step size \( \gamma \) as follows,
\[ p_u = p_u + \gamma (r_{u,i} - q_i^T p_u) \cdot q_i - \lambda \cdot p_u \]
\[ q_i = q_i + \gamma (r_{u,i} - q_i^T p_u) \cdot p_u - \lambda \cdot q_i \]

2.3.2.4 Random walk with restart

Random walk with restart (RWR) is another popular model-based CF method (Jamali and Ester 2009, Konstas et al. 2009). RWR is a graph-based model and recommends items to a user with the user’s personalized ordering of node-to-node proximities which are measured by a random surfer in a user-item bipartite graph (Park et al. 2017). In the user-item bipartite graph, the nodes represent users and items, and the weights of edges represent the rating values between users and items, called RWR score. The RWR scores are defined as the following recursive function.

\[ s = (1 - c) \tilde{A}^T s + cv \]

where \( s \) is the vector of RWR scores for the user \( u \). \( A \) is the adjacency matrix between nodes in the graph, and \( \tilde{A} \) is row-normalized matrix of \( A \). \( c \) is the restarting probability that it may go back to the original node \( u \). \( v \) is the starting vector whose \( u \)-th entry is 1, and other entries are 0. The iteration will continue until the change of any entry in \( s \) does not exceed a given threshold. Thus, the rank on the item by the user can be predicted based on the RWR scores.

2.3.3 Summary for recommendation method

Compared with collaborative filtering, the content-based method does not need the data of other users, so users are not required to share their profile, which ensures user’s
privacy (Isinkaye et al. 2015). Besides, content-based method can provide new items even if no rating is provided by users. However, the content-based method required the rich description information of items before recommendation can be made to users, but it is very hard to extract the features from some items, such as graphical images and audio streams (Adomavicius and Tuzhilin, 2005). Besides, the content-based method is difficult to make recommendations to new users because user’s profile is hard to be understood before the user rated a lot of items.

The advantage of collaborative filtering method is that it does not require the content associated with the items and users. Moreover, CF method has the ability to recommend user the items which are relevant to the users but does not match the user’s profile. There are several disadvantages of CF method, 1) cold start problem. It refers to the situation that only has little or no information of user or item, which will deteriorate the performance of recommendation. 2) sparsity problem. It means that only a few of the total number of items are rated in the database, which will lead to difficulty in searching effective similar users and weak performance. 3) scalability. With the increase of data volume in the database, the efficiency of the recommendation will decrease.

### 2.4 Mobility Recommendation based on GPS Trajectories

Many studies have been conducted utilizing GPS trajectories to guide human mobility. Taxi GPS trajectory data have been widely studied in the provision of recommendations for taxi pick-up locations and routes. Lee et al. (2008) discovered the taxi pick-up pattern by clustering pick-up and drop-off locations of taxis based on k-means method, and recommended refined cluster locations for empty taxis. Song et al. (2014)
ranked a set of historical pick-up locations by the potential waiting time with walking penalty and provided a real time pick-up location recommendation for taxis. Li et al. (2011) studied the hotspots of taxi pick-up locations and predicted the strategies for taxi drivers. They defined strategies as “hunt” or “wait” before a pick up, and “local” or “distance” after a drop off. Next, they defined taxi-patterns as features which are composed by time, location and strategy. Last, SVM was used to select the top positive taxi-patterns. Li et al. (2012) discovered patterns of passenger pick-up quantity from urban hotspots, studied the urban mobility pattern from GPS trajectories, and predicted the number of passengers in a certain hotspot in the next time interval to help taxi drivers to find next passengers. Ge et al. (2010) estimated the probability of a pick-up event happening at each pick-up location and considered the driving distance when making route recommendations for taxi drivers such that the potential cruising distance before having a customer was minimized. Yang et al. (2015) proposed a method for finding the most profitable routes for taxis to pick up passengers based on taxi trajectories, and they made a trade-off between the pickup probability and capacity at a location.

More recently, studies have investigated location and travel recommendations based on human movement GPS trajectories. Zheng et al. (2010) proposed a user-centred CF method to make mobile recommendations for locations and activities based on historical trajectories. They adopted collaborative filtering method by constructing a three-dimensional tensor by user, location and activity. Besides, they explored the relationships among user, location and activity with user-location matrix, user-user matrix and activity-activity matrix, and integrated them with the tensor. Zheng and Xie (2011) inferred the interest level of a location and a user’s travel experience to give both generic and
personalized recommendations for interesting locations and travel sequences by mining multiple users’ GPS traces.

2.5 Route Recommendation

2.5.1 Traditional route recommendation

Traditional travel route planning provides a route between an origin and a destination according to a certain metric, such as the shortest distance or travelling time. Dijkstra’s algorithm (Dijkstra 1959) is the most well-known method for solving the shortest path problem. However, Dijkstra’s algorithm needs to be modified for some real travel applications, as it only works for the single metric in the road network. To address this problem, Delling et al. (2011) proposed a customizable route planning for real-time queries in terms of arbitrary metrics, including avoidance of U-turns and/or left turns.

Traditional travel route planning has other two limitations. First, some factors, such as road safety and traffic jams, are usually latent in travelling and difficult to access; hence, traditional route planning only considers limited factors and cannot make a comprehensive consideration of travel. Second, traditional route planning algorithms are not personalized, because they do not consider user’s travel preferences and provide only generic recommendations to the public.

2.5.2 Popular route recommendation using trajectories

Very few research works, however, have examined travel route recommendation using trajectories in the last decade.
Chen et al. (2011) proposed a method to discover popular routes from GPS trajectories. First, they distinguished intersections as nodes in a transfer network by clustering GPS points based on direction and density. 1) they considered that the density of GPS points around an intersection should be larger than a threshold because drivers would like to slow down at intersections; 2) they considered that GPS trajectories had a larger probability to change moving directions at intersections. Thus, a coherence score between two GPS points was proposed by calculating the distance and direction between the two GPS points. Next, DBSCAN was applied to group GPS points into clusters based on the coherence scores, and each cluster was identified as a node in the transfer graph.

Then, a turning probability $Pr_{d}(n_{i} \rightarrow n_{j})$ is calculated between two nodes $n_{i} \rightarrow n_{j}$ by considering the number of trajectories passing through the starting node $n_{i}$ and the distance between the trajectories and the destination $d$. With the turning probability between nodes in the transfer graph, they considered the moving in the transfer graph as a random walk model, and calculated the transfer probability $Pr^{t}(n_{i} \rightarrow d)$ from a node $n_{i}$ to the destination $d$ within $t$ steps. Last, the popularity of the node $n_{i}$ with respective to the destination $d$ is set as the transfer probability, and the most popular route is defined as the route along which the product of the popularity of the nodes is maximized.

Besides, Zheng et al. (2012) and Wei et al. (2012) proposed two methods to find the top-$k$ possible popular routes from given two points based on historical trajectories.

In Zheng et al. (2012), they discovered the top-$k$ possible routes from low sampling GPS trajectories and developed the research based on two observations, 1) although the possible routes between certain locations are usually enormous, only a few of them are
travelled often; 2) similar trajectories can often complement each other to make themselves more complete.

They first extracted reference trajectories with respective to origin point $p_i$ and destination point $p_{i+1}$ based on the criterion, 1) the distance between $p_i, p_{i+1}$ and the trajectory $T$ should be smaller than a given radius, respectively; 2) for any point $a$ in the trajectory $T$, it should satisfy $d(p_i, a) + d(p_{i+1}, a) < (p_{i+1}, t - p_i, t) \cdot v_{max}$, $v_{max}$ is maximum allowed speed in the road network. The GPS points in reference trajectories were called reference points. With the reference trajectories with respective to $\langle p_i, p_{i+1} \rangle$, two local approaches were proposed. One local method was traverse graph-based approach, and this approach built a traverse graph. In this graph, the vertices, called traverse edges, were the road segments travelled by the reference trajectories; the links in the graph were the edges connecting the traverse edges in the road network. In the traverse graph, the k-shortest paths were identified as the k-top possible routes. Another local method was called nearest neighbour-based approach and this approach started from $p_i$, and searched its k nearest reference point as the new starting point till the $p_{i+1}$ was reached. Besides, a global method was proposed to connect consecutive local routes to obtain the global route for the whole query. To search the global route, a global route score was defined by integrating the popularity of the local route and the transition confidence between two connecting local routes, and the top-k global routes with the maximum global route scores were calculated.

In Wei et al. (2012), an algorithm was proposed to construct popular routes from historical trajectories without road networks. This algorithm first divided the whole study area into cells, and merged cells into regions based on the connection of cells derived from historical trajectories; next, a routable graph was generated by inferring the edges between
the regions; last, the optimal score of a route was defined and the top-k popular routes were calculated with the maximum scores in the routable graph.

All the algorithms mentioned above perform well to find the popular routes from users’ historical trajectories. However, neither of them study users’ personal route preferences, so the discovered popular routes are not personalized.

2.5.3 Personalized route recommendation using GPS trajectories

McGinty and Smyth (2000) developed a personalized route planning service called TURAS which utilized a case-based approach to make route recommendations based on GPS traces. TURAS had four components including digital map, user profile, route planner and profile manager. User profile was the route travelled by the user with an assigned grade by the user. Route planner provided the personalized route for particular target user. Profile manager updated user profile if user gave a grade to a new generated route by the system. The route planner was a two-step process. The first step searched for appropriate cases from the profile of the target user to fit the current problem; For example, to recommend a route between two points X and Y, if there was no trajectory travelling from X to Y, it would try to find the trajectory T so that the distance \( d(X, X_1) + d(Y, Y_1) \) is minimum, and \( X_1 \) and \( Y_1 \) are the GPS points in the trajectory T. Then, the route planning problem between X and Y became two sub-routing problems between X and \( X_1 \), and between Y and \( Y_1 \), respectively. The second step involved the use of A* algorithms to compute the remaining path if the distance between the start and end point in the sub-routing problem is smaller than a threshold.
McGinty and Smyth (2001) improved their case-based approach and made use of a collaborative case-based reasoning (CCBR) structure to undertake personalized route recommendation by sharing the knowledge between agents. CCBR first checked if the routing problem of the target user could be solved based on his/her cases or not. If not, CCBR broadcast the routing problem to other agents, and selected the solution with the best quality from other agents. The quality of the solution by the other agent with respective to the routing problem depended on the coverage of the solution that the agent provided to the routing problem and the similarity between the agent and the target user. The disadvantage of CCBR is that the recommendation was always static with different query time, which is unreasonable because users’ travel preferences and the travel environment may vary with time. Besides, the setting of the weight to balance the coverage and the similarity was not solved.

Letchner et al. (2006) indicated that the assumption about constancy and universality of route recommendation is poor, and that user’s preference might vary with time. They proposed the personalized route recommendation based on the traffic condition and the preferences of individual users. They divided one day into 96 time slices and calculated the speed on each road segment by average. For the road segment and time slice where no data was available, the speed on the road segment at the adjacent time slice was used. Besides, they proposed an inefficient ratio of routes for each driver based on the historical routes. The inefficient ratio was determined with the ratio of the duration of the fastest route between route endpoints, relying upon the estimated speeds on road segments, and the actual duration of user’s trip. The inefficient ratio measures the extent that user would like to extend the duration of the trip beyond the shortest time in order to satisfy
preferences unrelated to the efficiency. Last, the ratio was taken as a metric in an A* algorithm for a personalized route recommendation.

Dai et al. (2015) constructed a driving preference vector to describe users’ preference and searched the reference trajectories from users having similar preference for route recommendation. The proposed personalized route recommendation made analysis of different types of costs (such as travel time, fuel consumption or other costs) and took the ratios between different costs as the random variables to construct the driving preference vector. As the collection of preference ratio values could be obtained from the driver’s all trajectories, the distributions of the preference ratio values were estimated for each user. Thus, each driver was associated with one driving preference vector, and each preference ratio in the vector corresponds to a probability distribution.

With the driving preference vector, the satisfaction of a trajectory for a driver could be calculated with the proposed personalized satisfaction score function. The function made the integral with respective to each ratio in the vector in a small neighbour of the ratio value of the trajectory and added up the integrals for all ratios as the satisfaction score. With the query time, origin vertex, destination vertex and the user’s driving preference vector, the reference trajectories were the top-k satisfaction scores from the database, and these reference trajectories should pass through the origin and destination vertex and close to the query time.

Two route recommendation methods, the local route recommendation method and global route recommendation method, were proposed. The local route recommendation method extracted the road edges and vertices traversed by the reference trajectories to construct the local reference graph. The weights of edges in the local reference graph were
estimated with random walk algorithm, and the shortest path in the local reference graph was recommended to the driver. However, the local route recommendation method would not work if the set of reference trajectories with respective to the origin and destination vertex was empty. Thus, the global route recommendation method was proposed in this scenario. The global route recommendation method first divided the routing problem into several stages and identified transfer edges in each stage by a sweep and expand process. Then, the sequence of the transfer edges with the maximum probability was calculated based on the hidden Markov model. Next, local reference graphs were constructed for each adjacent transfer edges in the sequence, and the union of local reference graphs was regarded as the global reference graph. Last, random walk was used to estimate the weights on each edge in the global reference graph and the shortest route was recommended to the driver.

However, the features in the driving preference vector required by the algorithm can hardly be extracted from data, and should be defined with domain knowledge for the driving area. Besides, the driving preference vector is invariant to the time, which is unreasonable because user’s driving preference may vary with time. Furthermore, the reference trajectories are retrieved by comparing the similarity of features between users and trajectories, rather than between users. When the user travels to the different regions where the user’s driving preference vector can not reflect his/her preference, the obtained reference trajectories are ineffective. Moreover, the local recommendation method is a case-based method which may ignore the trajectories not passing the origin and destination, but they might be useful for the route recommendation.
Route recommendation by learning human activities from GPS trajectories is a promising area to improve route recommendation quality. Das and Winter (2016b) connected human activities to the semantic trajectories, and proposed an activity model on trajectories at different contexts, which helps better understand human activities based on their GPS trajectories. Wang et al. (2016) made a ridesharing planning from the perspective of user’s activities, which significantly enlarge the matching chance by considering user’s activities rather than the locations, and will be useful for route recommendation based on potential activities learned from user’s historical GPS trajectories.

2.6 Summary

In summary, this section reviews several research areas related to the thesis. First, GPS trajectory map matching is an important preprocessing step for applications based on GPS trajectories. However, most of current map matching methods are point-based methods which lead to inefficiency and low accuracy. Second, GPS trajectory segmentation, another important preprocessing step, is discussed. Different GPS trajectory segmentation methods are introduced and applied for distinct motivations, but none of them is proposed for map matching. Third, the recommendation techniques are reviewed and can be divided into two categories, content-based method and collaborative filtering method. Content-based method recommends the user the item which they share the similar features. Collaborative filtering method recommends user the item which the similar user may prefer, and they can deal with the issues of personalized travel route recommendations by considering users’ preferences that can be learned from their GPS trajectories. Fourth, GPS trajectories have been applied in many areas to guide human mobility, such as taxi
pick up route recommendation and activities recommendation for users. Last, the route recommendation methods are discussed and divided into three areas, the traditional route recommendation, the popular route recommendation and the personalized route recommendation. The traditional route recommendation methods can provide the customizable route recommendation to users. However, they only consider the limited factors so that it is very difficult to reflect user’s real travel preferences. The popular route recommendation methods discover the popular routes from historical trajectories, but they do not consider user’s preferences and are not personalized. Several personalized route recommendation methods are proposed by learning users’ historical GPS trajectories, and they have different disadvantages and should be improved.
Chapter 3: Segment-based hidden Markov model for map matching

This chapter proposes a segment-based hidden Markov model for map matching. Chapter 3.1 introduces the problem of the current map matching. Chapter 3.2 gives some definitions related to GPS trajectory and map matching. Chapter 3.3 discusses the proposed segment-based map matching method. Chapter 3.4 conducts experiments to give the evaluation for the proposed method. Chapter 3.5 makes a summary about the segment-based hidden Markov model for map matching.

3.1 Map Matching Problem

GPS trajectory is a sequence of GPS readings which can record the spatial track of moving objects effectively. With the popularity of the mobile devices, the massive GPS trajectory data is available and extensively used in various areas. However, as the spatial position of GPS trajectory is usually imprecise due to the measurement error and low sampling rates of GPS receivers, it requires to match GPS trajectory onto the road network during the preprocessing step for many applications, such as urban mobility computing, route navigation, transportation analysis and management, etc. Thus, map matching is highly demanded.

Map matching is an active research area these days. Most of the current map matching methods are point-based method which deals with GPS points individually. Generally, the point-based map matching first matches individual GPS points onto the candidate road segments by utilizing spatial features (such as geometry and topology) and non-spatial features (such as speed) of individual GPS points, and then searches a reasonable path to connect the candidate road segments in the road network. The advantage
of the point-based map matching is that the distance measurement between GPS point and its surrounding road segment is straightforward.

However, the point-based map matching methods usually have two deficiencies. First, the point-based map matching method is sensitive to the measurement errors of GPS points. For example, when matching individual GPS point onto the dense road network, the point-based method is likely to locate some GPS point onto the incorrect road segment because the GPS point may be closer to the incorrect road segment than the correct road segment due to the measurement error. Since the final map matching depends on individual GPS point matching, therefore, unnecessary detours are shown in the map matching results. Figure 3.1 illustrates the problem of the point-based map matching using a real taxi GPS trajectory.

In the figure, the taxi GPS trajectory and its true route is represented in black circle dots and a solid gray line, respectively. For this example, the widely used point-based hidden Markov model (Newson and Krumm, 2009), which is known as the best algorithm for map matching (Goh et al. 2012, Oran and Jaillet 2013, Li et al. 2015), is applied for this trajectory and produces the route shown in cross dots after map matching. The ellipses in the figure exhibit two matching errors which can be observed in Figure 3.1(a) and Figure 3.1(b), respectively. In Figure 3.1(a), the point-based method matched \( p_2 \) to an incorrect road which results in a mistake detour. In Figure 3.1(b), a part of GPS trajectory from \( p_2 \) to \( p_4 \) is matched onto a local way but the real route shows that they kept moving on the highway. Both incorrect matching is caused as some GPS points are closer to the road segments in the detour routes.
Furthermore, the point-based method, processing GPS point individually, is inefficient. Each trajectory usually includes hundreds of points, especially with the frequent sampling rates of the GPS receiver. Many adjacent GPS points have similar spatial
relationship with the road network, but the point-based map matching method treats individual GPS point separately and ignores the consecutive relations among the GPS points, which leads to large computation cost. For example, some point-based methods (Newson and Krumm 2009; Lou et al. 2009) consider the connectivity between the candidate road segments of adjacent GPS points in the road network. The shortest distance path calculation between every pair of candidate road segments of adjacent GPS points leads to high computation cost. If the consecutive GPS points with similar attribute can be processed together, the computation cost for the shortest distance paths between adjacent GPS points can be largely reduced.

To overcome the limitations of the point-based map matching methods, this thesis tends to approach the map matching problem from a segment-based perspective. Rather than considering each individual GPS point, the segment-based map matching method partitions a GPS trajectory into several GPS sub-trajectories where GPS points have similar attribute, and takes GPS sub-trajectory as the processing unit in map matching. The segment-based strategy compares the distance between GPS sub-trajectories and road segment sequences, and considers the connectivity between adjacent GPS sub-trajectories based on the topology of the road network, which can reduce the computation cost and alleviate the negative influence caused by mismatching of individual GPS points due to measurement error. For instance, the segment-based map matching can integrate GPS points from $p_1$ to $p_4$ in Figure 3.1(a) as one GPS sub-trajectory and then match it onto the road network. Similarly, GPS points from $p_1$ to $p_5$ can be integrated into one GPS sub-trajectory in Figure 3.1(b).
However, there are still some challenges for the segment-based map matching method. First, trajectory segmentation is crucial for the segment-based map matching. Previous trajectory segmentation methods are developed for different applications such as discovering interesting sub-path (Zhou et al. 2011), detecting flocking behavior (Buchin et al. 2010) and trajectory annotation (Das and Winter, 2016), but none of them is suitable for the map matching. Different trajectory segmentation methods produce GPS sub-trajectories with distinct characteristics, and to explore an effective trajectory segmentation method on the segment-based map matching is important. Second, GPS sub-trajectory corresponds to a path in the road network, so it needs to be matched onto a sequence of road segments. However, the number of road segment sequences close to a GPS sub-trajectory could be overwhelming. Thus, how to search the candidate road segment sequences and filter out unreasonable ones efficiently is unsolved. Third, it is important for map matching methods to measure the distance between GPS sub-trajectory and road segment sequences effectively. Many methods have been proposed for measuring the distance between trajectories, such as longest common subsequence (LCS) (Vlachos et al. 2002), Fréchet distance (Alt and Godau 1995), edit distance on real sequence (ERP) (Chen and Raymond NG 2004), dynamic time warping (DTW) (Keogh and Ratanamahatana 2005) and edit distance with real penalty (EDR) (Chen et al. 2005). However, LCS, ERP, DTW and EDR measure the distance between two trajectories by calculating the distance between pairs of points in the trajectories, so they do not have a good performance when measuring the distance between GPS sub-trajectory and road segment sequence because GPS point may correspond to certain point in the middle of the road segment, rather than the end points of road segments. Fréchet distance can be used to measure the distance
between GPS sub-trajectory and road segment sequence directly, but it is very sensitive to the measurement error of GPS point. Thus, to explore an effective method to measure the distance between GPS sub-trajectory and road segment sequence is important. Last, how to locate GPS sub-trajectories onto road networks to achieve a global optimum map matching is worthy studying. It should comprehensively consider both the distance between GPS sub-trajectory and candidate road segment sequence, and the reasonable accessible path between adjacent GPS sub-trajectories in the road network. Therefore, an effective model for the segment-based map matching is crucial.

3.2 Definitions

This section introduces some basic concepts for the segment-based map matching discussion.

**Definition 3.1 Road network.** The road network is a directed graph, \( G = (V, E) \), where \( V \) is a set of vertices representing the terminal points of road segments, and \( E \) is a set of directed edges representing the road segments. Vertex \( v_i \in V \) is a terminal point of road segments. An edge \( e_j \in E \) is a road segment with a starting point \( e_j \text{.start} \) and an end point \( e_j \text{.end} \), where \( e_j \text{.start} \in V \) and \( e_j \text{.end} \in V \).

**Definition 3.2 Road segment sequence (RSS).** The road segment sequence is a sequence of the adjacent road segments, denoted as \( rss = \{e_1, e_2, ..., e_m\} \), where \( e_k \) is a road segment and \( e_k \text{.end} = e_{k+1} \text{.start} \) (\( 1 \leq k \leq m \)). The first and last vertex of \( rss \) is denoted as \( rss \text{.first} \) and \( rss \text{.last} \), and \( rss \text{.first} = e_1 \text{.start} \) and \( rss \text{.last} = e_m \text{.end} \). Figure 3.2 gives an example of two RSSs. In Figure 3.2, two RSSs, \( rss_1 \) and \( rss_2 \), can be
constructed by three road segments $e_1, e_2$ and $e_3$. To be more specific, $rss_1 = \{e_1, e_2\}$, and $rss_2 = \{e_1, e_3\}$. In Figure 3.2, $rss_1. first = v_1$ and $rss_1. last = v_4$.

**Figure 3.2 Road segment sequence**

**Definition 3.3 GPS-reading.** A GPS-reading $p$ is a 3-tuple denoted as: $p = (t, lat, lng)$, where $t$ is the timestamp of the GPS-reading, and $lat$ and $lng$ are the latitude and longitude of the location of the GPS-reading at time $t$.

**Definition 3.4 GPS trajectory and GPS sub-trajectory (STRJ).** A GPS trajectory is a sequence of GPS-readings $trj = \{p_1, p_2, p_3, ..., p_m\}$ where $p_i. t - p_{i-1}. t > 0, 1 < i \leq m$. A GPS sub-trajectory $strj = \{p_k, p_{k+1}, p_{k+2}, ..., p_l\}$ is a contiguous subset of a GPS trajectory $trj = \{p_1, p_2, p_3, ..., p_m\}$ if $1 \leq k < l \leq m$ and $strj. first = p_k$, $strj. last = p_l$.

A GPS trajectory can be divided into a sequence of GPS sub-trajectories $strj_1, strj_2, ..., strj_n$ after trajectory segmentation, and the $k$-th GPS sub-trajectory is represented as $strj_k$.

**Definition 3.5 Heading.** The heading generated by two GPS readings ($p_i$ and $p_j$) is defined as clockwise in degrees from 0 (due north) to 360 (again due north, coming full
circle), denoted as $\theta_{i,j}$. The value of the heading indicates the orientation of the moving between the two GPS readings.

The angle difference $D(\theta, \theta')$ between two headings $\theta$ and $\theta'$, varying from 0 to 180 degrees, is defined as:

$$D(\theta, \theta') = \begin{cases} 
|\theta - \theta'|, & \text{if } |\theta - \theta'| \leq 180 \\
360 - |\theta - \theta'|, & \text{if } |\theta - \theta'| > 180 
\end{cases}$$

**Definition 3.6 Heading of GPS sub-trajectory.** Given a GPS sub-trajectory $strj = \{p_1, p_2, p_3, ..., p_l\}$, the heading of the sub-trajectory is denoted as $\theta_{strj} = median(\theta_{1,2}, \theta_{2,3}, ..., \theta_{l-1,l})$.

3.3 Segment-based Hidden Markov Model

3.3.1 Overview of SHMM

In this chapter, the segment-based hidden Markov model, called SHMM, is proposed for the map matching. Figure 3.3 shows the framework of SHMM which consists of three steps. The first step of SHMM is the data preprocessing where the outliers of GPS trajectories are removed, GPS trajectories are sampled, and an R-tree is built for the road network. In the second step, the GPS trajectory is partitioned into sub-trajectories based on heading homogeneity with an incremental trajectory segmentation method, and then candidate RSSs are searched for each GPS sub-trajectory. The last step is to build hidden Markov model based on GPS sub-trajectories and candidate RSSs. A distance metric, called LCS-HP, is proposed to measure the distance between GPS sub-trajectories and candidate RSSs, and the LCS-HP distance is used to calculate the emission probability in hidden Markov model. Besides, the transition probability is calculated based on the difference between the distance of adjacent GPS sub-trajectories and the distance of
adjacent RSSs. The sequence of RSSs with the maximum probability is calculated in the hidden Markov model with Viterbi algorithm (Viterbi, 1967). In the following, each step will be discussed in detail.

Figure 3.3 The framework of SHMM
3.3.2 Data preprocessing

Data preprocessing of SHMM is conducted in two parts. The first task is the outlier removal of GPS trajectories. As mentioned, the outliers in GPS trajectories are the GPS points which positions greatly deviate from their real locations, so the outliers severely hamper the performance of map matching. SHMM identifies outliers based on spatial proximity and speed constraint. First, it is assumed that if a GPS point is far from any road segment, it will be unlikely to be matched onto the road networks. Therefore, given a distance threshold \( r \), a GPS point is considered as outlier if there exist no road segment nearby within the radius \( r \). An example about spatial proximity is shown in Figure 3.4(a). In this figure, \( p_3 \) is removed from the trajectory since no road segment can be found within the radius \( r \). However, some other outliers can not be removed based on spatial proximity. In this case, a speed constraint limit \( \omega \) is used to discriminate these outliers. If the speed of a GPS point exceeds \( \omega \), it will be considered as an outlier. For example, Figure 3.4(b), the GPS point at \( p_3 \) is located close to a road segment, but it will be identified as an outlier because it is impossible for this object moving from \( p_2 \) to \( p_3 \) within the time interval based on the speed constraint. The parameter \( r \) and \( \omega \) are set as 40 meters and 150 km/h, respectively.
3.3.3 Trajectory segmentation and candidate road segment sequence search

Rather than dealing with each individual GPS point, in SHMM, each GPS trajectory is divided into several GPS sub-trajectories, and then GPS sub-trajectories are matched onto potential RSSs. However, as the number of the road segments around each GPS reading point could be large especially in the dense road networks, the number of candidate RSSs composed by the combinations of these road segments would be huge. To reduce the number of candidate RSSs for each GPS sub-trajectory, this thesis proposes trajectory segmentation method based on heading homogeneity. The heading attribute of the GPS sub-trajectory helps in filtering out unreasonable candidate RSSs.
3.3.3.1 Trajectory segmentation with heading homogeneity

The trajectory segmentation based on heading homogeneity (TSHH) divides a GPS trajectory into GPS sub-trajectories associated with a heading attribute so that the consecutive GPS points in each GPS sub-trajectory have similar heading. Given a GPS trajectory \( trj = \{p_1, p_2, p_3, ..., p_m \} \) and a threshold \( \gamma \), a GPS sub-trajectory \( strj = \{p_i, p_{i+1}, p_{i+2}, ..., p_j \} \) \( 1 \leq i < j \leq m \) generated by TSHH have the following characteristics:

1) For any adjacent GPS readings \( p_k, p_{k+1} \) \( (i \leq k < j - 1) \) and \( p_{k+1}, p_{k+2} \) in \( strj \), the angle between their headings \( \theta_{k,k+1} \) and \( \theta_{k+1,k+2} \) cannot exceed the threshold \( \gamma \), namely \( D(\theta_{k,k+1}, \theta_{k+1,k+2}) < \gamma \).

2) For any adjacent GPS points \( p_k, p_{k+1} \) in \( strj \), the angle between the heading \( \theta_{k,k+1} \) and the heading \( \theta_{i,j} \) of the first and last GPS point \( p_i, p_j \) cannot exceed the threshold \( \gamma \), namely \( D(\theta_{k,k+1}, \theta_{i,j}) < \gamma \) where \( (i \leq k < j - 1) \).

The GPS sub-trajectory generated by TSHH that satisfies the heading homogeneity condition is called a STRJ-HH condition, and the value of \( \gamma \) is set as 20 degrees in the thesis.

To generate GPS sub-trajectories which satisfy the STRJ-HH condition, an incremental method is developed for trajectory segmentation. The pseudocode of TSHH is shown in Figure 3.5. The proposed incremental method of TSHH first creates a new GPS sub-trajectory \( strj \) from a starting GPS reading \( p_i \), and adds the starting GPS reading \( p_i \) and the next GPS reading \( p_{i+1} \) into \( strj = \{p_i, p_{i+1}\} \) (line 1-3). Then, it considers the next GPS reading \( p_{i+2} \) in \( trj \) as the target GPS reading point, if the target GPS reading point is
the last GPS reading $p_n$ in $trj$, the trajectory segmentation is finished, and the algorithm stops; otherwise it checks whether the target GPS point satisfies the STRJ-HH condition for the current sub-trajectory $strj$. Next, if the target GPS point satisfies the STRJ-HH condition, add the GPS reading point into the current sub-trajectory $strj$ (line 5-10); otherwise stop the growth of $strj$, and create a new sub-trajectory by taking the last GPS point of $strj$ as the starting GPS point (line 11-14).

---

**Algorithm: TSHH**

**Input:** GPS trajectory $trj = \{p_1, p_2, ..., p_m\}$; heading change threshold $\gamma$.

**Output:** the set of GPS sub-trajectories $strj_set$

1. $start_pid = 0, current_pid = 2$
2. $strj = CreateNewSubTrajectory(start_pid)$
3. **while** ($start_pid \neq m - 1$)
4.   **if** ($STRJ_HH(strj, current_pid, \gamma) == true$)
5.     $strj.Add(current_pid + 1)$
6.     **if** ($current_pid == m + 1$)
7.     $strj_set.Add(strj)$
8.     **Break**;
9.   **end if**
10. **else**
11.   $strj_set.Add(strj)$
12. $strj = CreateNewSubTrajectory(current_pid - 1)$
13. **end if**
14. **end while**

---

**Figure 3.5 Pseudocode of TSHH**

TSHH guarantees that GPS points in GPS sub-trajectory satisfy STRJ-HH condition, and the heading of GPS sub-trajectory is calculated based on Definition 3.6.
Figure 3.6 Trajectory segmentation based on heading homogeneity

Figure 3.6 gives an example of trajectory segmentation for \( trj = \{ p_1, p_2, \ldots, p_7 \} \) based on TSHH. The algorithm starts a GPS sub-trajectory \( strj_1 \) from the first GPS point \( p_1 \), then integrates \( p_1 \) and \( p_2 \) into \( strj_1 \) and checks the next GPS point \( p_3 \). As the difference among \( \theta_{1,3}, \theta_{1,2} \) and \( \theta_{2,3} \) does not exceed the given threshold \( \gamma \) (setting as 20
degrees in this case), \( p_3 \) will be integrated into \( strj_1 \). Similarly, \( p_4 \) will be integrated into \( strj_1 \). When it comes to \( p_5 \), as the difference between \( \theta_{1,5} \) and \( \theta_{4,5} \) exceeds \( \gamma \). The GPS sub-trajectory \( strj_1 \) ends and a new GPS sub-trajectory \( strj_2 \) starts from \( p_4 \) which is the last GPS point in \( strj_1 \). After the GPS trajectory is processed by TSHH, two GPS sub-trajectories are generated \( strj_1 = \{ p_1, p_2, p_3, p_4 \} \) and \( strj_2 = \{ p_4, p_5, p_6, p_7 \} \).

3.3.3.2 Candidate RSS search

The next task of SHMM is to match the generated GPS sub-trajectories onto the potential RSSs. Thus, candidate RSSs for each GPS sub-trajectory should be searched. The candidate RSS search can be divided into two steps: (1) road segment query for each GPS point and (2) RSS creation by connecting road segments.

In the first step, the road segments within the search radius \( r \) of the individual GPS point will be searched out; in the second step, RSS is created by searching a path connecting the road segments of every adjacent GPS points in the GPS sub-trajectory.

However, as a large amount of road segments could be found for a GPS point in the dense road network, it may lead to large computation for candidate RSS search. Therefore, unreasonable road segments should be filtered out. As each GPS sub-trajectory has a heading after TSHH, the heading of GPS sub-trajectory can be used to filter out the road segments with the unreasonable heading. If the angle between the heading of GPS sub-trajectory and the heading of road segment exceeds a threshold \( \beta \), the road segment will be filtered out. In this thesis, the threshold \( \beta \) is set as \( 90^\circ \).
Figure 3.7 Example of candidate RSS search

Figure 3.7 gives an illustration of candidate road segment query. In this figure, a GPS sub-trajectory is composed by three GPS points \( strj = \{p_1, p_2, p_3\} \). Besides, there are four road lanes located around the GPS sub-trajectory, two lanes from east to west and two lanes in the opposite direction. Based on R-tree, the road segments within the distance \( r \) of \( p_1 \) are \( e_4, e_6, e_7, e_9, e_{10}, e_{12} \). The road segments within \( r \) for \( p_2 \) and \( p_3 \) are \( e_7, e_{10} \) and \( e_5, e_7, e_8, e_{10}, e_{11}, e_{13} \). With heading constraint, \( e_9, e_{10}, e_{11}, e_{12}, e_{13} \) are filtered out since their headings are greatly deviant from the heading of \( strj \). After filtering with heading, the road segments for \( p_1 \) is \( e_4, e_6, e_7 \), for \( p_2 \) and \( p_3 \) are \( e_7 \) and \( e_5, e_7, e_8 \), respectively.

Next, the depth-first search is applied to search the RSS by connecting the candidate road segments of consecutive GPS points. It starts from a road segment of a GPS reading, until one road segment of the next GPS reading is reached. The RSS search from the first GPS point to the last GPS point creates the candidate RSS for a GPS sub-trajectory. Given a GPS sub-trajectory \( strj = \{p_1, p_2, ..., p_n\} \), the pseudo-code of candidate RSS search is shown in Figure 3.8.
**Algorithm: Candidate RSS Search**

**Input:** GPS sub-trajectory \( strj = \{p_0, p_1, ..., p_n\} \); road networks \( G = (V, E) \); search radius \( r \); heading change threshold \( \beta \).

**Output:** the set of RSSs \( rss \)

1. For \( i = 0; i \leq n; i + + \) 
2. \( rs_i = \text{SearchCandidateRoadSegment}(p_i, G, r, \beta) \) 
3. End For 
4. For \( i = 0; i \leq n; i + + \) 
5. \( rss = \text{Connect}(rss, rs_i, G) \) 
6. End For

**Figure 3.8 Pseudocode of candidate road segment search**

In Figure 3.8, the road segments within the distance \( r \) of each GPS point are calculated out in line 1-3. Next, the potential RSSs between the candidate road segments of consecutive GPS points in the GPS sub-trajectory are searched out with the depth-first search in line 4-6.

### 3.3.4 LCS-HP Distance measurement

To check if a GPS sub-trajectory \( strj \) can be matched onto a reasonable road segment sequence \( rss \), the distance between \( strj \) and \( rss \) should be measured. The smaller distance between \( strj \) and \( rss \), the larger probability they can be matched. GPS sub-trajectory can be taken a trajectory composed by a sequence of points, and RSS as a trajectory composed by a sequence of road edges. Longest Common Subsequence (LCS) can measure the distance between two trajectories by finding the longest subsequences where elements are matched in two trajectories and allowing some elements unmatched. However, LCS should be improved for measuring the distance between GPS sub-trajectory and RSS due to the following reasons.
1. LCS only allows each element in the trajectory be matched once (Vlachos et al. 2002, Zhu et al. 2017), while consecutive GPS points in GPS sub-trajectory might be located on the same road segment in RSS. Therefore, a road segment in RSS should be allowed to match with multiple consecutive GPS points, rather than only one GPS point.

2. Previous research assigns a fixed similarity value to the pair of elements which are matched (Vlachos et al. 2002). In this research, the similarity between the matched GPS point and road segment is determined by the Euclidean distance from the GPS point to the road segment. The further Euclidean distance between GPS point and road segment, the smaller similarity between them. Different from the similarity only considering distance (Zhu et al. 2017), the heading similarity should be also considered when matching GPS point and road segment in this research.

3. As GPS sub-trajectory moves in the similar heading, this thesis assumes that the corresponding RSS should be unlikely a detour, which means that the length of the matched GPS points and the length of the corresponding road segments should be similar.

In this thesis, an improved version of LCS, called Longest Common Subsequence with Heading Penalty (LCS-HP), is proposed to measure the distance between GPS sub-trajectory and RSS. Given a \( strj = \{p_1, p_2, ..., p_m\} \) and a \( rss = \{e_1, e_2, ..., e_n\} \), the similarity between the GPS point \( p_i \) and road segment \( e_j \) is defined as,

\[
Sim(p_i, e_j) = \begin{cases} 
0, & \text{Dist}(p_i, e_j) > \varepsilon_d \text{ and } D\left(\theta_{strj}, \theta_{e_j}\right) > \varepsilon_h \\
1 - \frac{\text{Dist}(p_i, e_j)}{\varepsilon_d}, & \text{otherwise}
\end{cases}
\]
where \( \text{Dist}(p_i, e_j) \) is the Euclidean distance between \( p_i \) and its closest point on \( e_j \), \( \theta_{strj} \) and \( \theta_{e_j} \) are the headings of the GPS sub-trajectory and the road segment \( e_j \), respectively. \( \varepsilon_d \) is the threshold for distance between \( p_i \) and \( e_j \), and \( \varepsilon_h \) is the threshold for the heading difference between \( p_i \) and \( e_j \). It is defined that \( p_i \) and \( e_j \) can be matched if \( \text{Sim}(p_i, e_j) > 0 \).

If the subsequence of \( strj \) from the first GPS point to the \( i \)-th GPS point is defined as \( strj(i) = \{p_1, p_2, ..., p_i\}, 1 \leq i \leq m \) and the subsequence of \( rss \) from the first road segment to the \( j \)-th road segment as \( rss(j) = \{e_1, e_2, ..., e_j\}, 1 \leq j \leq n \). The LCS value \( L(strj(i), rss(j)) \) between \( strj(i) \) and \( rss(j) \) is defined as,

\[
L(strj(i), rss(j)) = \max\{ L(strj(i - 1), rss(j)), L(strj(i), rss(j - 1)), \\
L(strj(i - 1), rss(j - 1)) + \text{Sim}(p_i, e_j), \\
L(strj(i - 1), rss(j)) + \text{Sim}(p_i, e_j) \} \tag{3.1}
\]

This equation is based on the dynamic programming and represents four situations for LCS value calculation. If the last point \( p_i \) in \( strj(i) \) and the last road segment \( e_j \) in \( rss(j) \) are not matched, \( L(strj(i), rss(j)) \) must be the larger value between \( L(strj(i - 1), rss(j)) \) and \( L(strj(i), rss(j - 1)) \). If the last point \( p_i \) and the last road segment \( e_j \) are matched, \( L(strj(i), rss(j)) \) must be the larger value between \( L(strj(i - 1), rss(j - 1)) \) and \( L(strj(i - 1), rss(j)) \), as multiple consecutive GPS points are allowed to be matched to the same road segment.

Moreover, the following two inequalities always hold,

\[
L(strj(i - 1), rss(j)) + \text{Sim}(p_i, e_j) \geq L(strj(i - 1), rss(j)),
\]

When \( p_i \) and \( e_j \) are not matched (\( \text{Sim}(p_i, e_j) = 0 \)), the equality is established.
\[ L(strj(i - 1), rss(j)) + Sim(p_i, e_j) \geq L(strj(i - 1), rss(j - 1)) + Sim(p_i, e_j), \]

When \( p_{i-1} \) and \( e_j \) are not matched \( (Sim(p_{i-1}, e_j) = 0) \), the equality is established.

Thus, Equation (3.1) can be converted to Equation (3.2) with the inequalities,

\[ L(strj(i), rss(j)) = \max \{ L(strj(i), rss(j - 1)), \]

\[ L(strj(i - 1), rss(j)) + Sim(p_i, e_j) \} \quad (3.2) \]

With Equation (3.2), the LCS value between \( strj \) and \( rss \) can be solved with the dynamic programming method. Next, the LCS-HP distance is defined as,

\[ LCS_{HP}(strj, rss) = 1 - \text{penalty} \times L(strj, rss)/m \quad (3.3) \]

\[ \text{penalty} = \max (1, \frac{\text{Length of matched } strj}{\text{Length of matched } rss}) \quad (3.4) \]

Where \( LCS_{HP}(strj, rss) \) is the LCS-HP distance between \( strj \) and \( rss \). First, LCS value between \( strj \) and \( rss \), ranging from \([0, m]\), is normalized by dividing the size of \( strj \). A penalty is then applied based on the assumption that the greater difference between the heading of \( strj \) and the heading of \( rss \) will lead to the larger difference between the length of \( strj \) and the length of \( rss \). The penalty value, defined in Equation (3.4), ranges from 0 to 1, and the larger penalty value is due to the greater difference between the length of \( strj \) and the length of \( rss \).

### 3.3.5 Map matching in hidden Markov model

After GPS sub-trajectories generation and candidate RSS search, a sequence of GPS sub-trajectories \( STRJ: strj_1, strj_2, ..., strj_m \) and the set of RSSs \( \{rss_1, rss_2, ..., rss_n\} \) are generated. The segment-based map matching can be solved in hidden Markov model (HMM). Figure 3.9 gives an illustration of hidden Markov model for the segment-based
map matching. In this figure, each GPS sub-trajectory can be represented a round-corner rectangle. The black nodes inside of the rectangle are the candidate RSSs for the sub-trajectory. For instance, \( str_j_1 \) has two candidate RSSs \( rss_1 \) and \( rss_2 \), and \( str_j_2 \) has three candidate RSSs \( rss_2, rss_3 \) and \( rss_4 \). The arrow connecting two RSS nodes between two sub-trajectories represents that there exists an accessible path connecting the two RSSs in the road networks. Thus, the segment-based map matching problem is to find the most likely sequence of RSSs that result in a sequence of the current sub-trajectories.

Figure 3.9 Segment-based map matching using HMM
Formally, the segment-based map matching problem can be formally defined as:

**Definition 3.7. Segment-based Map Matching problem.** Given a sequence of GPS sub-trajectories $STRJ = \{strj_1, strj_2, ..., strj_m\}$, a set of RSSs $RSS = \{rss_1, rss_2, ..., rss_n\}$, the **Segment-based Map Matching problem** is to find the sequence $Q = \{q_1, q_2, ..., q_m\}$ with the maximum probability over the sequence $STRJ$, where $q_i \in RSS$.

$$
\max_{q_i \in RSS} P(q_1, q_2, ..., q_m, strj_1, strj_2, ..., strj_m) \quad (3.5)
$$

In this thesis, to solve the problem, Viterbi algorithm (Viterbi 1967) is used. $v_t(rss_j)$ is the probability of the sequence of the hidden states where the last state is $rss_j$ at time $t$, denoted as

$$
v_t(rss_j) = P(q_1, q_2, ..., q_{t-1}, strj_1, strj_2, ..., strj_t, q_t = rss_j) \quad (3.6)
$$

Equation (3.6) can be converted to Equation (3.7) based on Markov assumption,

$$
v_t(rss_j) = \max_{1 \leq i \leq n} v_{t-1}(rss_i) * Pr(rss_j|rss_i, strj_{t-1}, strj_t) * Pr(strj_t|rss_j) \quad (3.7)
$$

Where emission probability $Pr(strj_t|rss_j)$ is to measure the probability of $strj_t$ located on the state $rss_j$ and transition probability $Pr(rss_j|rss_i, strj_{t-1}, strj_t)$ represents the probability that the object moves from the candidate RSS $rss_i$ of $strj_{t-1}$ to the candidate RSS $rss_j$ of $strj_t$.

Starting from $v_0(rss)$, the probabilities $v_m(rss)$ for all candidate RSSs at the time $m$ can be calculated out recursively with Equation (3.7). $v_0(rss)$ is called initial probability and is set as the same value for all RSSs. After the probabilities $v_m(rss)$ for all
candidate RSSs at the time $m$ are known, the candidate RSS $rss^*$ with the maximum probability $v_m(rss)$ at time $m$ can be found.

$$rss^* = \text{Argmax}_{rss \in RSS} v_m(rss)$$

As mentioned above, each probability $v_t(rss_j)$ at time $t$ is calculated based on certain probability $v_{t-1}(rss_i)$ at the previous time $t - 1$, so the path with the maximum probability in the hidden Markov model can be traced back starting from $rss^*$ at the time $m$.

Next, the calculation of emission probability and transition probability in SHMM are discussed.

The emission probability $Pr(str_j | rss_j)$ is used to measure the probability of str$_j$ located on the rss$_j$. For a moving object, it is assumed that its spatial trajectory and its route in the road network should have small distance, and that the distance follows Gaussian distribution. Thus, the emission probability is defined as in Equation (3.8),

$$Pr(rss_j | str_j) = \frac{1}{\sqrt{2\pi \sigma_z}} e^{-\frac{1}{2} \left( \frac{\text{Dist}(str_j, rss_j)}{\sigma_z} \right)^2} \quad (3.8)$$

Where $\text{Dist}(str_j, rss_j)$ is the distance between str$_j$ and rss$_j$, which can be any distance metric between str$_j$ and rss$_j$, such as the proposed LCS-HP distance and Fréchet distance. $\sigma_z$ is the standard deviation of the distance between GPS sub-trajectories and road segment sequences.

The transition probability $Pr(rss_j | rss_i, str_{j-1}, str_j)$ is the probability that the object moves from the candidate RSS rss$_i$ of str$_{j-1}$ to the candidate RSS rss$_j$ of str$_j$. Newson and Krumm (2004) assume that the transition probability between two RSSs
follows exponential distribution and this assumption is adopted in this thesis. Thus, the transition probability is defined as follows:

\[
Pr(rss_j|rss_i,str_{j_{t-1}}) = \frac{1}{\rho} e^{-\frac{|Dist(prj_i, prj_j) - Dist(p_{last}, p_{first})|}{\rho}}
\]  

(3.9)

Where \( prj_i \) is the projection point of \( str_{j_{t-1}.last} \) on the last road segment in \( rss_i \), and \( prj_j \) is the projection point of \( str_{j_{t}.first} \) on the first road segment in \( rss_j \), \( Dist(prj_i, prj_j) \) is the network distance between \( prj_i, prj_j \) in the road network. \( p_{first} \) and \( p_{last} \) are \( str_{j_{t}.first} \) and \( str_{j_{t-1}.last} \), respectively, and \( Dist(p_{first}, p_{last}) \) represents the Euclidean distance between \( p_{first} \) and \( p_{last} \). Figure 3.10 gives an example for transition probability.

In Figure 3.10, the GPS trajectory will be divided into two parts \( str_{j_1} = \{p_1, p_2, p_3\} \) and \( str_{j_2} = \{p_4, p_5, p_6\} \). In the road networks, the road segment \( e_1, e_2 \) are one-way road and \( e_3, e_4 \) are two-way roads. The three candidate RSSs of \( str_{j_1} \) are found \( rss_1 = \{e_1\} \), \( rss_2 = \{e_1, e_2\} \), \( rss_3 = \{e_1, e_3\} \), and two candidates RSSs of \( str_{j_2} \) are found \( rss_4 = \{e_3\} \), \( rss_5 = \{e_4\} \). The \( p_3 \) is a connection point shared by \( str_{j_1} \) and \( str_{j_2} \), and its projection on \( e_1, e_3, e_4 \) are \( p_a, p_b, p_c \). As the network distance between \( p_a \) and \( p_b \) is smaller than the distance between \( p_a \) and \( p_c \), the transition probability \( Pr(rss_4|rss_1,str_{j_1},str_{j_2}) \) is larger than \( Pr(rss_5|rss_1,str_{j_1},str_{j_2}) \).
The parameters $\sigma_z$ and $\rho$ are estimated with the median absolute deviation method for Gaussian distribution (Huber 1981) and the standardized median method for exponential distribution (Gather and Schultze 1999), respectively.

3.4 Experiments

3.4.1 Dataset

In this section, the performances of the proposed SHMM method is evaluated with a real dataset. The method was implemented in C#. The experiments were conducted on a 2.5 GHz Core i7 PC with 16GB of RAM. The GPS trajectories were obtained from a taxi GPS trajectory dataset in Shenzhen, China, and the measurement errors of GPS trajectories
are around 5 meters. Besides, the road network contains 210,544 road vertices and 243,626 road segments. 90 GPS trajectories were extracted from the taxi GPS trajectory dataset. The distribution of length of the GPS trajectories is illustrated in Figure 3.11.

![Figure 3.11 Distribution of length of GPS trajectories]

It can be observed that the lengths of most GPS trajectories in the experiment dataset is around 22km. GPS trajectories of less than 15km and more than 30km took account for 23.4% and 21.3%, respectively, while GPS trajectories with length between 15km and 30km accounted for 55.3%. The minimum value and maximum value of the trajectories are 2.709km and 44.476km, respectively.

In this experiment, the performance of SHMM is compared with two baseline methods, point-based hidden Markov model (called PHMM) (Newson and Krumm, 2009)
and the incremental map matching method (called Incremental) (Greenfeld 2002). The performance of SHMM with different trajectory distance metrics, Fréchet distance (called SHMM-FD) and LCS-HP distance (called SHMM-LCSHP), are also compared.

In this experiment, two metrics Accuracy by Number of road edges (AN) and Accuracy by Length of road edges (AL) are used to evaluate the effectiveness of map matching algorithms, and they are defined as follows.

\[
AN = \frac{\text{#correctly matched road segments}}{\text{#all road segments of the trajectory}}
\]

\[
AL = \frac{\text{Length of correctly matched road segments}}{\text{Length of all road segments of the trajectory}}
\]

### 3.4.2 Accuracy evaluation

![Figure 3.12 Accuracy evaluation of the experiments.](image)

Figure 3.12 Accuracy evaluation of the experiments.
The evaluation for accuracies of three methods are shown in Figure 3.12. The experiment results in this figure show that the proposed SHMM outperforms the point-based HMM and incremental method in both the accuracy by number and by length. Moreover, SHMM with LCS-HP distance (SHMM-LCSHP) achieves the best performance. The accuracy by number and by length for SHMM-LCSHP were 90.4% and 89.8%, which were improvement of the point-based HMM method by 7.2% and 9.4%. The accuracy by number and by length for SHMM with Fréchet distance (SHMM-FD) were 87.2% and 86.1%, which were improvement of the point-based HMM method by 4% and 5.7%, respectively. The improvements mainly resulted from the following facts. First, the incremental method is a local based map matching method. The incremental method only considers the geometric attribute of GPS points to the road network, but it does not consider the accessibility between candidate road segments of adjacent GPS points in the road network, which makes it the smallest accuracies out of the three methods. When the incremental map matching method is very likely to locate GPS point onto the incorrect road segments only based on geometric information, and the mismatching of a GPS point will lead to continuous mismatching of next GPS points. Compared with the incremental method, the point-based HMM method is a method for global optimization and considers both the spatial proximity of individual GPS points to the road network and the accessibility between adjacent GPS points in the road network, so it performs better than incremental method. However, when accessibility of two paths between adjacent GPS points are similar, GPS point could be matched onto the closer road segment based on the distance. When the measurement error of GPS point increases, GPS point is likely to be located onto the incorrect road segment. Segment-based HMM method measures the distance between
GPS sub-trajectories and road segment sequence, so the negative impact of measurement error of individual GPS point on the map matching result is decreased. Furthermore, LCS-HP considers the heading feature of GPS sub-trajectories and is robust to the measurement error of individual GPS point, so it made a better performance than Fréchet distance. Figure 3.13 gives an example of real GPS trajectory and Figure 3.14 shows the map matching result of this GPS trajectory by SHMM-LCSHP.

3.4.3 Impact of sampling rate

10 trajectories with 5s sampling rate were extracted from the dataset to evaluate the impact of sampling rate on SHMM. The trajectories were resampled to different sampling rate (10s, 20s, 30s, 40s and 50s), and SHMM with LCS-HP distance were used to match the trajectories with different sampling rates onto the road network. The results are displayed in Figure 3.15. It can be observed that the accuracy decreases slowly with the increase of the sampling rates. The accuracies by number of road segment and by length of road segment did not change from 5s to 10s sampling rate, and the accuracies were 97.1% and 97.4%, respectively. When the sampling rate gets to 20s, the accuracy by number and by length increased by 0.2% and 0.1%, respectively. Thus, the accuracies were kept at a stable level when the sampling rate is smaller than 20s. When the sampling rate exceeds 20s, the accuracies decreased gradually. With the increase of sampling rate, the distance between two adjacent GPS points is enlarged, which leads to the growth of the uncertainty of GPS sub-trajectories, which may hamper the performance of SHMM.
Figure 3.13 a real GPS trajectory
Figure 3.14 Map matching result of the GPS trajectory by SHMM
Moreover, it could be observed that the accuracies might increase slightly with the decline of sampling rates. The reason is that some GPS points with high measurement error were removed from the GPS trajectory after resampling, which lead to the higher accuracies even with the lower sampling rates.

3.4.4 Running time

This section evaluates the efficiency of SHMM, and the running time is studied with respective to the number of GPS points and the number of GPS sub-trajectories respectively. Figure 3.16 shows the average running time for one trajectory in SHMM is around 6 seconds, while PHMM will cost around 83 seconds on average. As both SHMM and PHMM are map matching methods for global optimization, they will check the reasonability of transition between hidden states in adjacent stages in HMM. However, each GPS point is a stage in PHMM, while each GPS sub-trajectory is a stage in SHMM. The number of stages in SHMM is much less than that in PHMM, which will reduce a large
amount of computation to check the reasonability of transition between hidden states in adjacent stages. Thus, SHMM is more efficient than PHMM on average.

Figure 3.16 Comparison of running time between SHMM and PHMM.

Figure 3.17 displays the running time grows slowly with the increase of the number of GPS points in a trajectory. Similarly, it has been demonstrated in Figure 3.18 that the running time of SHMM increases slowly with the number of the generated GPS sub-trajectories after trajectory segmentation.
3.5 Summary

In this chapter, the segment-based hidden Markov model method, called SHMM, is proposed. SHMM divides GPS trajectories into GPS sub-trajectories and applies the hidden Markov model on the GPS sub-trajectories for map matching. The first step of
SHMM is trajectory segmentation. A GPS trajectory will be divided into GPS sub-trajectories based on heading homogeneity so that GPS points in each GPS sub-trajectory have similar headings. The trajectory segmentation based on heading homogeneity has two advantages. First, the heading attribute can be used as a constraint to filter out unreasonable candidate road segment sequences. Second, heading attribute can be used as a significant feature when computing the similarity between GPS sub-trajectory and road segment sequence. The next step is the search for the candidate road segment sequence. SHMM first searches road segments based on R-tree and heading constraint, and then constructs candidate road segment sequences by connecting the road segments of the consecutive GPS points in a sub-trajectory. In the last step, hidden Markov model is applied for the segment-based map matching. SHMM considers GPS sub-trajectories as observations and road segment sequences as hidden states. Emission probability is used to measure the probability that a GPS sub-trajectory is located on a road segment sequence based on the distance between them, and transition probability is used to measure the reasonability of connections between adjacent GPS sub-trajectories in the road network. Moreover, a metric, called LCS-HP, is proposed to measure the distance between GPS sub-trajectory and road segment sequence by considering the heading attributes.

SHMM improves the point-based HMM map matching method in both efficiency and effectiveness. First, the hidden Markov model requires to calculate the transition probability between the hidden states in adjacent time stages, so the shortest distance path should be calculated between the candidate road segments of adjacent GPS points in the point-based HMM. Compared with the point-based hidden Markov model, SHMM enormously reduce the calculation of the shortest distance path, because the number of
GPS sub-trajectories is much less than the number of GPS points. Moreover, the distance measurement between GPS sub-trajectories and road segment sequences is more reliable than the distance measurement between individual GPS point and the road segment, as more information is considered rather than only considering the location of one GPS point.

Specifically, the contributions of this research are summarized as follows:

1) The segment-based hidden Markov model map matching method is proposed in this thesis. SHMM contains four steps: data preprocessing, trajectory segmentation, candidate road segment sequence search and map matching in hidden Markov model.

2) A trajectory segmentation method based on heading homogeneity, called TSHH, is developed to partition GPS trajectories into GPS sub-trajectories. The segmentation method generates the GPS sub-trajectories with heading attribute, which is beneficial for candidate road segment sequence search, and contributes to the distance measurement between GPS sub-trajectories and road segment sequences.

3) Longest common subsequence with heading penalty, called LCS-HP, is proposed to measure the distance between GPS sub-trajectory and road segment sequence. LCS-HP is robust to the measurement error of GPS point, and considers the heading information when measure the distance between GPS sub-trajectory and road segment sequence.

4) Hidden Markov model is applied for segment-based map matching. The emission probability in hidden Markov model is determined by the distance between GPS sub-trajectory and road segment sequence. The transition probability reflects the accessibility between the road segment sequences of two consecutive GPS sub-
trajectories in the road network.

5) A case study is conducted on a real taxi GPS trajectory dataset, and the experiment results show that the proposed SHMM method outperforms the baseline method in accuracy and efficiency.
Chapter 4: Collaborative travel route recommendation method

This chapter proposes the collaborative travel route recommendation method, called CTRR, to provide user a personalized route recommendation. Chapter 4.1 states the personalized travel route recommendation problem. Chapter 4.2 introduces the related definitions of CTRR. Chapter 4.3 gives the overview of CTRR. Chapter 4.4 makes the preprocessing for GPS trajectories with the proposed entropy-based histogram thresholding method and SHMM method. Chapter 4.5 utilizes matrix factorization and Laplace smoothing method to estimate the frequency and the probabilities of users’ appearance behaviours. Chapter 4.6 calculates the maximum probability route based on the naïve Bayes model. Chapter 4.7 extends CTRR to CTRR+ by integrating distance. Chapter 4.8 evaluates the experiment results. Chapter 4.9 gives a summary about collaborative travel route recommendation method.

4.1 Personalized Travel Route Recommendation Problem

Route recommendation provides users specific route guidance when they travel to some locations. As most users only travel on the routine routes daily and are unfamiliar with many locations in the city, route recommendation has become increasingly significant especially in this era of rapid urban development.

One deficiency of the current route recommendation services is that the route recommendation is not personalized, which provides the identical route recommendation result for all users from an origin location to a destination. However, users may have different travel preferences, such as distance, travelling time, fuel consumption, traffic
flow, road condition, etc. Thus, to provide user a personalized route recommendation by considering their personal travel preferences is useful.

Nonetheless, how to understand users’ travel preferences is challenging. To investigate the travel preferences for each user based on surveys is inapplicable because surveys for users’ travel preferences would cost a lot of resources and they might be imprecise because users’ travel preferences may vary with different travelling time and area. Fortunately, with the pervasiveness of mobile devices and the development of positioning technology, GPS trajectories provide an alternative way to understand users’ travel preferences.

As mentioned in Chapter 1, there are some challenges to provide the personalized route recommendation based on GPS trajectories. First, how to extract user’s travel behaviours from the historical GPS trajectories and estimate the probabilities of the travel behaviours for each user should be solved. Second, as the GPS trajectories of each user only cover a small part of the study area, how to estimate the probabilities of user’s travel behaviours in the area where they have never travelled before is worthy studying. Last, it is important to calculate the personalized route for each user based on the estimated probabilities of travel behaviours.

4.2 Definitions

GPS trajectories usually record the user’s movements over a long period of time, e.g. a whole day; therefore, a user’s trajectory may contain several sub-trajectories. When a GPS trajectory is not updated for a long time, two GPS readings would not be “logically” consecutive, meaning that their physical distance is not a good estimation of the distance
that the user travelled in that time interval. The original trajectory is divided to sub-trjectories, called GPS traces. A GPS trace can be detected by using a threshold on the updating time of the GPS readings.

**Definition 4.1 GPS trace.** A GPS trace of a user is a sequence of GPS-readings 
\[
\text{trace} = (p_1, p_2, p_3, \ldots, p_m),
\]
where \(0 < p_i.t - p_{i-1}.t \leq \varphi, 1 < i \leq m\) and \(\varphi\) is a defined threshold to split trajectories into traces. Parameter \(\varphi\) is an empirical value and is specified in implementation. When the time interval between two consecutive GPS-readings exceeds threshold \(\varphi\) in a trajectory, it means the trace is ended, and a new trace should be generated.

**Definition 4.2 Route.** Given road network \(G = (V, E)\), a route from vertex \(v_i\) to vertex \(v_j\) is a sequence of connected road segments \(rt = (v_i, r_1, r_2, r_3, \ldots, r_n, v_j)\) which starts at vertex \(v_i\), and ends at vertex \(v_j\) where \(v_i, v_j \in V, r_i \in E\) and \(r_i\) is the \(i\)-th road segment in \(rt\), \(r_i \neq r_j\) if \(i \neq j\), and \(r_1.\text{start} = v_i, r_n.\text{end} = v_j\).

In this thesis, it describes which road segment a user travels at a given time and this information can be extracted from GPS trajectories, leading to the definition of the concept of appearance behaviour.

**Definition 4.3 Appearance behaviour.** Given road network \(G = (V, E)\) and set of time intervals \(T = \{t_1, t_2, \ldots, t_y\}\) of a day, the appearance behaviour \(b\) of a user is a tuple denoted as:

\[
b = (e, t), \quad \text{where } e \in G.E \text{ and } t \in T.
\]

The concept of appearance behaviours describes the location and time of a user’s movements. Given user \(u\) and his/her appearance behaviour \(b\), \(\text{frq}(u, b)\) is the frequency
of the times that user $u$ has appearance behaviour $b$. The frequency of appearance behaviours is referred to as appearance behaviour frequency for conciseness.

The following paragraphs present an example to illustrate the concept of appearance behaviours.

Figure 4.1 shows that two users, A and B, are travelling from 7 am to 8 am and from 5 pm to 6 pm on different road segments. Red arrows indicate user A’s traces, and blue arrows represent user B’s traces. The thickness of the lines represents the number of the traces the user travelled on the road segment. As shown in the figure, from 7 am to 8 am, user A had 5 traces of $ab \rightarrow bc$ and 1 trace of $ab \rightarrow bg$; and, user B had 3 traces of $bc \rightarrow cd$. From 5 pm to 6 pm, user A had 3 traces of $bg \rightarrow gf \rightarrow fe \rightarrow ed$, and user B had 5 traces of $ah \rightarrow hg$.

![Figure 4.1 Example of two users travel in the road networks.](image)

In this simple example, the appearance behaviours were extracted from the traces, with Table 4.1 summarizing the appearance behaviours of the two users from Figure 4.1. User A exhibited appearance behaviours of $b_1 = (ab, 7am \rightarrow 8am), b_2 = (bc, 7am \rightarrow 8am), b_3 = (bg, 7am \rightarrow 8am), b_4 = (bg, 5pm \rightarrow 6pm), b_5 = (gf, 5pm \rightarrow 6pm), b_6 = (fe, 5pm \rightarrow 6pm)$ and $b_7 = (ed, 5pm \rightarrow 6pm)$. The frequencies of the appearance behaviours by user A were $frq(A, b_1) = 6$, $frq(A, b_2) = 5$, $frq(A, b_3) = 6$, $frq(A, b_4) = 5$, $frq(A, b_5) = 5$, $frq(A, b_6) = 5$, $frq(A, b_7) = 5$. The frequencies of the appearance behaviours by user B were $frq(B, b_1) = 5$, $frq(B, b_2) = 5$, $frq(B, b_3) = 5$, $frq(B, b_4) = 5$, $frq(B, b_5) = 5$, $frq(B, b_6) = 5$, $frq(B, b_7) = 5$. The frequencies of the appearance behaviours for both users are given in Table 4.1.
1, \( frq(A, b_4) = 3, \) \( frq(A, b_5) = 3, \) \( frq(A, b_6) = 3 \) and \( frq(A, b_7) = 3. \) Similarly, the appearance behaviours of user B were \( b_2 = (bc, 7am-8am), \) \( b_8 = (cd, 7am-8am), \) \( b_9 = (ah, 5pm-6pm) \) \( b_{10} = (hg, 5pm-6pm) \), and \( frq(B, b_2) = 3, \) \( frq(B, b_8) = 3, \) \( frq(B, b_9) = 5, \) \( frq(B, b_{10}) = 5. \)

<table>
<thead>
<tr>
<th>User</th>
<th>Appearance behaviour</th>
<th>Segment</th>
<th>Time interval</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( b_1 )</td>
<td>( ab )</td>
<td>7am-8am</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( b_2 )</td>
<td>( bc )</td>
<td>7am-8am</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( b_3 )</td>
<td>( bg )</td>
<td>7am-8am</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>( b_4 )</td>
<td>( bg )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_5 )</td>
<td>( gf )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_6 )</td>
<td>( fe )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_7 )</td>
<td>( ed )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>( b_2 )</td>
<td>( bc )</td>
<td>7am-8am</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_8 )</td>
<td>( cd )</td>
<td>7am-8am</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_9 )</td>
<td>( ah )</td>
<td>5pm-6pm</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( b_{10} )</td>
<td>( hg )</td>
<td>5pm-6pm</td>
<td>5</td>
</tr>
</tbody>
</table>

Neither of the users travelled on all road segments at the given travel time in this example. What if the user B would like to travel from origin \( a \) to destination \( e \) at 5-6 pm? Which route will he/she prefer to take? In this case, it needs to estimate user B’s appearance behaviour frequencies on the missing road segments at the given time.

### 4.3 Framework of Collaborative Travel Route Recommendation

CTRR studies the probabilities of user’s appearance behaviours to provide user the travel route with the maximum probability which he/she may prefer. The framework of CTRR is shown in Figure 4.2:

The first step of CTRR is data preparation. Two types of datasets were used in the study: road networks and users’ historical GPS trajectories. First, GPS trajectories are
divided into GPS traces using the proposed entropy-based histogram thresholding method. To extract the appearance behaviours of each user, the GPS traces are then matched to the road networks based on the hidden Markov model (Newson and Krumm 2009). After the map matching, user traces are then represented as user routes on the road networks. The appearance behaviours of each user can then be extracted, and the user’s appearance behaviour frequencies are counted.

Figure 4.2 Framework of the CTRR method.
As previously mentioned, users generally only travel on limited road segments in the road network. Therefore, for each user, the appearance behaviour frequency of many road segments and time intervals are zero. The second step of the CTRR is the estimation of the appearance behaviour frequency through the construction of a user-appearance behaviour matrix, with each element in the matrix as the frequency of the user’s appearance behaviour. First, it smooths the frequencies of users’ appearance behaviours by considering the correlation among user’s appearance behaviours in nearby time interval. The matrix factorization method then processes the user-appearance behaviour matrix to obtain the estimated appearance behaviour frequency, which reflects the user’s appearance behaviour probability.

The third step of CTRR is route computation, which searches for the maximum probability route of users based on the naïve Bayes model. The naïve Bayes model assumes that appearance behaviours are independent of each other, allowing it to efficiently find a route with maximum probability. First, the appearance behaviour probability is calculated from the estimated appearance behaviour frequency, which represents the probability that a user would exhibit this appearance behaviour. Next, a road network graph is weighted based on the log-inversed appearance behaviour probability that is transformed from the appearance behaviour probability. Finally, the least weighted route is computed with Dijkstra’s algorithm.

Each step is discussed in detail in the following subsections.
4.4 Data Preprocessing

In CTRR, data preparation includes three tasks: division of trajectories into GPS traces, matching of GPS traces on the road network, and extraction of appearance behaviours for each user.

As previously mentioned, sometimes two GPS readings in one GPS trajectory are not logically consecutive, because the GPS readings may not have been updated for a long time due to GPS signal loss. The ideal way of dividing trajectories would be based on a priori knowledge and contextual information of the trajectories, if available. However, the context and reason of the GPS signal loss is usually not available and unknown. An empirical method is needed to separate the trajectories into GPS traces without any contextual information. One empirical solution for GPS trajectory segmentation is based on time interval (Zheng 2015). Applying the time interval threshold for trajectory segmentation basically classifies GPS sampling rates into two categories: normal sampling rates which are smaller than or equal to the threshold, and anomaly sampling rates which are larger than the threshold. The trajectory will be divided when the time interval between two consecutive GPS readings falls into anomaly sampling rates. In this study, if the threshold is set too small, a trajectory may be split unnecessarily and some user appearance behaviours between two GPS traces will be lost; if the threshold is set too large, the uncertain part of trajectory will make user’s appearance behaviour extraction imprecise. Therefore, an effective method for threshold determination is critical when no contextual information is available.

The entropy-based histogram thresholding method is proposed in this research to determine the time interval threshold. As mentioned, the trajectory segmentation method
based on sampling rate (i.e. time interval) is to split the sampling rates as normal or anomaly classes so that the trajectory can be divided when the sampling rate is anomaly. The splitting criterion should tell us which threshold value to use by determining the “best” way to partition the sampling rates into the normal and anomaly classes. The proposed entropy-based histogram thresholding method is based on the principle of maximum entropy (Kapur et al. 1985, Wong and Sahoo 1989). Entropy is a well-known measurement for randomness in information theory, and maximum entropy is a general technique for estimating probability distributions from data. The principle of maximum entropy considers that, when nothing is known, the distribution should be as uniform as possible, that is, have the maximum entropy (Nigam et al. 1999). In this study, a histogram about the distribution of GPS sampling rates is built. When no other information is available about the normal and anomaly sampling rates, the principle of maximum entropy considers that the normal and anomaly sampling rates can be best classified when the distribution of normal sampling rates and the distribution of anomaly sampling rates are both as uniform as possible. Specifically, the most reasonable distribution of normal sampling rates and anomaly sampling rates should be the one which are most randomized when no other prior information is available, namely when the summation of the entropy of normal sampling rate distribution and anomaly sampling distribution reaches the maximum.

Figure 4.3 provides an example of a histogram representing the distribution of GPS sampling rates. The horizontal axis of the histogram represents the sampling rate between two consecutive GPS readings. The vertical axis represents the frequency of the corresponding sampling rate. As shown in the figure, if a threshold is given, the normal sampling rates are smaller than or equal to the threshold and anomaly sampling rate are
larger than the threshold, so the distribution is skewed to the right. However, it is difficult to determine the threshold, as the histogram is not bimodal. A reasonable threshold to distinguish normal sampling rates and anomaly sampling rates can be identified based on the proposed entropy-based histogram thresholding method.

Figure 4.3 Histogram of sampling rates between GPS readings.

If there are \( n \) distinct sampling rates \( (s_1, s_2, \ldots, s_n) \), let \( f_1, f_2, \ldots, f_n \) be the observed number for \( s_1, s_2, \ldots, s_n \) respectively, and \( N = \sum_{i=1}^{n} f_i \) is the total number of sampling rates, the probability of sampling rate \( i \) is \( p_i \):

\[
p_i = \frac{f_i}{N}, \text{ and } \sum_{i=1}^{n} p_i = 1, \quad i = 1, \ldots, n;
\]  

(4.1)

If \( t \) is the selected threshold differentiating two kinds of sampling rates, the entropy associated with the distribution of normal sampling rates is:

\[
E_1 = -\sum_{i=1}^{t} \frac{p_i}{\sum_{j=1}^{t} p_j} \ln \frac{p_i}{\sum_{j=1}^{t} p_j}
\]  

(4.2)
The entropy associated with the distribution of anomaly sampling rates can be calculated as:

\[ E_2 = - \sum_{i=t+1}^{n} \frac{p_i}{\sum_{j=t+1}^{n} p_j} \ln \frac{p_i}{\sum_{j=t+1}^{n} p_j} \]  

(4.3)

In Equation (4.4), the histogram threshold discriminates between the two kinds of sampling rates ranges from 1 to \( n \). Its aim is to find threshold \( \varphi \) that leads to the maximum summation entropy of the distribution of the normal sampling rates \( E_1 \) and the anomaly sampling rates \( E_2 \):

\[ \varphi = \text{Arg} \max_{t=1,...,n} (E_1 + E_2) \]  

(4.4)

\( \varphi \) can effectively differentiate the distributions of the two kinds of sampling rates; and, when the time interval between two consecutive GPS readings exceeds threshold \( \varphi \), the GPS trace is ended, and a new GPS trace should be generated.

The second task of the data preparation step is the matching of GPS traces to the road networks. The map matching method is used in this thesis to locate the GPS traces onto the road networks. After matching GPS traces to the road network, the appearance behaviours for each user can be extracted from the GPS traces.

4.5 Appearance Behaviour Frequency Estimation

As previously mentioned, users generally travel on very limited routes daily, covering only a small number of road segments of the study area. Thus, a CF technique is used to estimate users’ appearance behaviour frequencies on all road segments. The first step is the generation of a user-appearance behaviour matrix.
4.5.1 User-appearance behaviour matrix

To simplify the discussion, the day is divided into regular time intervals. The first step of appearance behaviour frequency estimation is to find the set of \( n \) appearance behaviours \( B = \{ b_1, b_2, \ldots, b_n \} \) where \( B \) contains all possible appearance behaviours. Thus, given a set of \( m \) users \( U = \{ u_1, u_2, \ldots, u_m \} \) and a set of \( n \) appearance behaviours \( B = \{ b_1, b_2, \ldots, b_n \} \), pairs of \((u, b)\) can be used to construct user appearance behaviour matrix \( UB_{m \times n} \) in Figure 4.4. The element in the matrix is the frequency of the pair \((u_i, b_j)\), i.e. \( frq(u_i, b_j) \), denoted as \( UB_{i,j} \), i.e. \( UB_{i,j} = frq(u_i, b_j) \).

\[
UB_{m \times n} = \begin{bmatrix}
frq(u_1, b_1) & frq(u_1, b_2) & \ldots & frq(u_1, b_n) \\
frq(u_2, b_1) & frq(u_2, b_2) & \ldots & frq(u_2, b_n) \\
\vdots & \vdots & \ddots & \vdots \\
frq(u_m, b_1) & frq(u_m, b_2) & \ldots & frq(u_m, b_n)
\end{bmatrix}
\begin{bmatrix}
 u_1 \\
 u_2 \\
 \vdots \\
 u_m
\end{bmatrix}
\]

Figure 4.4 User-appearance behaviour matrix.

4.5.2 Appearance behaviour smooth for temporal correlation

As mentioned, users’ appearance behaviours extracted from historical GPS trajectories are sometimes very limited, so the frequencies of the user’s appearance behaviours in many time intervals on the road segment are unknown. However, users usually perform similar behaviours at the similar time of the day (Rahimi and Wang, 2013). In other words, the appearance behaviours happened in a certain time interval should have some correlations with the appearance behaviours happened at the approximate time.
Therefore, in this thesis, it considers the temporal correlations among appearance behaviours using a decay function, which gives a higher weight on the appearance behaviours in the closer time interval. Thus, the frequency of the user $u$ with appearance behaviour $b_i$ can first be represented by the following equation.

$$ f rq(u, b_i) = \frac{\sum_{b} f rq(u, b_j) e^{-|t_i - t_j|}}{n}, \quad b_i = (e, t_i) \text{ and } b_j = (e, t_j) $$ (4.5)

Where $n$ is the total number of time intervals in one day, and $b$ is the set of appearance behaviours happening on the road segment $e$. Equation (4.5) considers the correlation between appearance behaviours in the approximate time intervals, and estimates the frequency of user’s travel frequencies by smoothing the frequencies of the appearance behaviours by the nearby time intervals.

**4.5.3 Matrix factorization for appearance behaviour frequency estimation**

MF is an effective CF method for prediction of users’ preferences by constructing a user-item matrix. In this thesis, the user-appearance behaviour matrix $UB_{m \times n}$ can be constructed given $m$ users and $n$ appearance behaviours. The MF method is then utilized to predict users’ frequencies for appearance behaviours.

Users and appearance behaviours are characterized by two vectors of latent factors. To be more specific, each user $u_i$ is associated with a vector of latent factors, $p_{u_i} = (f_{u_i1}, f_{u_i2}, ..., f_{u_i k})$, and each appearance behaviour $b_j$ is associated with vector $q_{b_j} = (f_{b_j1}, f_{b_j2}, ..., f_{b_j k})$, where $k$ is the length of the vectors. It is assumed that the latent factors underline the interactions between the users and the appearance behaviours and that a high
inner product of vectors $p_{u_i}$ and $q_{b_j}$ implies a possible high frequency of appearance
behaviour $b_j$ by user $u_i$. The predicted frequency $\hat{U_{B_{ij}}}$ of appearance behaviour $b_j$ by user
$u_i$ is approximated by the dot product of $p_{u_i}$ and $q_{b_j}$:

$$\hat{U_{B_{ij}}} = q_{b_j}^T p_{u_i} \quad (4.6)$$

The vectors of latent factors are learned by minimizing the regularized squared
error of the set of known ratings:

$$L = \sum_{(u_i, b_j) \in S} (UB_{i,j} - q_{b_j}^T p_{u_i})^2 + \lambda (\|q_{b_j}\|^2 + \|p_{u_i}\|^2) \quad (4.7)$$

where $S$ is the set of the $(u_i, b_j)$ pairs for which appearance behaviour $b_j$ by user
$u_i$ is known (the training set), $UB_{i,j}$ is the frequency of appearance behaviour $b_j$ by user
$u_i$, and the constant $\lambda$ controls the extent of regularization.

---

**Algorithm EstimateTravelBehaviourFrequency ($UB_{m \times n}, k, \epsilon, \gamma$)**

| Input: | the user-travel behaviour matrix $UB_{m \times n}$, the number of latent factors $k$, the threshold $\epsilon$, the step size $\gamma$. |
| Output: | the estimated travel behavior matrix $\hat{UB}_{m \times n}$, and each element of $\hat{UB}_{ij}$ is predicted frequency of $(u_i, b_j)$. |

1. Initialize vectors of users $p_{u_1}, p_{u_2}, ..., p_{u_m}$ and vectors of travel behaviours $q_{b_1}, q_{b_2}, ..., q_{b_n}$ with random values.
2. $\hat{UB}_{m \times n} = UB_{m \times n}$ // Initialize the matrix $\hat{UB}_{m \times n}$
3. $t = 1$
4. **While** ($L_t - L_{t+1} > \epsilon$) **do**
5.  select a $u_i, b_j$ randomly
6.  $\nabla p_{u_i} = (UB_{i,j} - q_{b_j}^T p_{u_i}) \cdot q_{b_j} - \lambda \cdot p_{u_i}$
7.  $\nabla q_{b_j} = (UB_{i,j} - q_{b_j}^T p_{u_i}) \cdot p_{u_i} - \lambda \cdot q_{b_j}$
8.  $p_{u_i} = p_{u_i} + \gamma \nabla p_{u_i}$
9.  $q_{b_j} = q_{b_j} + \gamma \nabla q_{b_j}$
10. **End While**
11. **For** ($i = 0; i < m; i++$)
12.   **For** ($j = 0; j < n; j++$)
13.      $\hat{UB}_{ij} = q_{b_j}^T p_{u_i}$
14. **End For**
15. **End For**

**Figure 4.5 Pseudo-code of estimating users’ appearance behaviour frequency.**
The pseudo-code for estimating users’ appearance behaviour frequency by MF is shown in Figure 4.5. The vectors of latent factors are learned by stochastic gradient descent (lines 1-10). First, the vectors of latent factors are initialized with random values. User $u_i$ and appearance behaviour $b_j$ are then selected randomly for each step; and, the gradient of Equation (4.7) is calculated in the direction of vectors $p_{u_i}$, and $q_{b_j}$ of $u_i$ and $b_j$, respectively (lines 6-7). The vectors are then updated based on the gradients (lines 8-9). Finally, users’ appearance behaviour frequencies are estimated based on the learned vectors of latent factors (lines 11-15).

If an appearance behaviour with respect to time and road segment is defined, each user’s appearance behaviour frequency on the road network can be estimated by the MF method.

In this thesis, it has been assumed that the appearance behaviour frequency reflects the user’s travel preference. Moreover, presuming that the appearance behaviours have been repeated many times under similar circumstances, the relative frequency of the appearance behaviour is an approximate of the actual probability of the occurrence of the appearance behaviour.

The personalized travel route recommendation problem can be achieved by finding a route along which the probability of user’s appearance behaviours is maximized based on user’s historical GPS trajectories.

In the following subsection, the use of the naïve Bayes model in finding the route, as the third step of the CTRR method, is discussed.
4.6 Route Computation with Naïve Bayesian Model

Given time $t$, origin vertex $o$ and destination $d$, it aims to recommend user $u$ with route $R = (r_1, r_2, r_3, ..., r_n)$, $r_1$ start = $o$ and $r_n$ end = $d$, where the appearance behaviour probability along route $P(R|u, t)$ is the maximum, i.e. the following probability is maximized:

$$P(R|u, t) = P(r_1, r_2, r_3, ..., r_n|u, t) = P(r_1, r_2, r_3, ..., r_n, u, t)/P(u, t) \quad (4.8)$$

Since $P(u, t)$ is constant when $u$ and $t$ are given, maximization can be calculated as:

$$P(r_1, r_2, r_3, ..., r_n, u, t)$$

$$= P(b_1, b_2, b_3, ..., b_n, u), \quad \text{where} \ b_i = (r_i, t)$$

$$= P((u, b_1), (u, b_2), (u, b_3), ..., (u, b_n)) \quad (4.9)$$

The naïve Bayes model assumes that variables are independent of each other. In this thesis, it was assumed that the appearance behaviours on each road segment are independent and the route is searched where the probability of appearance behaviour is maximized with the naïve Bayes model.

According to the rule of conditional probability, Equation (4.9) is equal to:

$$P(u, b_1) * P((u, b_2)|P(u, b_1)) * ... * P((u, b_n)|(u, b_1), ..., (u, b_{n-1})) \quad (4.10)$$

As appearance behaviours are independent of each other in the naïve Bayes model, Equation (4.10) is equal to:

$$P(u, b_1) * P(u, b_2) * ... * P(u, b_n) \quad (4.11)$$
where \( P(u, b_i) \) is the appearance behaviour probability for user \( u \) with behaviour \( b_i \). Therefore, a sequence of appearance behaviours needs to be determined in order to maximize Equation (4.11).

Based on the above discussion, the personalized travel route recommendation problem can be formally defined as in Definition 4.4.

**Definition 4.4. Personalized travel route recommendation problem.** Given user \( u \), departure time \( t \), origin vertex \( o \) and destination vertex \( d \), the objective of the personalized travel route recommendation is to provide route \( R \) to the user so that the overall probability \( P(R|u, t) \), along the route is maximized. It is formulized as:

\[
R = \text{Arg max }_{R \in G} P(R|u, t) = \text{Arg max }_{r_1, ..., r_n \in G.E} P(u, b_1) \times P(u, b_2) \times ... \times P(u, b_n),
\]

where \( R = (r_1, r_2, r_3, ..., r_n) \), \( r_1 \).start = \( o \) and \( r_n \).end = \( d \); \( G = (V, E) \) is the road network; and, \( b_i = (r_i, t) \).

### 4.6.1 Appearance behaviour probability

The appearance behaviour probability of a user describes the likeliness of a user’s appearance behaviours. It can be calculated as the frequencies of the appearance behaviours over the total number of a user’s appearance behaviours. Given road network \( G = (V, E) \) and appearance behaviour \( b \), the probability of appearance behaviour \( b \) of user \( u \) can be calculated simply as:

\[
P(u, b) = \frac{\text{freq}(u, b)}{\sum_{b_i \in G} \text{freq}(u, b_i)}, \quad b_i = (e_i, t)
\]

(4.12)
where $P(u, b)$ is the probability of $u$’s appearance behaviour on road segment $e$ at time $t$, $S$ is the set of the appearance behaviours of user $u$, and $\hat{frq}(u, b_i)$ is the estimated appearance behaviour frequency using MF.

There may, however, exist very few road segments where users in the system have never travelled, according to their historical GPS trajectories. In other words, some appearance behaviours may never occur for all users. In this case, the Laplace smoothing method is used to estimate the probability of these appearance behaviours, in order to avoid assigning a zero probability to any appearance behaviour, so that it is possible for users to travel to any road segment in the road network. Moreover, it is possible that the estimated frequencies of some of a user’s appearance behaviours are negative: the Laplace smoothing method is also used to solve the problem.

$$P(u, b) = \begin{cases} \frac{\sum_{i=1}^{d} \hat{frq}(u, b_i) + a}{\sum_{i=1}^{d} \hat{frq}(u, b_i) + a + d}, & \hat{frq}(u, b) > 0 \\ \frac{a}{\sum_{i=1}^{d} \hat{frq}(u, b_i) + a + d}, & \hat{frq}(u, b) \leq 0 \text{ or } b \text{ never happens for any users} \end{cases} \quad (4.13)$$

where $d$ is the number of total appearance behaviours, and $a$ is the smoothing parameter. Equation (4.13) can assign a non-zero probability to road segments where users have never travelled.

4.6.2 Route computation based on log-inversed appearance behaviour probability

As previously mentioned, the personalized route recommendation problem requires the maximization of $P(u, b_1) \times P(u, b_2) \times \ldots \times P(u, b_n)$ in Equation (4.10), in order to find the maximum appearance behaviour probability route. To transform the multiplication to
the summarization format required by the typical route planning algorithms, let \( L = \frac{1}{P(u,b_1) \cdot P(u,b_2) \cdot \ldots \cdot P(u,b_n)} \); thus, the problem is equivalent to the minimization of \( L \).

Taking the logarithm for both sides:

\[
\ln L = \ln \prod_{i=1}^{n} \frac{1}{P(u,b_i)} = \sum_{i=1}^{n} \ln \frac{1}{P(u,b_i)} \quad (4.14)
\]

where \( \ln \frac{1}{P(u,b_i)} \) is called the users’ log-inversed appearance behaviour probability in this thesis. The pseudo-code of calculating the log-inversed appearance behaviour probability is shown in Figure 4.6.

---

**Algorithm CalculateLogInversedAppearanceBehaviourProbability\( (u_0, t_0, a, G, UB_{m \times n}) \)**

**Input**: The user \( u_0 \); the time \( t_0 \); the smoothing parameter \( a \); the road network graph \( G = (V, E) \) and \( G.E = \{e_1, e_2, \ldots, e_d\} \); the estimated user-appearance behaviour matrix \( UB_{m \times n} \).

**Output**: The graph \( G \) with log-inversed appearance behaviour probability information

1. \( UB\_Sum = 0 \)
2. **Foreach** \((u, b) \) in \( UB_{m \times n} \)
3.  \( \text{If} \ (u = u_0 \text{ and } b = (e, t_0)) \)
4.  \( UB\_Sum = UB\_Sum + f_{\text{req}}(u, b) \)
5.  **End If**
6. **End Foreach**
7. **Foreach** \( (e \) in \( G.E \))
8.  \( \text{If} \ f_{\text{req}}(u, b) \geq 0 \)
9.  Set log-inversed probability for \( e \) to \( \ln \frac{UB\_Sum + a \cdot d}{f_{\text{req}}(u, b) + a} \) \( \text{ // } b = (e, t_0) \)
10. **Else If** \( (b \text{ never happened } || f_{\text{req}}(u, b) < 0) \)
11. Set log-inversed probability for \( e \) to \( \ln \frac{UB\_Sum + a \cdot d}{a} \)
12. **End If**
13. **End Foreach**

---

**Figure 4.6 Pseudo-code of calculating user’s log-inversed appearance behaviour probability on a road segment.**

If user \( u \) and time \( t \) are given, each road segment \( e_i \) in road network \( G \) corresponds to appearance behaviour \( b_i = (e_i, t) \). In the other words, each edge \( e_i \) in \( G \) can be regarded as weighted with log-inversed appearance behaviour probability \( \ln \frac{1}{P(u,b_i)} \).
Since \( \ln L = \sum_{i=1}^{n} \ln \frac{1}{P(u,b_i)} \), the minimization of \( L \) finds the route from origin vertex \( o \) to destination vertex \( d \) with the least weight of the appearance behaviour probability in the road network, which can be solved with Dijkstra’s algorithm. Dijkstra’s algorithm involves the continuous calculation of the shortest path, which starts at the origin and extends to other vertices until the destination is reached. There is a requirement in Dijkstra’s algorithm that the weight in each edge must be non-negative, and the log-inversed appearance behaviour probability value on each edge satisfies the requirement.

### 4.7 CTRR+: Appearance Behaviour Probability Integrating Distance

Cold start is challenging for recommendations. In our case, cold start users are users who do not have any trajectories in the dataset. If user \( u_0 \) has not had any appearance behaviour extracted from historical GPS trajectories, latent factor vector \( p_{u_0} \) of the user cannot be directly calculated with MF. In CTRR+, the cold start user’s latent factor vector is estimated using the average value of all other users’ latent factor vectors. Specifically:

\[
p_{u_0} = \frac{1}{n} (p_{u_1} + p_{u_2} + \ldots + p_{u_n})
\]  \hspace{1cm} (4.15)

\[
\hat{frq}(u_0, b_j) = q_{b_j}^T p_{u_0}
\]  \hspace{1cm} (4.16)

where \( n \) is the number of users who have appearance behaviours, \( p_{u_1}, p_{u_2}, \ldots, p_{u_n} \) are users’ latent factor vectors learned with MF, and \( q_{b_j} \) is the latent factor vector of appearance behaviour \( b_j \). Therefore, the estimated appearance behaviour frequency of cold start user \( u_0 \) for appearance behaviour \( b_j \) is \( \hat{frq}(u_0, b_j) \).
CTRR studies the appearance behaviour probability on each road segment and recommends the route with the highest appearance behaviour probability that the user would prefer based on the naïve Bayes model. However, distance is always a key factor for travel route recommendation. The calculation of the appearance behaviour probability along a route in CTRR may be affected with the formation of the road network. Therefore, CTRR is extended to CTRR+, in order to integrate distance with the appearance behaviour probability. The new appearance behaviour probability $P'(u, b)$ of user $u$ for appearance behaviour $b$ on road segment $e$ at time interval $t$ is redefined as:

$$P'(u, b) = (P(u, b))^{\text{Dist}(e)\alpha}, \text{ where } b = (e, t) \quad (4.17)$$

where $P(u, b)$ is the appearance behaviour probability in CTRR, $\text{Dist}(e)$ is the length of road segment $e$, and $\alpha$ is the power parameter of the distance. Therefore, in CTRR+, the appearance behaviour probability on a road segment considers both distance and the estimated road segment preferences of users.

Based on Equation (4.17), the objective function can be changed to:

$$\ln L = \ln \prod_{i=1}^{n} \left(\frac{1}{P(u, b_i)}\right)^{\text{Dist}(e_i)\alpha} = \sum_{i=1}^{n} \text{Dist}(e_i)\alpha \ln \frac{1}{P(u, b_i)}$$

$$= \sum_{i=1}^{n} w_i \ln \frac{1}{P(u, b_i)}, \quad w_i = \text{Dist}(e_i)\alpha \quad (4.18)$$

To minimize Equation (4.18), a shortest weighted log-inversed appearance behaviour probability route where the weight is the distance on each road segment should be determined. The rest of the CTRR+ procedure is the same as CTRR, except the calculation for the appearance behaviour probability.
4.8 Experiment

4.8.1 Dataset

In this section, the performances of the proposed CTRR and CTRR+ methods are evaluated with a real world dataset. The methods were implemented in C#. The experiments were conducted on a 2.5 GHz Core i7 PC with 16 GB of RAM. Users’ travelling GPS trajectories were obtained from the GeoLife trajectory dataset, which contained 17,621 trajectory files of 182 users. Each trajectory included user identification, travel mode (car, bus, subway, walk) and a set of GPS readings.

Trajectories of 22 users travelling by car were extracted and divided into 728 GPS traces using the entropy-based method previously discussed. The study area was in the central district of Beijing, China, ranging from 39.69 °N to 40.11 °N and from 116.09 °E to 116.62 °E. Totals of 43,381 road segments and 38,485 nodes were in the road network of the area. The travelling behaviours were generated for each user using intervals of one hour.

Figure 4.7 presents the distribution of the length of the GPS traces in the experiment’s dataset. It can be observed that the lengths of most GPS traces were around 20 km. GPS Traces of less than 10 km and more than 40 km accounted for 22.4% and 10.7% of the dataset, respectively, while GPS traces with lengths between 10 km and 40 km accounted for 66.9% of the dataset.
For each of the following experiments, the datasets were separated into training datasets and testing datasets. The testing dataset contained two tenth randomly chosen GPS traces of each user; and, the training dataset contained all the remaining GPS traces.

### 4.8.2 Performance measures

Both precision and recall were used in the experiments to evaluate the performance of the recommended travel routes. Since the lengths of road segments are different, the precision and recall were defined in terms of the number of road segments and the distance. They were calculated using the following definitions.

Precision of the travel route recommendation:
The precision of the recommended travel route was measured with the percentage of the correct road segments in the recommended route.

Recall of the travel route recommendation:

\[
\text{Recall}_{\text{road segment}} = \frac{\# \text{ of correct recommended road segments}}{\# \text{ of road segments on true route}}
\]

\[
\text{Recall}_{\text{distance}} = \frac{\text{distance of correct recommended road segments}}{\text{distance of true route}}
\]

The recall of the recommended travel route was measured with the percentage of the correctly recommended road segments in the true route. The true route is the route that the user actually travelled based on trajectories.

The higher the precision and recall results of the experiment, the better the performance of the travel route recommendation.

4.8.3 Experimental result

4.8.3.1 Evaluation of recommendation

This experiment evaluated the performances of CTRR and CTRR+ and compared them with the shortest path route method. The number of latent factors was set at 10, the regularization parameter \( \lambda \) was set at 0.01, and the smoothing parameter \( a \) was set as 0.01.
There were 588 traces in the training dataset and 140 traces in the testing dataset. Figure 4.8 illustrates the precision and recall values of three methods.

The results show that both the CTRR and CTRR+ methods outperformed the shortest distance path method in both precision and recall. Among the three methods, CTRR+ had the highest precision values: 66.3% in terms of the distance and 63.7% in terms of the number of road segments, which were improvements over the shortest distance path method of about 28% and 27.3%, respectively. The precision values of CTRR were 43.6% in terms of distance and 47.5% in terms of the number of road segments, which were improvements over the shortest distance path of around 5.8% and 11.1%, respectively.

Similar observations were made for the recall values. CTRR+ had the highest recall values: 55% in terms of the distance and 52.2% in terms of the number of road segments. The recall values of CTRR were 36.7% and 33.5%, respectively; and, the shortest distance path method had the recall values of 28.2% and 26.7%, respectively.

These results demonstrate that the route with the largest appearance behaviour probability was preferred by users compared to the shortest path, indicating that users choose a path not only considering the distance, but with comprehensive consideration of many factors. Nonetheless, CTRR+ achieved better results than CTRR, suggesting that distance plays a very important role in route recommendation and that integration of the appearance behaviour probability with distance can improve the effectiveness of travel route recommendation.
Figure 4.8 Performance evaluation of three methods in precision (a) and recall (b).

It then took a close look at the route recommendation results for some users. Figure 4.9 shows the route recommendation results of the three methods and the actual route taken by user #3 in time interval $t = 8\text{am} - 9\text{am}$. In this figure, the route recommendation result of CTRR+ (shown in circle line) had the best match with the real route taken by user #3 (shown in gray line), when compared to CTRR and the shortest path method. In this real case, user #3’s real route almost had no overlap with the shortest distance route.
4.8.3.2 Performance for cold start users

This thesis also investigated the performances of the shortest distance path and CTRR+ method on cold start users. In our dataset, 9 out of the 22 users had less than 10 trajectories. For each of the 9 users, all trajectories of the user were removed out of the training set for each experiment and used instead as the testing set. As previously discussed, the appearance behaviour frequency of cold start users is estimated by the average of all other users’ appearance behaviour frequencies in CTRR+. Therefore, in this experiment, the appearance behaviour frequencies of 21 users were averaged as the estimated appearance behaviour frequencies of a cold start user.

The precision and recall values used in the experiment were the averages of all 9 cold start users’ precision and recall measures of route recommendation by the shortest distance path method and CTRR+. The smoothing parameter $a$ was set as 0.01, the regularization parameter $\lambda$ was set at 0.01, and the number of latent factor was set at 10. The results are shown in Figure 4.10.
Figure 4.9 Route recommendation results of three methods and the real route.
It can be observed that CTRR+ worked well for the 9 cold start users. The shortest distance path method had precision values of 50.8% and 54.7%, in terms of the number of road segments and the distance, respectively. The corresponding precision values for CTRR+ were 58.3% and 62.1%, which were improvements on the shortest distance path method by 7.6% and 7.4%, respectively. The recall values for the shortest distance path method were 42.3% and 42.7%, respectively. For CTRR+, the recall values were 45.6% and 47.1%, respectively, which were improvements of 3.3% and 4.4%, respectively, over the short distance path method.

Therefore, even if the system does not have any information about the user, the proposed CTRR+ method can be an effective travel route recommendation for the user by taking advantage of other users’ historical travel information.

Figure 4.11 shows a real case of the route recommendation result of cold start user #2. It can be observed in this figure that the route provided by CTRR+ (circle line) had a better match with the true travel route by the cold start user (gray line).

4.8.3.3 Performance versus size of the time interval

In this thesis, the appearance behaviour of a user is defined as the tuple of a road segment the user is traveling on and the current time interval. With a smaller time interval size, users’ travel behaviors can be more precisely described. Nonetheless, a smaller size of the time intervals may also result in a sparser user-appearance behaviour matrix. This experiment studies the relationship among the size of the time interval and performance of the model. The experimental results with respect to different time interval sizes are shown in Table 4.2. In the table, sparsity of the user-appearance behaviour matrix is defined as
the number of unknown-valued elements divided by the total number of elements of the matrix (e.g., \( m \times n \) for an \( m \times n \) matrix).

Figure 4.10 Precision (a) and recall (b) for cold start users.
Figure 4.11 Example of route recommendation result for a cold start user.
Table 4.2 Performances of CTRR with respect to different time interval sizes

<table>
<thead>
<tr>
<th>Time intervals in one day (Size (min.))</th>
<th>User-appearance behaviour Matrix</th>
<th>CTRR Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Num.</td>
<td>Num. of known elements</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>8</td>
</tr>
</tbody>
</table>

As shown in the table, the size of the time interval increased from 30 minutes to 180 minutes. The performance of CTRR indicated by the values of precision and recall became better with the increase of time interval size from 30 minutes to 90 minutes. One possible reason is that the sparsity of the user-appearance behaviour matrix decreases as shown in the table, which means more information is available when estimating the unknown travel behaviors. Therefore, it results in a larger precision and recall of CTRR. However, when time interval kept increasing from 90 minutes to 180 minutes, the precision and recall values of CTRR decreased continuously though the sparsity of user-appearance behaviour matrix continued getting smaller. This is because of the fact that the appearance behaviour defined by larger time interval cannot describe users’ travel preferences effectively.

4.8.3.4 Performance of CTRR+ versus the Power Parameter of Distance

This experiment is conducted to study the performance of CTRR+ with respect to the power parameter $\alpha$ on distance. Different value of $\alpha$ will lead to the distinct weights
during personalized route recommendation in CTRR+. In this experiment, the power parameter $\alpha$ is varied from 0 to 2 with a step of 0.1, and the experiment result is shown in Figure 4.12.

It can be observed that, with the increase of $\alpha$ from 0 to 2, the performance of CTRR+ displayed a trend of rise first and then fall. When $\alpha$ is set 0, the precision values are 52.5% and 49.0% in terms of the number of road segments and the distance, respectively; with the increase of $\alpha$, the precision values gradually increased, and reached to the summit of 65.6% and 67.0% respectively when $\alpha$ was set to 0.8; then, the precision
values declined with the increase of $\alpha$. A similar pattern exists for the recall values. As $\alpha$ reached to 0.8, the recall values reach the maximum value as 52.4% and 55.9% in terms of the number of road segments and the distance, respectively. The experiment results exhibited that the power parameter on distance has a large influence on the route recommendation result. Greater value of $\alpha$ assigns larger influence on distance. However, an optimal value $\alpha$, making a reasonable trade-off between distance and the log-inversed appearance behaviour probability, will improve the quality of route recommendation. The experiment results showed that the value of $\alpha$ around 0.8 can improve the performance of CTRR+ for this dataset. The setting of $\alpha$ are different and should be learned in each dataset.

4.9 Summary

In this chapter, a collaborative travel route recommendation method, called CTRR, is proposed to provide user personalized route recommendation based on users’ historical GPS trajectories. CTRR can be divided into three steps. In the first step, users’ GPS trajectories are located onto the road networks to get their historical travel routes with map matching method, and then users’ appearance behaviours are extracted from the routes. In the second step, the frequencies of users’ appearance behaviours are estimated with matrix factorization method. Besides, temporal correlation is also considered for the appearance behaviour estimation. In the last step, the probabilities of users’ appearance behaviour are calculated with Laplace smoothing method, and CTRR computes the route with the maximum probability of appearance behaviours based on naïve Bayes property. An extension of CTRR, called CTRR+, is also proposed by integrating distance with users’ appearance behaviours.
The experiment results show that both CTRR and CTRR+ outperform the shortest distance path method in both precision and recall, which indicates that studying users’ travel behaviours is significant in route recommendation. Besides, CTRR+ has achieved the best performance because it considers both distance and users’ travel behaviours.

The contribution of this chapter can be summarized as follows:

1) The concept of appearance behaviour is proposed as the basic element to describe users’ travel from GPS trajectories. To better extract and represent the appearance behaviour from the GPS trajectory data, an entropy-based histogram thresholding method is proposed to divide the trajectories into sub-trajectories.

2) The appearance behaviour frequencies of each user on a certain road segment are estimated on the basis of the MF method. User’s appearance behaviour probabilities are then calculated based on the estimated appearance behaviour frequencies. Additionally, a temporal decay function is used to describe the temporal correlations among appearance behaviours.

3) The naïve Bayes model is used to generate a route with the maximized implicit appearance behaviour probability along the route. With the naïve Bayes model, users’ appearance behaviour probabilities could be calculated efficiently, and the maximum probability route is effective for recommendation.

4) CTRR+ is proposed to improve CTRR by addressing the problem of cold start users and integrating distance with the appearance behaviour probability, which can correct the limitation of CTRR and improve the performance of travel route recommendation.
5) Case studies were conducted based on a real GPS taxi trajectory dataset from Beijing, China. The experimental results show that both the proposed CTRR and CTRR+ methods performed better for travel route recommendation than the shortest distance path method and that CTRR+ achieved the best performances of the three methods.
Chapter 5: Maximum Probability Route Recommendation

This chapter proposes an improved personalized route recommendation method, called MaP2R, to provide user a personalized travel route. Chapter 5.1 gives the motivation of MaP2R. Chapter 5.2 introduces the related definitions. Chapter 5.3 discusses the methodology of MaP2R. Chapter 5.4 evaluates the experiment results of MaP2R. Chapter 5.5 gives a summary of this Chapter.

5.1 Motivation

Personalized route recommendation is developed based on user’s personal travel preferences. A collaborative travel route recommendation method, called CTRR, has been proposed in Chapter 4 to provide personalized route recommendation based on users’ historical GPS trajectories. CTRR extracts user’s appearance behaviours from GPS trajectories and provides the route with the maximum appearance behaviour probability based on naïve Bayes property. However, CTRR only considers the probabilities of appearance behaviours in route recommendation, but ignores the dependency relationship between appearance behaviours, which may hamper the route recommendation model and produce ineffective route planning. Figure 5.1 illustrates the significance of the dependency relationship between appearance behaviours.

In this figure, three historical trips $gf \to fc, gb \to bc$ and $ab \to bc$ of a user are displayed, and the number of the trips are 3, 1 and 10, respectively. It can be observed that the user prefers to travel on the route $gf \to fc$ from $g$ to $c$ compared with the route $gb \to bc$. However, as the probability of appearance behaviour on the road segment $bc$ is highly enhanced by another trip $ab \to bc$, CTRR would recommend the route $gb \to bc$ to the user.
from $g$ to $c$ because the probability of appearance behaviour on the road segment $bc$ is large. Nonetheless, the route $gb \rightarrow bc$ is unreasonable in this case because the large probability of appearance behaviour on $bc$ is highly dependent on its previous appearance behaviour. If the previous appearance behaviour of the appearance behaviour on $bc$ is on $gb$, the probability of the route is very small and should not be recommended.

![Diagram](image)

**Figure 5.1 Dependency relationship between appearance behaviours**

The problem above of CTRR is due to the ignorance of the dependency relationship between appearance behaviours. As the appearance behaviour on the road segment $ab$ will lead to the sequential appearance behaviour on the road segment $bc$ with a large probability, but on the road segment $bg$ with a small probability. Thus, the user would prefer to travel along the trip $ab \rightarrow bg$ with a small probability.

Thus, how to extract the dependency relationship between appearance behaviours from user’s historical GPS trajectories is worthy studying. Moreover, how to use the dependency relationship between appearance behaviour to improve the performance of route recommendation should be explored.
5.2 Definitions

**Definition 5.1 Transition behaviour.** A transition behaviour \( tb = (b_i \rightarrow b_j) \) is an ordered tuple to describe that the appearance behaviour \( b_i = (e_i, t_i) \) is followed by the appearance behaviour \( b_j = (e_j, t_j) \) if \( e_i.end = e_j.start \) and \( t_i = t_j \). It can also be denoted as \( tb_{i \rightarrow j} \) in short.

Given a set of appearance behaviours \( B \), a set of transition behaviours \( TB \) can be generated by considering all possible transition behaviours between adjacent appearance behaviours in \( B \). In other words, for \( \forall \, tb_k \in TB, \, tb_k = (b_i \rightarrow b_j) \) where \( b_i \in B \) and \( b_j \in B \). \( frq(u, tb_k) \) is the frequency of the transition behaviour \( tb_k \) by user \( u \). In the following discussion, travel behaviour will be used to refer to the above two types of behaviours for conciseness.

<table>
<thead>
<tr>
<th>User</th>
<th>Transition behaviour</th>
<th>Time interval</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>( b_1 \rightarrow b_2 )</td>
<td>7am-8am</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( b_1 \rightarrow b_3 )</td>
<td>7am-8am</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>( b_4 \rightarrow b_5 )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_5 \rightarrow b_6 )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_6 \rightarrow b_7 )</td>
<td>5pm-6pm</td>
<td>3</td>
</tr>
<tr>
<td>( B )</td>
<td>( b_2 \rightarrow b_8 )</td>
<td>7am-8am</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( b_9 \rightarrow b_{10} )</td>
<td>5pm-6pm</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.1 summarizes the transition behaviours of the two users from Figure 4.1. User A exhibited transition behaviours of \( tb_1 = (b_1 \rightarrow b_2, 7am - 8am) \), \( tb_2 = (b_1 \rightarrow b_3, 7am - 8am) \), \( tb_3 = (b_4 \rightarrow b_5, 5pm - 6pm) \), \( tb_4 = (b_5 \rightarrow b_6, 5pm - 6pm) \) and \( tb_5 = (b_6 \rightarrow b_7, 5pm - 6pm) \). The frequencies of the transition behaviours by user A were \( frq(A, tb_1) = 5 \), \( frq(A, tb_2) = 1 \), \( frq(A, tb_3) = 3 \), \( frq(A, tb_4) = 3 \) and
$frq(A, tb_5) = 3$. Similarly, the transition behaviours of user B were $tb_6 = (b_2 \rightarrow b_8, 7am - 8am)$, $tb_7 = (b_9 \rightarrow b_{10}, 5pm - 6pm)$, and $frq(B, tb_6) = 3, frq(B, tb_7) = 5$.

### 5.3 Maximum Probability Route Recommendation Method

This section proposes a personalized maximum probability route recommendation (MaP2R) algorithm and discuss it in detail. As mentioned, MaP2R assumes that the route a user actually takes is preferred by the user over any other route he/she could have taken between the same endpoints. Therefore, MaP2R extracted and estimated the probabilities of travel behaviours (appearance behavior and transition behaviour) on the road network from the users’ historical GPS trips and deals with the personalized route recommendation problem by searching route with the maximum travel behaviour probability. Specifically, given the origin $o$ and the destination $d$, $P(R|u, t)$ is the probability of the route $R$ which may be preferred by the given user $u$ at the time $t$ in the road network, can be represented as follows:

$$P(R|u, t) = P(e_1, e_2, e_3, ..., e_n|u, t) = P(e_1, e_2, e_3, ..., e_n, t|u) / P(t|u) \quad (5.1)$$

Where $e_1, start = o$ and $e_n, end = d$. Since the probability $P(t|u)$ is constant when $u$ and $t$ are given, to find the personalized maximum probability route is equal to maximize $P(e_1, e_2, e_3, ..., e_n, t|u)$.

$$P(e_1, e_2, e_3, ..., e_n, t|u) = P(b_1, b_2, b_3, ..., b_n|u), \text{where } b_i = (e_i, t) \quad (5.2)$$

According to the rule of conditional probability,

$$P(b_1, b_2, b_3, ..., b_n|u) = P(b_1|u)P(b_2|b_1, u) ... P(b_n|b_1, ..., b_{n-1}, u) \quad (5.3)$$
As Markov property is often used to describe the behaviours in movement models (Letchner et al. 2006, Chen et al. 2011, Dai et al. 2015), this study assumes the probability of user’s current appearance behaviour $b_i$ depends on the last appearance behaviour $b_{i-1}$. Therefore, Equation (5.3) becomes:

$$P(b_1, b_2, b_3, ..., b_n | u) = P(b_1 | u) ... P(b_i | b_{i-1}, u) ... P(b_n | b_{n-1}, u)$$  \hspace{1cm} (5.4)

where $P(b_1 | u)$ is the appearance behaviour probability starting from the origin $o$ by $u$, and $P(b_i | b_{i-1}, u)$ is the transition probability from the appearance behaviour $b_{i-1}$ to the next travel behaviour $b_i$ and is represented as $P(tb_{i-1} \rightarrow i | u)$ for conciseness in the below. Therefore, the personalized travel route recommendation problem can be defined as follows:

**Definition 5.2 Personalized maximum probability route recommendation problem.** Given time interval $t$, origin vertex $o$ and destination vertex $d$, the personalized maximum probability route recommendation problem is to find the maximum probability route $R = (e_1, e_2, e_3, ..., e_n)$ in the road network $G = (V, E)$, $e_1.\text{start} = o$ and $e_n.\text{end} = d$ so that

$$R = \text{Argmax}_{e_1, ..., e_n \in E} (P(b_1 | u)P(tb_1 \rightarrow 2 | u) ... P(tb_{n-1} \rightarrow n | u))$$  \hspace{1cm} (5.5)

MaP2R includes four steps: data preparation, frequency calculation for travel behaviours, probability estimation for travel behaviour and maximum probability travel behaviour route search. In data preparation, trajectories are firstly split into sub-trajectories that are trips with origin and destination points by using the method in (Yang et al. 2015). The second task of the data preparation step is to match the GPS trips to the road network by applying the map matching method. After matching trips to the road network, users’
travel behaviours are extracted from the routes and their frequencies are counted. In the following, it will focus on the next three steps of MaP2R.

5.3.1 Estimation for transition behaviour frequency

As mentioned, users generally travel on very limited routes daily, covering only a small number of road segments in a city. To estimate the frequency of missing travel behaviours of each user, matrix factorization (Koren et al. 2009) is used in this study. The first step is the generation of user-appearance behaviour matrix and the user-transition behaviour matrix.

**Definition 5.3. User-Appearance Behaviour Matrix.** Given a set of appearance behaviours \( B = \{ b_1, b_2, ..., b_l \} \) and a set of \( m \) users \( U = \{ u_1, u_2, ..., u_m \} \), the pairs of \((u, b)\) are used to construct a user-appearance behaviour matrix \( UB_{m \times l} \). The element in the user-appearance behaviour matrix \( UB_{m \times l} \) is the frequency of the pair \((u_i, b_j)\), i.e. \( frq(u_i, b_j) \), denoted as \( UB_{i,j} = frq(u_i, b_j) \).

To estimate the frequency of missing appearance behaviour of each user, users and their appearance behaviours are characterized by two vectors of latent factors. To be more specific, each user \( u_i \) is associated with a vector of latent factors, \( p_{u_i} = (f_{u_i}^1, f_{u_i}^2, ..., f_{u_i}^k) \), and each appearance behaviour \( b_j \) is associated with vector \( q_{b_j} = (f_{b_j}^1, f_{b_j}^2, ..., f_{b_j}^k) \), where \( k \) is the length of the vectors. The predicted frequency \( \hat{UB}_{i,j} \) of appearance behaviour \( b_j \) by user \( u_i \) is approximated by the dot product of \( p_{u_i} \) and \( q_{b_j} \):

\[
\hat{UB}_{i,j} = q_{b_j}^T p_{u_i} \tag{5.6}
\]
The vectors of latent factors are learned by minimizing the regularized squared error of the set of known appearance behaviour frequencies:

\[ L = \sum_{(u_i, b_j) \in S} (UB_{i,j} - q_{b_j}^T p_{u_i})^2 + \lambda \left( \| q_{b_j} \|^2 + \| p_{u_i} \|^2 \right) \]  

(5.7)

where \( S \) is the set of the \((u_i, b_j)\) pairs for which appearance behaviour \( b_j \) by user \( u_i \) is known, \( UB_{i,j} \) is the frequency of appearance behaviour \( b_j \) by user \( u_i \), and the constant \( \lambda \) controls the extent of regularization. After learning the vectors of latent factors \( p_{u_i} \) and \( q_{b_j} \) with alternating least squares method, the frequencies of user’s appearance behaviours can be predicted in Equation (5.7). Similarly, the matrix factorization for estimation of users’ transition behaviour frequencies can be implemented after constructing a user-transition behaviour matrix.

### 5.3.2 Estimation for transition behaviour probability

The initial appearance behaviour probability describes the probability \( P(b|u) \) of the first appearance behaviour that a user would take given a starting vertex \( o \) in the road network. Given the set \( S_o \) of the appearance behaviours starting from \( o \), the probability \( P(b|u) \) of the appearance behaviour \( b \) starting from \( o \) by user \( u \) can be calculated as the frequency of the appearance behaviour \( b \) over the total frequency of the possible initial appearance behaviours \( b_i \in S_o \).

There may, however, exist very few road segments starting from \( o \) where users have never travelled, according to their historical GPS trajectories. In this case, the Laplace smoothing method is used to estimate the probability of the missing initial appearance behaviours in Equation (5.8), in order to avoid assigning a zero probability to any initial
appearance behaviour, so that it is possible for users to travel to any road segment in the road network.

\[
P(b|u) = \begin{cases} 
\frac{\hat{frq}(u,b) + a}{\sum_{b_i \in S_0} \hat{frq}(u,b_i) + a}, & \hat{frq}(u,b) > 0 \\
\frac{\sum_{b_i \in S_0} \hat{frq}(u,b_i) + a}{\sum_{b_i \in S_0} \hat{frq}(u,b_i) + a}, & \text{otherwise}
\end{cases}
\]

where \(P(b|u)\) is the probability of \(u\)'s initial appearance behaviour \(b\) on road segment \(e\) at time \(t\). For any appearance behaviour \(b_i = (e_i, t)\) in the set \(S_o\), \(e_i, start = o\). \(\hat{frq}(u, b_i)\) is the estimated appearance behaviour frequency using matrix factorization and \(a\) is the smoothing parameter.

Given the current appearance behaviour \(b_i\) and the set \(S'\) of the next appearance travel behaviours, the transition behaviour probability \(P(tb_{i \rightarrow j}|u)\) measures the likeliness of transferring from the current appearance behaviour \(b_i\) to the next adjacent appearance behaviour \(b_j\) by the user \(u\).

If all transition behaviours in \(S'\) have never been conducted by any user, a uniform value \(c\) will be assigned to the probability of each transition behaviour in \(S'\) based on priori knowledge.

\[
P(tb_{i \rightarrow j}|u) = c
\]

Otherwise, the transition probability from appearance behaviour \(b_i\) to \(b_j\) is estimated as the frequency of the transition behaviour \(tb_{i \rightarrow j}\) over the total frequency of the possible transition behaviours \(tb_{i \rightarrow k}\) and \(b_k \in S'\).

Similarly, to avoid assigning a zero probability to any transition behaviour, Laplace smoothing method is utilized as follows:
\[ P(t_{b_{i\rightarrow j}} | u) = \begin{cases} \frac{\hat{f}(u, t_{b_{i\rightarrow j}}) + a}{\sum_{k=1}^{d} \hat{f}(u, t_{b_{i\rightarrow k}}) + a}, & \hat{f}(u, t_{b_{i\rightarrow j}}) > 0 \\ \frac{a}{\sum_{k=1}^{d} \hat{f}(u, t_{b_{i\rightarrow k}}) + a}, & \text{otherwise} \end{cases} \]  

(5.10)

where \( d \) is the number of all appearance behaviours in \( S' \), \( a \) is the smoothing parameter based on a priori knowledge. Equations (5.9) and (5.10) can assign a non-zero probability to any transition behaviours that never happened in users’ GPS trajectories.

### 5.3.3 Maximum probability route search

Appearance behaviours and transition behaviours are utilized to describe the maximum probability route in Equation (5.4). In the following, a behaviour graph is defined and can be generated from the road network and probabilities of travel behaviours, which will be used for searching the maximum travel behaviour probability route.

**Definition 5.4. Behaviour Graph.** Given time \( t \), user \( u \), road network \( G \), a start vertex \( v_{\text{start}} \) and an end vertex \( v_{\text{end}} \), a behaviour graph is denoted by \( G' = (V', E') \) where the set of vertices \( V' \) includes three parts, a set of vertices \( V^* \) where each element presents an appearance behaviour of the user, the start vertex \( \{v_{\text{start}}\} \), and the end vertex \( \{v_{\text{end}}\} \), i.e. \( V' = V^* \cup \{v_{\text{start}}\} \cup \{v_{\text{end}}\} \); the set of edges \( E' \) is also constituted of three parts \( E' = E^* \cup E_{\text{start}} \cup E_{\text{end}} \), where \( E^* \) is a set of edges in which each element presents a transition behaviour from one appearance behaviour vertex to another, \( E_{\text{start}} \) includes all edges starting from \( v_{\text{start}} \) to its possible adjacent appearance behaviour vertices and \( E_{\text{end}} \) includes edges connecting from appearance behaviour vertices to \( v_{\text{end}} \). There is a weight \( \theta \) on each edge in a behaviour graph associated with travel behaviour probabilities.
Figure 5.2 gives an example of a behaviour graph from a road network. The behaviour graph describes all possible appearance behaviours and transition behaviours from a start vertex $v_{\text{start}}$ to an end vertex $v_{\text{end}}$ for the given time and user.

As defined, the personalized route recommendation problem requires the maximization of $P(b_1 | u) P(t_{b_1 \rightarrow 2} | u) P(t_{b_2 \rightarrow 3} | u) \ldots P(t_{b_{n-1} \rightarrow n} | u)$ in Equation (4). In order to find the maximum travel behaviour probability route, the multiplication of probabilities is transformed to the summarization format required by the typical route planning algorithms, let $L = \frac{1}{P(b_1 | u) P(t_{b_1 \rightarrow 2} | u) P(t_{b_2 \rightarrow 3} | u) \ldots P(t_{b_{n-1} \rightarrow n} | u)}$; thus, the problem in this thesis is equivalent to find the minimization of $L$.

Taking the logarithm for both sides:

$$\ln L = \ln \left( \frac{1}{P(b_1 | u)} \prod_{i=1}^{n-1} \frac{1}{P(t_{b_i \rightarrow (i+1)} | u)} \right) = \ln \frac{1}{P(b_1 | u)} + \sum_{i=1}^{n-1} \ln \frac{1}{P(t_{b_i \rightarrow (i+1)} | u)} \quad (5.11)$$

Let $\vartheta(v_{\text{start}}, b_k) = \ln \frac{1}{P(b_k | u)}$, $\vartheta(b_i, b_j) = \ln \frac{1}{P(t_{b_i \rightarrow b_j} | u)}$ and $\vartheta(b_k, v_{\text{end}}) = 0$, the personalized maximum probability route recommendation problem is to find a minimum weight path in the generated behaviour graph.

Figure 5.2 An example showing the relationship between the road work and the behaviour graph.
As shown in Figure 5.2, the weights on the edge connecting the start vertex \( v_1 \) to the appearance behaviour \( b_1 \) and \( b_4 \) are \( \theta(v_1, b_1) \) and \( \theta(v_1, b_4) \) respectively, the weights on the edges connecting appearance behaviour \( b_3 \) and \( b_6 \) to the end vertex \( v_6 \) are \( \theta(b_3, v_6) \) and \( \theta(b_6, v_6) \), respectively, and the weights between two adjacent appearance behaviours \( b_i \) and \( b_j \) are \( \theta(b_i, b_j) \). To find the maximum probability route from \( v_1 \) to \( v_6 \) is equal to find the minimum weight path in Figure 5.2, which can be solved with Dijkstra’s algorithm.

### 5.4 Experiment

#### 5.4.1 Dataset

In this section, MaP2R is evaluated on a real world GPS trajectory dataset, Geolife (Zheng et al. 2010). The dataset contains 17,621 trajectory files of 182 users. Out of them, 22 drivers’ GPS trajectories are extracted and divided into 728 trips. The study area is in the central district of Beijing, China, ranging from 39.69° N to 40.11° N and from 116.09° E to 116.62° E. Totally 43,381 road segments and 38,485 nodes are in the road network of the area. The travel behaviours are generated using one hour interval.

#### 5.4.2 Sensitivity analysis on number of latent factors for matrix factorization

In MaP2R, matrix factorization (MF) is used to estimate the frequencies of travel behaviours on the road segment. The number of latent factors is one parameter of the matrix factorization. In this experiment, the effect of number of latent factors on the performance of the MaP2R is tested. Figure 5.3 shows that root mean squared error (RMSE) and running time with respect to the number of latent factors in matrix factorization. The number of latent factors ranges from 5 to 15. It can be observed that with the increase of the number
of the latent factors, the RMSE values of both the user-appearance behaviour matrix and the user-transition behaviour matrix slightly change, but then keep at a steady level while the training time for MaP2R increases with the number of latent factors. Hence, the number of latent factors is chosen as 5 in the following experiments.

![Graph](image)

**Figure 5.3** The effect of number of latent factors in MF vs. (a) RMSE and (b) training time.

### 5.4.3 Overall performance vs. route distance

This experiment evaluates the performances of MaP2R and compare with the shortest distance route method (SDR) and the collaborative travel route recommendation
method (CTRR) in terms of the trip lengths. SDR recommends the route with the shortest distance. CTRR first estimates the appearance behaviour probabilities of all road segments and then recommends a route with the highest consecutive appearance probabilities from the origin to the destination. The route of CTRR usually contains frequently traveled road segments.

Trips are first separated in the testing dataset into four different groups based on the trip length. Each distance range group contains 35 trips. The distance ranges for the four groups are: Group 1: (2.05-14.58km), Group 2: (14.58-19.21km), Group 3: (19.21 – 28.74km) and Group 4: (28.74-56.72km). The number of latent factors is set as 5, the regularization parameter $\lambda$ as 0.1, and the smoothing parameter $a$ and $c$ as 0.01, respectively. Figure 5.4 illustrates the precision and recall values of MaP2R, SDR and CTRR on four groups.

The results show that MaP2R outperforms the other two methods in both precision and recall in all four groups. When the trip is short, i.e. for trips in Group 1, the performances of the three methods do not have much discrepancy. SDR method is slightly better than CTRR method because when the distance between the origin and the destination is small, users prefer to the shortest distance path rather than passing through the high frequency road segments to reach the destination, which is reasonable in reality. With the increase of the distance of trips, MaP2R and CTRR outperform SDR because MaP2R and CTRR both consider users’ preference from historical trajectories while SDR does not consider the factor. Moreover, MaP2R outperforms CTRR by 3~18% in precision and 13~21% in recall for the Groups 2~4. The reason is that MaP2R considers user’s preference for each travel behaviour and the dependencies between the travel behaviours, which could
Figure 5.4 Performance Comparison among SDR, CTRR and MaP2R for four groups of trips with different lengths (a) precision of number of road segments; (b) recall of number of road segments; (c) precision of length of road segments; (d) recall of length of road segments.
better reflect users’ travel preference, but CTRR only considers high frequency of road segments separately. Therefore, these results demonstrate that the route from MaP2R method has a larger correspondence with users’ preference compared to the SDR and CTRR method.

5.4.4 Case study for performance

This experiment compares the route recommendations from the proposed MaP2R with the two most popular online route recommendation applications, i.e., Google Map and Baidu Map.

Figure 5.5 shows a case of Geolife dataset that a user intends to travel from the origin location Sigma mall on the Zhichun road to the destination a research institution around 10 am. The recommended route provided by MaP2R is to take 5\textsuperscript{th} ring road to get to the destination while Google Map and Baidu Map recommends user travel on the 4\textsuperscript{th} ring road and the 3\textsuperscript{rd} ring road, respectively. The route recommended by MaP2R is the same as the user’s real travel route but the routes recommended by Google Map and Baidu Map have a large discrepancy with the real route. The reason is that Google Map and Baidu Map would recommend all users the same route in the figure from Sigma mall to the research institution based on a certain criterion, but MaP2R would learn users’ preference from historical trajectories and would recommend the route which is high corresponding with users’ preferences. In fact, the target user is found indeed prefer to travel on 5\textsuperscript{th} ring road by scrutinizing this users’ historical GPS trajectories manually.
5.5 Summary

In this chapter, the maximum probability route recommendation method, called MaP2R, is proposed to provide user personalized route recommendation. MaP2R considers the dependency relationship between user’s appearance behaviours, and proposes transition behaviour to represent the dependency relationship in this thesis. MaP2R estimates the probabilities of users’ appearance behaviours and transition behaviours based on matrix factorization and Laplace smoothing method, and calculates the maximum probability route with Markov property. Experiments are conducted on the real GPS trajectory dataset to evaluate the performance of MaP2R, and the experiment results show that MaP2R outperforms CTRR and shortest distance path. Besides, a real case study of route recommendation shows that the route provided by MaP2R matches better with user’s real route preference than the routes provided by Google Maps and Baidu Maps. The better performance of MaP2R indicates that considering users’ personal route preference can provide a better route recommendation result.

The contributions of this chapter can be summarized as follows:

1) Transition behaviour is proposed to represent the dependency relationship between user’s appearance behaviours. Both appearance behaviours and transition behaviours are extracted from users’ historical GPS trajectories for the personalized route recommendation.

2) The frequency of user’s appearance behaviour and transition behavior are estimated with matrix factorization method. The maximum probability route is calculated based on user’s appearance behaviours and transition behaviours with Markov property. Besides, a behaviour graph is built to facilitate the maximum probability
route calculation.

3) Experiments are conducted on the real GPS trajectory dataset, and the experimental results show that MaP2R outperforms CTRR and the shortest distance path in both precision and recall.
Figure 5.5 Comparison among MaP2R, Google Map and Baidu Map
Chapter 6: Conclusion and Discussion

6.1 Conclusions

This thesis proposes a segment-based hidden Markov model for GPS trajectory map matching, and three personalized route recommendation methods based on GPS trajectories.

The segment-based hidden Markov model, called SHMM, improves the performance of the point-based map matching in both effectiveness and efficiency. In the preprocessing part, outliers are removed from GPS points based on the spatial proximity between a GPS point and the road network and a speed constraint. Besides, GPS points are resampled by removing some points which are located close to their previous GPS points. Next, GPS trajectory is partitioned into several GPS sub-trajectories based on the heading homogeneity, and candidate road segment sequences are searched out for each GPS sub-trajectory. Then, a segment-based hidden Markov model is built based on the generated GPS sub-trajectories and their candidate road segment sequences. In the segment-based hidden Markov model, the emission probability is calculated based on the distance between GPS sub-trajectories and their candidate road segments; the transition probability measures the reasonability of a connectivity between adjacent GPS sub-trajectories in the road network. Last, Viterbi algorithm is used to find the route of the GPS sub-trajectories with the maximum probability in the road network.

In SHMM, a distance metric, called LCS-HP, is proposed to better measure the distance between GPS sub-trajectories and road segment sequences. LCS-HP is an extension based on LCS distance metric for trajectories. Compared with other distance
metric, such as LCS, EDR and ERP, LCS-HP considers the heading information of GPS trajectories and sets a penalty on the distance if the heading between the GPS trajectory and the road segment sequence is dissimilar. Besides, LCS-HP is more robust to outliers compared to the Fréchet distance.

Experiments are conducted based on a real GPS trajectory dataset, and the experiment results show that the proposed SHMM method outperformed two baseline methods, the point-based HMM method and the incremental method in accuracy. Besides, the experiment results show that SHMM with LCS-HP outperforms SHMM with Fréchet distance, because LCS-HP distance measurement is more robust to the measurement error of GPS point, and it considers the heading similarity between the GPS sub-trajectories and the road segment sequences. Furthermore, SHMM is more efficient than PHMM because the computation cost to check the reasonability of transition between adjacent stages is decreased greatly.

This thesis proposes CTRR and CTRR+ methods for providing users with personalized travel route recommendations based on users’ historical GPS trajectories. In the data preprocessing stage, the concept of appearance behaviour was suggested to describe users’ travel preferences and extract the travel behaviours from historical GPS trajectories. Appearance behaviour describes the road segment that a user prefers to travel on at a given time. Besides, original GPS trajectories are divided into GPS traces with the time interval based trajectory segmentation method when the GPS signal is lost for a long time, which can prevent inaccurate extraction of users’ appearance behaviours from the uncertainty of GPS trajectory. To determine the time interval threshold in the time interval
based segmentation method, the entropy-based histogram thresholding method is proposed.

After GPS trajectories are divided into GPS traces, users’ appearance behaviour frequencies are then extracted from each GPS trace. Next, CTRR method estimates users’ appearance behaviour frequencies using a temporal decay function and the matrix factorization (MF) method and calculates their appearance behaviour probabilities with the Laplace smoothing method. A route is then computed for recommendation with the naïve Bayes model.

The CTRR+ method improves upon the CTRR method by taking into account cold start users and by integrating distance with the appearance behaviour probability to make route recommendations. On one hand, the appearance behaviour probability can implicitly reflect users’ preference on each road segment at a certain time, which is highly personalized. On the other hand, the estimated probability of certain appearance behaviour for some users takes advantage of information from users’ who had previously exhibited this travel behaviour.

Both CTRR and CTRR+ can provide users with personalized travel route recommendations. Different users may be recommended distinct travel routes, even if the query conditions are the same for the origin, destination and time.

To evaluate the performances of the proposed two methods, experiments were conducted based on a real GPS trajectory dataset. The results show that both CTRR and CTRR+ achieved better performances than the baseline method (i.e. the short distance path method) in both precision and recall. CTRR+ outperformed CTRR.
Last, a personalized maximum probability route recommendation method (MaP2R) is proposed to improve the performance of CTRR based on historical GPS trajectories. CTRR does not consider the sequential relationship between appearance behaviours, which is unreasonable and would recommend user the routes with low quality. To overcome the deficiency, transition behaviour is proposed to describe the sequential relationship between appearance behaviours in MaP2R. First, users’ travel behaviours (appearance behaviors and transition behaviours) are extracted from historical GPS trajectories. Matrix factorization and Laplace smoothing method is then used to estimate the frequencies and probabilities of users’ travel behaviours. Finally, the route with the maximum probability by considering both appearance behaviors and transition behaviours is recommended to the user. The experiment results show that MaP2R outperforms the shortest distance path method and the CTRR method. Moreover, a real travel route recommendation case is conducted to compare Baidu Map, Google Map and MaP2R, and the case shows that the recommendation result from MaP2R matches with user’s real preference best in the three route recommendation methods.

6.2 Future works

SHMM can be improved in the following perspectives. First, the measurement for spatial proximity between GPS sub-trajectory and road segment sequence could be further studied. LCS-HP gives penalty to road segment sequence which has the dissimilar heading with the GPS sub-trajectory, and the penalty is set based on the assumption that the length of GPS sub-trajectories and the length of road segment sequence should be similar. In future, the penalty can be further studied by investigating the heading similarity between
each road segment in road segment sequence and the GPS points in GPS sub-trajectory, which may provide a more precise penalty for the distance measurement between them. In future, other distance measurement methods, such as ERP, EDR, Fréchet distance between GPS sub-trajectory and road segment will be studied by considering the heading similarity to measure the distance between GPS trajectory and road segment sequence. Second, the threshold of heading difference used in the trajectory segmentation with heading homogeneity could be further studied. As different threshold could generate distinct GPS sub-trajectories, which may affect the performance of the segment-based map matching. Thus, an optimum threshold of heading difference for the trajectory segmentation method with heading homogeneity is worth studying. Third, the proposed segment-based method will be studied for low sampling GPS trajectories in future. The performance of SHMM will decreases as the sampling rate of GPS trajectory grows. An important reason is that the large sampling rate may result in a large distance between two adjacent GPS points so that the heading property of GPS sub-trajectories can not effectively measure the distance between GPS sub-trajectory and road segment sequence. In future, a possible solution is that when the distance between adjacent GPS points in a GPS sub-trajectory exceeds a threshold, the GPS sub-trajectory will be further divided, and the integration of point-based method and segment-based method can be considered.

The entropy-based histogram thresholding method is proposed in Chapter 4 to determine the time interval threshold for trajectory segmentation, which can reduce the uncertainty of GPS trajectories for extraction of travel behaviours. However, the proposed entropy-based histogram thresholding method does not take the contextual adaptive GPS
tracking frequency problem into account. In future, it will continue to study the trajectory segmentation method for the GPS trajectories with context-adaptive sampling rates.

CTRR, CTRR+ and MaP2R can be improved from the following perspectives.

First, CTRR, CTRR+ and MaP2R assume that traveling happens in the same time interval in the route recommendation. However, in the real world, a travel may cross multiple time intervals. There are two key problems to be considered. The first one is that the travelling time on each road segment should be estimated so that the elapsed time of travelling can be tracked. When a user travels across into the next time interval, his/her travel behaviour probabilities in the road network should be dynamically updated with respect to different time intervals. The second issue is that the transition between travel behaviours on one road segment across the adjacent time interval should be considered. As travel behaviour is assumed to be composed of a set of latent factors, one possible solution is to consider the latent factors as time-dependent, and study the transition between the latent factors over the adjacent time interval. With the time-dependent latent factors involving transition relationship over time, the frequencies and probabilities estimation of travel behaviours can be better estimated.

Second, the spatial correlation between travel behaviours should be considered in the personalized route recommendation. The travel behaviours in CTRR, CTRR+ and MaP2R have a spatial component, i.e. the location of road segment. However, the spatial correlation between travel behaviours are not considered in the current methods. For instance, travel behaviour may have its adjacent travel behaviours in the road network. Thus, if a user has had a travel behaviour before, the user may also prefer to have the adjacent travel behaviours of the preferred travel behaviour.
Third, matrix factorization is used to estimate the frequency of a user’s travel behaviours in CTRR, CTRR+ and MaP2R by considering the frequency of other users who has similar travel preference with the user. In collaborative filtering methods, matrix factorization is better with the explicit feedback of users, such as explicit rating on the routes. However, the travel behaviours extracted from GPS trajectories are implicit feedback of users. Thus, other methods associated with user’s implicit feedback, such as random walk with restart, should be tested to make a better estimation of user’s travel preference.

Fourth, CTRR, CTRR+ and MaP2R only consider users’ travel behaviours, but more factors should be integrated into the personalized route recommendation method. First, many other factors, such as weather, traffic volume, road condition, may affect users’ travel preferences and have a large impact on the recommendation result. Second, integrating these factors can provide user a real-time contextual personalized route recommendation, which is more useful for users in daily travel. Tensor decomposition is a potential method to integrate all relevant factors into a personalized route recommendation model. Tensor is a high dimensional array and each factor such as user, travel behaviours and weather can be taken as a dimension in the array. Tensor decomposition can factorize the tensor into several matrices, and each matrix is the latent factors associated with each dimension. High correspondence of latent factors of different factor dimension implies a high preference.

In CTRR+, the parameter $\alpha$ is to balance the weights between the travel behaviour probability and distance. The setting of the optimal $\alpha$ depends on the dataset, and should be learned from the dataset. A potential method for learning $\alpha$ is gradient ascent method.
which is an iterative process. The precision of CTRR+ varies with the value of $\alpha$, and the
gradient of the precision at $\alpha$ is the direction in which the precision is increased fast. Based
on gradient ascent method, $\alpha$ can take steps proportional to the gradient of the precision at $\alpha$ until the process converges. The deficiency of the method is that the precision may
approach to a local maximum.

In future, it will continue to explore the spatiotemporal correlations between travel
behaviours and integrate with the current estimation model. More experiments will be
conducted to compare CTRR, CTTR+ and MaP2R with other personalized route
recommendation methods. Besides, some route recommendation applications will be
developed to provide user a real time personalized route recommendation.

6.3 Future applications

In future, applications of personalized route recommendation system will be
developed on mobile devices (e.g. smart phones and vehicles). After a user enters an origin
and destination of a trip in the applications, a personalized route will be recommended to
the user based on the real time contextual information (e.g. traffic volume and weather)
and the user’s historical route preferences.

The recommendation system will adopt the client-server structure. The clients are
responsible for user-machine interaction and data collection. In the clients, users can make
requests for the real time route recommendations. Besides, users’ GPS trajectories and the
contextual information of the trips are collected in the clients and then uploaded into
servers. The servers are responsible for data preprocessing, users’ travel behaviours
estimation, and route computation. After users’ historical GPS trajectories are uploaded
into the servers, the probabilities of users’ travel behaviours under different contextual information are estimated and stored in the servers. When route recommendations are requested from users, the servers will compute the route and return the result to the clients.

The personalized route recommendation system should respond to the users in real time, which can be achieved by implementing two tasks. One task is calculating the probabilities of users’ travel behaviours. As users’ travel behaviours may change over time and the calculation for the probabilities of users’ travel behaviours are time consuming, this task can be processed offline and the probabilities of users’ travel behaviours can be updated weekly or monthly. The other task is route computation. The servers should estimate the routes which the users may prefer and return the recommendation results to the clients in real time.

In future, the applications of route recommendation may change with the rapid development of automotive vehicle technology.

One possible trend is that the current general route recommendation might be substituted by the unified route planning for all automotive vehicles in the whole city. The current route recommendation would recommend all vehicles to the routes with the lowest traffic volume, which may lead to traffic congestion, while the unified route planning aims to make a global traffic optimization for all automotive vehicles.

Besides, automotive vehicle technology may also have a large impact on the personalized route recommendation. For instance, as users are liberated from driving by automotive vehicle technology, many factors, such as road surface, might have smaller impact on users’ route preferences in the future. However, other factors may still have large impact on users’ route preferences, such as travelling time and fuel consumption. Hence,
the emergence of automotive vehicle technology would not change the fact that users have different route preferences. In the era of automotive vehicles, the applications of personalized route recommendation should consider both users’ personal route preference and the global optimization of the transportation.

In future, the personalized route recommendation can also be applied in many areas. For instance, the personalized route recommendation for tourism is a potential application. In the application of the personalized tourism route recommendation, in addition to users’ preferences on the scenic spots, users’ personal route preferences between the scenic spots should be also studied. As users may have different route preferences between two scenic spots, the personalized tourism route recommendation can provide a better tourism experience. Thus, an optimal personalized route recommendation for tourism should consider users’ personal preferences on both the scenic spots in a city and the route to visit the scenic spots.
Bibliography


Publications


2. Qu, B., Yang, W., **Cui, G.**, and Wang, X., “Profitable Taxi Travel Route Recommendation Based on Big Taxi Trajectory Data”. *Submitted to IEEE Transactions on Intelligent Transportation Systems (under review).*


