TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Kartographie

Labeling Spatial Trajectories in Road Network Using Probabilistic Graphical Models

Jian Yang

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.- Prof. Dr.phil.nat. Urs Hugentobler
Prüfer der Dissertation:	1. Univ.- Prof. Dr.-Ing. Liqiu Meng
	2. Univ.- Prof. Dr.-Ing. habil. Monika Sester Leibniz Universität Hannover

Die Dissertation wurde am 05.11.2015 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 25.04.2016 angenommen.

ABSTRACT

Movement has been essential to many scientific and social studies. Understanding human mobility in the urban context is of great importance to ease individual travel planning and improve the transportation infrastructure as well. With the development of positioning and communication technologies, urban mobility can be captured in the form of temporally ordered location sequences, namely spatial trajectories, at an unprecedented massive scale in nowadays modern cities. Furthermore, open data initiatives such as OpenStreetMap have transformed the way that these data are shared and accessed. More and more trajectory data are exposed to academia and industry and has been proved to be invaluable resource to comprehend urban mobility. However, these data often suffer from poor data quality and lacking semantic information. Many tasks such as map matching, activity recognition that serve the purposes of enhancing the data quality or enriching data semantic can be formulated as labeling spatial trajectories. This thesis has revisited these tasks from a unified perspective and develop probabilistic models using probabilistic graphical model (PGM) for a holistic representation of the trajectory data, in particular, in the context of urban road networks.

The key issue precedent to the model development is to understand the uncertainty that resides in the trajectory data and the specific tasks. In the comparative study of tasks of labeling spatial trajectories, two types of tasks are identified, namely localization and behavioral classification. They distinct in label size and the semantics that the labels bear. More importantly, the comparison has facilitated the comprehension of three sources of the uncertainty in labeling spatial trajectories. And map matching of low sampling rate GPS trajectories and taxi status inference, are selected in the thesis as core tasks for the model development.

With the insight from the comparison of labeling spatial trajectories, the tasks are formulated as computing the probability of the label assignments given spatial trajectory data. This discriminative formulation eases the probabilistic modeling by allowing inducing arbitrary non-dependent features to compute the overall probability mass. The modeling follows the standard procedure of application of PGM. First, the graphical structure is designed to represent the structural dependency among the data instances and the labels. Specialized treatments are made for localization task, as it requires to deal with huge set of labels. Secondly, a large set of features are induced based on both empirical evidences and domain knowledge. Each of these features are associated with parameters that need to be estimated from the training data. Parameter-tying strategy are discussed for practical concerns. Thirdly, inference and learning are developed based on the graphical structure. In order to avoid overfitting the data and to find most relevant features in the model, the training objective is regularized using ℓ_1 norm.

To evaluate the proposed models for study tasks, two test datasets are derived from the real world dataset, Shanghai taxi floating car data (FCD). The implementations consist of label data preparation for localization task, feature extraction in spatial database and model development for training and testing. Experiments on test datasets have shown that the proposed models can reach the equivalent performance to the state-of-the-art in solving the tasks of labeling trajectories and exhibit merits in providing comprehensive representation and reliable label assignments.

ZUSAMMENFASSUNG

Bewegung ist für viele wissenschaftliche und soziale Studien unerlässlich. Das Verständnis menschlicher Mobilität im urbanen Kontext ist von großer Bedeutung, um die Planung individueller Reiserouten zu unterstützen und zu der Verbesserung von Verkehrsinfrastrukturplanungen beizutragen. Mit der Entwicklung der Positionierungs- und Kommunikationstechnologien kann urbane Mobilität in Form von zeitlich geordneten Standortsequenzen, nämlich räumlichen Trajektorien, in den heutigen modernen Städten mit einem beispiellos enormen Ausmaß erfasst werden. Darüber hinaus haben Open Data Initiativen wie OpenStreetMap die Art und Weise wie diese Daten gemeinsam genutzt und abgerufen werden gänzlich verändert. Immer mehr Trajektorien werden der Wissenschaft und Industrie zur Verfügung gestellt, die sich als wertvolle Ressource erweisen um urbane Mobilität zu verstehen. Allerdings besitzen diese Daten häufig eine schlechte Datenqualität sowie mangelnde semantische Informationen. Viele Aufgaben wie Map Matching und Aktivitätserkennung die Zwecke, die der Verbesserung der Datenqualität oder der Datenanreicherung mit Semantik dienen, können als Labeling räumlicher Trajektorien bezeichnet werden. Ziel dieser Arbeit ist es, diese Aufgaben aus einer gesamthaften Sicht zu überdenken und Wahrscheinlichkeitsmodelle mit einem Probabilistisch Graphischen Modell (PGM) zu entwickeln für eine ganzheitlichen Darstellung der Trajektorien, insbesondere im Kontext urbaner Straßennetze.

Die zentrale Frage, die der Modellentwicklung vorangeht, ist es die Unsicherheit zu verstehen, mit denen die Trajektorien Daten und die spezifischen Aufgaben behaftet sind. In der Vergleichsstudie von Aufgaben zum Labeling räumlicher Trajektorien werden zwei Arten von Aufgaben identifiziert, nämlich die Lokalisierungs- sowie die Verhaltensklassifizierung. Diese beiden Aufgaben unterscheiden sich sowohl in der Labelgröße, als auch in der Semantik, die die Labels innehaben. Darüber hinaus fördert der Vergleich das Verständnis der Unsicherheit aller drei Quellen beim Labeling räumlicher Trajektorien. Zwei Aufgaben, Map Matching einer niedrigen GPS-Trajektorien Abtastrate sowie Taxistatus Schlussfolgerungen wurden ausgewählt, um die Modellentwicklung voranzubringen.

Mit den Erkenntnissen aus dem Vergleich des Labeling räumlicher Trajektorien werden die Aufgaben, nach der Berechnung der Wahrscheinlichkeit der Labelzuordnung, gegebener räumlichen Trajektorien formuliert. Diese unterschiedliche Formulierung ermöglicht der probabilistischen Modellierung aus induzierten beliebig unabhängigen Features die Gesamtwahrscheinlichkeit zu berechnen. Die Modellierung folgt dem Standardverfahren für die Anwendung des PGMs. Zuerst wird die graphische Struktur entworfen, um die strukturelle Abhängigkeit zwischen den Dateninstanzen und den Labels darzustellen. Spezialisierte Anwendungen werden für die Lokalisierungsaufgabe kreiert, da diese riesige Mengen von Labels verarbeiten muss. Zweitens wird eine große Auswahl an Features induziert, basierend sowohl auf empirischen Beweisen wie auch auf Fachwissen. Jedes dieser Features wird mit Parametern, die aus den Trainingsdaten geschätzt werden müssen, verbunden. Die Parametergebundenen Strategien werden für praktische Belange diskutiert. Drittens werden Schlussfolgerungen und Lernprozesse auf Basis der graphischen Struktur entwickelt. Um eine Überanpassung der Daten zu verhindern und auch die wichtigsten Features im Modell zu finden, wird das Trainingsziel mit der ℓ_1 Norm reguliert.

Um die entwickelten Modelle die Anwendung in Aufgaben zu evaluieren, werden zwei Testdatensätze aus dem realen Datensatz, Shanghai Taxi Floating Car Data (FCD) abgeleitet. Die Implementierungen bestehen aus der Labeldatenvorbereitung für Lokalisierungsaufgaben, Merkmalsextraktion in der räumlichen Datenbank und Modellentwicklung für Training und Prüfung. Experimente mit Testdatensätzen haben gezeigt, dass die vorgeschlagenen Modelle eine äquivalente Leistung im Vergleich zu neueren Methoden bei der Lösung von Aufgaben des Labeling von Trajektorien und der Präsentation eines Mehrwerts bei der Bereitstellung einer umfassenden und zuverlässigen Labelzuordnung erreicht.

TABLE OF CONTENTS

ABSTRACT .... **I**

ZUSAMMENFASSUNG .... **III**

TABLE OF CONTENTS .... **V**

ABBREVIATIONS .... **VII**

LIST OF TABLES .... **VIII**

LIST OF FIGURES .... **IX**

CHAPTER 1. .... **1**

- 1.1. MOTIVATION 1
- 1.2. GOAL 3
- 1.3. THESIS STRUCTURE 4

CHAPTER 2. .... **5**

- 2.1. SPATIAL TRAJECTORIES IN ROAD NETWORK 5
  - *2.1.1. Moving Objects and Spatial Trajectories 5*
  - *2.1.2. Urban Road Network 7*
- 2.2. LABELING SPATIAL TRAJECTORY 9
  - *2.2.1. Knowledge Discovery in Trajectory Data: A Retrospective Overview 10*
  - *2.2.2. Labeling Spatial Trajectories 11*
- 2.3. STATE-OF-THE-ART OF STUDY TASKS 17
  - *2.3.1. Map Matching 17*
  - *2.3.2. Inferring Taxi Status 24*

CHAPTER 3. .... **27**

- 3.1. UNCERTAINTY IN LABELING SPATIAL TRAJECTORIES 27
  - *3.1.1. Imprecise Positioning 27*
  - *3.1.2. Sampling 28*
  - *3.1.3. Behavioral Dynamic 30*
- 3.2. DISCRIMINATIVE MODELS FOR SEQUENCE LABELING 32
  - *3.2.1. Probabilistic Graphical Models 32*
  - *3.2.2. Generative versus Discriminative Classifiers 33*
  - *3.2.3. Conditional Random Fields 34*
  - *3.2.4. Feature Extraction 36*
  - *3.2.5. Inference on Graphical Models 37*
  - *3.2.6. Discriminative Learning 38*

CHAPTER 4. .... **39**

- 4.1. CHAIN STRUCTURED CRF 40
- 4.2. STATE GENERATION 44
  - *4.2.1. State Generation Workflow 44*
  - *4.2.2. Redundancy Elimination 47*
- 4.3. FEATURES EXTRACTION 48
  - *4.3.1. Features 48*
  - *4.3.2. Parameter Tying 50*

4.4. INFERENCE ON CHAIN 51
4.5. PARAMETER ESTIMATION 53
4.5.1. Maximum Likelihood 53
4.5.2. Gradient Ascent 55
4.5.3. Learning with Partially Observed Model 56
4.5.4. Feature Selection via ℓ_1 Regularization 57
4.6. CHAIN CRF FOR BEHAVIORAL CLASSIFICATION 58
4.6.1. Inferring Taxi Status 58
4.6.2. Model 59
4.7. SUMMARY 61

CHAPTER 5. 63

5.1. RAW DATASETS 63
5.1.1. Shanghai Taxi FCD 63
5.1.2. OSM Road Network 64
5.2. IMPLEMENTATIONS 66
5.2.1. Labelling Using An Interactive Routing Tool 66
5.2.2. Feature Extraction in Spatial Database 68
5.2.3. Path: A Matlab Toolbox For Labeling Spatial Trajectories 69
5.3. LABELING TASK I – MAP MATCHING OF LOW-SAMPLING RATE GPS TRAJECTORIES 70
5.3.1. Experiment Setup 70
5.3.2. Experiment Results 72
5.3.3. Case Study 75
5.4. LABELING TASK II – INFERRING TAXI STATUS 77
5.4.1. Experiment Setup 77
5.4.2. Experiment Results 78
5.5. DISCUSSIONS 82

CHAPTER 6. 83

6.1. CONCLUSIONS 83
6.2. OUTLOOK 84

REFERENCES 91

ACKNOWLEDGEMENTS 99

CURRICULUM VITAE 101

ABBREVIATIONS

OSM	OpenStreetMap
PGM	Probabilistic Graphical Model
UMG	Undirected Graphical Model
HMM	Hidden Markov Model
CRF	Conditional Random Fields
API	Application Programming Interface
SQL	Structured Query Language

LIST OF TABLES

Table 2.1 Clarification of spatial trajectory related terminology. 7
Table 2.2 Comparison of example labelling tasks... 14
Table 3.1 Generative versus Discriminative Models. 34
Table 4.1. Features used in the CRF for map matching. 50
Table 4.2 Features used in the CRF for status inferring. 61
Table 5.1 Attributes of Shanghai taxi FCD records. 64
Table 5.2 Attributes of routable OSM road network data. 65
Table 5.3 Specification of training/test set.. 70
Table 5.4 Evaluations of map matching results on GPS trajectories of 120s sampling interval. 74
Table 5.5 Specification of training/test set for taxi status inference. 78
Table 5.6 Evaluations of taxi status inference of occupied taxi GPS. 80
Table 5.7 Confusion matrix for CRF_L1. 81

LIST OF FIGURES

Figure 1.1 OpenStreetMap database statistics of registered users (blue curve) and user gpx uploads (track points, pink curve). 1

Figure 2.1 Trajectory of a moving object (an ant), representative of its movement path over time (Dodge, 2011). 6

Figure 2.2 Road network around TUM main campus: (left) mapping of roads around TUM campus (source: Google Maps), (right) plain visualization of geometries of roads (road network data from OSM). 8

Figure 2.3 An overview of steps that compose the KDD process (Fayyad et al., 1996). 10

Figure 2.4 Failure cases of finding nearest roads for map matching in crossover, spur/spaghetti intersections, parallel roads, bypass (Krumm et al., 2007). Black thin lines indicate trajectory of GPS observations, white thick roads are correct matching results obtained from Krumm's method, and grey thick roads are results of finding nearest roads 18

Figure 3.1 Distance between location observations and the nearest roads (Y. Wang et al., 2011). A comparison of estimated positioning accuracies of taxis GPS data in three cities, Beijing, Shanghai and Guangzhou, in China. (a) Log-scale histogram of the count of distinct location observations according to their distance to the nearest road. (b) The ratio between the difference to the nearest and the second nearest roads within 100 meters 28

Figure 3.2. Trajectory data of the same movement in road network with varying sampling intervals. 30

Figure 3.3 Temporal dynamics of the mobility status in terms of speed (top), turning angle (middle) of a sample series of the taxi #10058 and its status (bottom). The sample series starts at 12:25 and ends at 15:19 in 2010-4-1. 31

Figure 3.4. Different perspectives on probabilistic graphical models (Koller & Friedman, 2009): (a) medical diagnosis using a Bayesian network to infer the causal relationships among diseases (Flu, Hayfever) and symptoms (Muscle-Pain, Congestion), (b) a sample Markov network. 33

Figure 3.5. Different graphical structure CRF. (a) chain, (b) grid and (c) tree. Note that only the graphical structures for output variables are depicted for grid and tree. 36

Figure 4.1 An example of map matching of spatial trajectory (red dash line) in road network (grey lines). The ground truth driving route is marked in a green line. 39

Figure 4.2. A chain-structured CRF for 3 GPS observations. The map on top illustrates a simplified situation of identifying roads and paths given GPS observations in the road network. This requires 5 random variables $y_1 \in \{r_1, r_2\}$, $y_2 \in \{p_1, p_2, p_3\}$, $y_3 \in \{r_3, r_4\}$, $y_4 \in \{p_4, p_5\}$, $y_5 \in \{r_5, r_6\}$ to build the CRF. Thus, nodes y_1, y_3, y_5 linking with observations (shaded nodes) are point nodes while nodes y_2, y_4 are path nodes. 40

Figure 4.3 Mapping of a sequence of location observations (red dots) and their projections on the associated point states (blue dots). The 1st Node has only 1 point states while the 5th node has the most, 7 point states. 42

Figure 4.4 Mapping of path states (blue lines) of a sequence location observations (red dots). 35 alternative path states are found between the 5th node and the 7th node. 42

Figure 4.5. State generation workflow. 45

Figure 4.6. Example of point state (blue dot) redundancy of a GPS point (red dot). 4 redundant states are marked with red circles. 46

Figure 4.7. State transition graph of a 4 node sequence. Redundancy elimination yields to retain fully connected states (green nodes) while remove partially connected states (red nodes)........ 47
Figure 4.8. An example of missing values. Because of the missing road (red link in the red circle), the actual path (indicated by the green arrow) chosen by car cannot be covered by all the path states (blue lines) between 3rd node and 5th node........ 56
Figure 4.9 Mapping of service trajectories of a taxi in one day in Shanghai, China. Occupied trajectories is illustrated in green while nonoccupied trajectories are illustrated in red. The trajectories are also marked with serial numbers which indicate the taxis' temporal activities throughout the day. 59
Figure 5.1 Shanghai OSM road network at scale 1:203,669. 66
Figure 5.2 Web-based interactive routing tool, OSRM, for labelling GPS trajectories for map matching........ 67
Figure 5.3 Entity Relationship Diagram of the database design for feature extraction........ 69
Figure 5.4 The spatial distribution of GPS trajectories in the test data for map matching (Top). The statistics of sample trajectories (Bottom): travel distance (upper left), trip duration (upper right), observation count (bottom left) and daytime period in hour (bottom right). ... 71
Figure 5.5 Distribution of the sample feature data. In each block, (up) is raw feature, (middle) is rescaled feature, (bottom) is standardized feature........ 72
Figure 5.6 5-fold cross-validation estimates for optimal hyper-parameter λ........ 74
Figure 5.7 Learned weights in ℓ_1 regularized CRF for map matching........ 75
Figure 5.8 Map matching results of CRF_L1 (red) overlaid by ground truth (green) on the road networks (grey). 76
Figure 5.9 Error instance: parallel road. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green. 76
Figure 5.10 Error instance: U-turn. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green........ 77
Figure 5.11 Error instance: Starting/ending points. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green. 77
Figure 5.12 50 taxis' one-day trips in Shanghai, China. Blue lines represent occupied taxi trips and red lines are vacant. 78
Figure 5.13 5-fold cross-validation estimates for optimal hyper-parameter λ. Horizontal green dash lines mark the highest performance, and the vertical green dash lines indicate the power used to compute the hyper-parameter. 80
Figure 5.14 Learned weights in ℓ_1 regularized CRF for taxi status inference. 81

CHAPTER 1.

Introduction

1.1. Motivation

With the development of positioning and communication technology, collecting and sharing position data has become an everyday routine in modern cities for many social sectors. Moreover, this practice has reached a wide range of levels of the social structure, which leads to a wide coverage in both spatial and temporal scale of urban mobility on various granularities. Thousands of taxis report their position logs via telecommunication to the taxi dispatch centers for fleet management, millions of mobile users' locations are collected anonymously to provide contextual information in the location based service (LBS), and GPS-enabled wearable devices (e.g., Garmin Fenix) also boost huge interests of recording running tracks for sharing and analysis for personalized fitness plan. These temporally ordered location data, which capture all sorts of the urban mobility in the *road networks* and share a common data scheme, are often referred to as *spatial trajectories*.

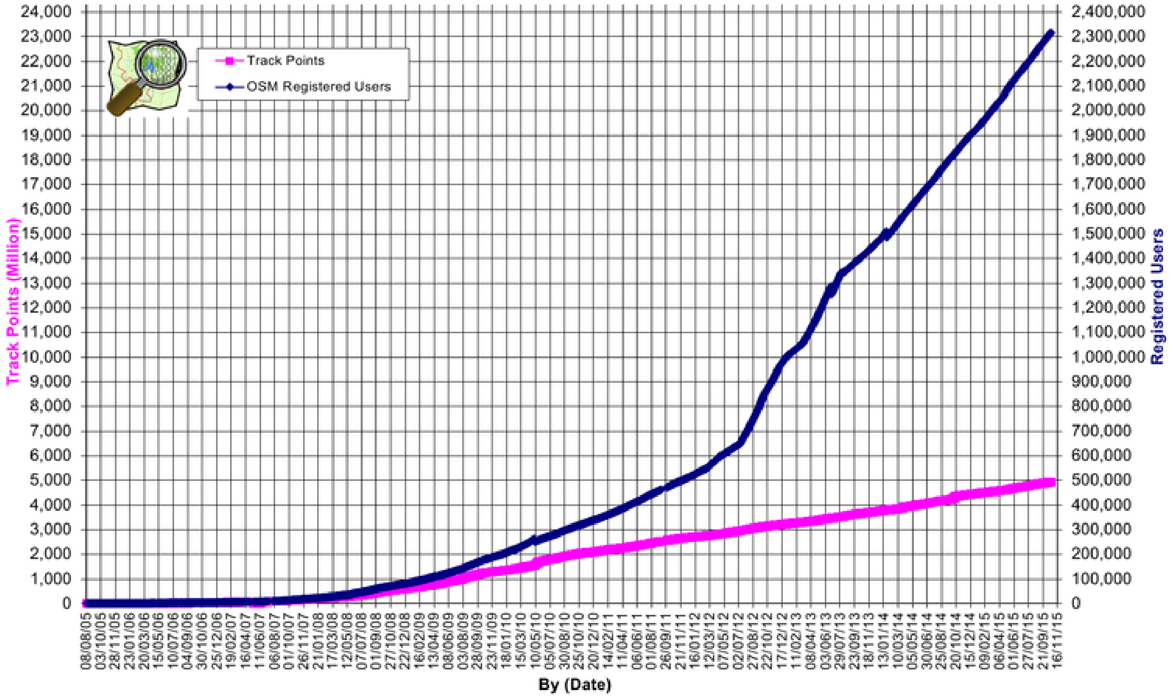


Figure 1.1 OpenStreetMap database statistics of registered users (blue curve) and user gpx uploads (track points, pink curve).

Besides the unprecedented scale of the data being captured, the way that these trajectory data are distributed and accessed have also been revolutionized. Rather than being kept alone as a private asset, more and more trajectory data are shared via the internet for free.

By the time of writing, OpenStreetMap, a crowdsourced world mapping project, has received around 5 billion track point uploads from its worldwide volunteer users, and nearly 50% of the data are uploaded in recent three years (see Figure 1.1[1]). Furthermore, the open data initiatives have been promoted in some modern cities in the form of legislation. A cartographer has managed to FOIL (the act of acquiring the data using FOIL - The Freedom of Information Law) one-year 50GB taxi trip and fare data from the local Taxi and Limousine Commission in New York for free[2]. And now the initiative of urban open data has endorsements also beyond US cities.

Due to movement's essential role to scientific and social studies, the availability of the massive trajectory data has triggered many efforts from the industry. For instance, Google is reportedly to have collected anonymized GPS data from authorized android handsets to provide a so-called time-in-traffic feature, which allows users to check current traffic conditions and estimate the travel time of their routing plans[3]. Turning to these constantly refreshed data has largely shortened the update cycles of dynamic traffic status. Moreover, as mobile apps becomes increasingly contextual, knowing what user's current activity – still, walking, cycling and in-vehicle can help to determine the right content to display. The mobile operating system (OS) android now has a built-in application programming interface (API) to extract this contextual information by classifying sensor data streams on the mobile devices. These data-driven endeavors have proved the trajectory data to be an invaluable resource to obtain better knowledge of urban mobility and lay grounds for many real world applications.

Unfortunately, a wide range of positioning sensors are used in the data collection process, which has no guaranteed individual positioning accuracy and no rigid data collection protocol is followed to ensure the consistency. Thus, trajectory data often suffer from poor data quality, e.g., imprecise positional data, missing data attributes, erroneous attribute values. Moreover, the trajectory data are often captured for positioning purpose only and seldom record semantic information intentionally. A lot of work has been done to enhance the data quality or enrich the data semantics so that to ease the utilization. For instance, map matching of trajectory data onto road networks (Lou et al., 2009a; Newson & Krumm, 2009; Yang & Meng, 2015), infer transportation modes (Yuan, Zheng, & Xie, 2012; L. Zhang, 2014), activity recognition (Lin Liao, Patterson, Fox, & Kautz, 2004). Note that all these tasks share the common output format, that is, a sequence of discrete label assignments.

These tasks of *labeling spatial trajectories* all have to resolve the uncertainty in the data, which roots either in the imprecise measurement or the ambiguous correspondences between observed data and the inquest labels. Many proposals are suggested to address these issues for individual tasks, which in general either adopts probabilistic modeling philosophy (i.e., to model the probabilistic distribution of the given data and the inquest labels) or favors a practical solution that yields the best outcome. The latter solution, though achieve

---

[1] Users and GPX uploads. http://wiki.openstreetmap.org/wiki/Stats#Accumulated_users_and_GPX_uploads.

[2] FOILing NYC's Taxi Trip Data. http://chriswhong.com/open-data/foil_nyc_taxi/.

[3] Google Maps gets real-time traffic, crowdsources Android GPS data. http://www.techspot.com/news/48015-google-maps-gets-real-time-traffic-crowdsources-android-gps-data.html

better result sometimes, is not capable of providing holistic representation of the trajectory data and thus couldn't facilitate the comprehension of the data. As for probabilistic methods for labeling spatial trajectories, the performance of the model mostly relies on finding the relevant variables (or *features*), which are mostly handcrafted. In this thesis, the tasks of labeling spatial trajectory in road network are investigated from a unified perspective, and we attempt to build a comprehensive model using probabilistic graphical model.

1.2. Goal

Driven by the ever fast growing need of making sense of open access trajectory data collected in urban context, labeling spatial trajectories in road networks serves the purpose of facilitating the general utilization of these noisy, ill-structured, semantically poor raw trajectory data. In particular, it intends to improve the data quality and enrich the data semantic upon which high quality LBS and various other applications can be built. Moreover, this thesis leverages machine learning techniques to approach these goals. It involves following research tasks.

- **Comparative study of the tasks of labeling spatial trajectory**. Labeling spatial trajectory refers to many trajectory-related tasks that share the formalism of sequential labeling. To materialize the concept, further discussion should be raised so as to identify the commons and uniqueness among the corresponding tasks. In order to motivate the model development, specific labeling tasks should be selected according to suggested categorization.

- **Model development using probabilistic graphical model**. The model development proceeds to resolve the task-dependent uncertainty using probabilistic graphical model (PGM). PGM is a modeling language that leverages the merits from both graph theory and probability and is suitable for capturing the structural dependencies in the trajectory data. With the designed graphical structure, features are induced to approximate the probability mass, given the label assignments.

- **Feature selection in structure prediction.** Feature selection is to find the most relevant feature set for the predictive models, which helps to reduce the model complexity and avoid overfitting of the data. In particular, the requirement for the application of feature selection technique to labeling spatial trajectory needs to be discussed.

- **Implementation and evaluation of proposed model.** In order to test the feasibility and performance of the proposed predictive model, implementation is done, which consists of three tasks, namely preparation of test dataset and label data, feature extraction from geospatial vector data and modeling training and testing. Furthermore, the effects of preprocessing on final performance also need to be discussed.

This thesis covers the entire pipeline of developing a predictive model for labeling spatial trajectories, and elaborates both theoretical concerns and engineering aspects of the data driven practice on the real world dataset.

1.3. Thesis Structure

Having discussed the motivations and clarified the goals, the author organizes the rest of the thesis as follows.

Chapter 2 introduces the data and the problem of labeling spatial trajectories. The data, including both spatial trajectories and road network, are described briefly in terms of concepts, data models and various data sources. As for the labeling tasks, first a general scope of knowledge discovery is given to identify the overlaps and gaps between our focus and the neighboring topics, then the labeling spatial trajectories is categorized into two types of tasks, namely localization and behavioral classification. The classification is further ramified in the characteristics.

Chapter 3 proceeds to the discussion of challenges in labeling spatial trajectories and introduces the theoretical basis of our work, PGM. The challenges arise from both the noisy and sparse data and the tasks themselves, that is, the distinguishability of the labels inquest. The theoretical fundamentals of the model development are explained by narrowing the candidate modeling tools that fit the target problem most. In addition, standard issues on designing graphical structure, feature induction and selection, inference and learning are discussed.

Chapter 4 discusses the model development for the two study tasks using undirected graphical model (UGM). First, a chain structured graphical structure is developed for labeling spatial trajectory and specific refinement is made for the task of localization, which needs to review a large number of labels. Then, two feature sets are developed for the two tasks accordingly. Furthermore, the choices of inference and learning algorithms are made for training and feature selection using ℓ_1 regularization.

Chapter 5 elaborates the practical issues in the implementation and evaluation using two test datasets. Three tasks are involved in the implementation: label data preparation for localization tasks, feature extraction in spatial database, and model development for training and testing. Both test datasets are derived from Shanghai floating car data (FCD) using specific preprocessing procedures for individual tasks. In the end, experiment results and case study are given to demonstrate the feasibility of the proposed models.

Chapter 6 concludes the major findings of the thesis and envisions the further development that can be built on our work.

CHAPTER 2.

Fundamentals and Related Works

This chapter aims to introduce the basic concepts, identify the study domain and discuss the specific tasks that have motivated the author. First, two research subjects, spatial trajectories and road networks, are conceptually discussed. Secondly, the research domain that labeling spatial trajectory resides in is sketched. Finally, two study tasks, map matching and taxi status inference are discussed by reviewing the state-of-the-art.

2.1. Spatial Trajectories in Road Network

2.1.1. Moving Objects and Spatial Trajectories

The study of movement has always been an key issue in many areas of scientific investigation or social analysis, which involves a broad range of moving objects, such as human, animal and vehicles (F. Giannotti & Pedreschi, 2008). Cartographers leverage the GPS traces of vehicles to update and refine outdated road network in terms of more accurate geometry or updated semantic information. Ecologists analyze patterns in animals' traces collected in the field or from tracking devices for animal behavioral study. Traffic engineers explore the city-wide taxi GPS traces in order to understand urban mobility and develop more realistic traffic models. Urban planners investigate the activities revealed in the movement and thus to evaluate the regional functions. That is, a variety of application domains enjoy the insightful outcome from the study of movement across geographic space (Gudmundsson, Laube, & Loon, 2012).

A *Moving object* can be referred to as a point (object) that changes its location over a certain period of time. The term is derived using the modeling language adopted in geoscience that treats the objects as point, line or polygon. The resulting paths in the space they move can be represented as time-referenced location sequences, namely *spatial trajectories*. For practical reasons, the movement can only be observed or recorded at finite moments and thus making spatial trajectories contain only a finite set of location observations. See Figure 2.1 for an illustration of moving object, spatial trajectories.

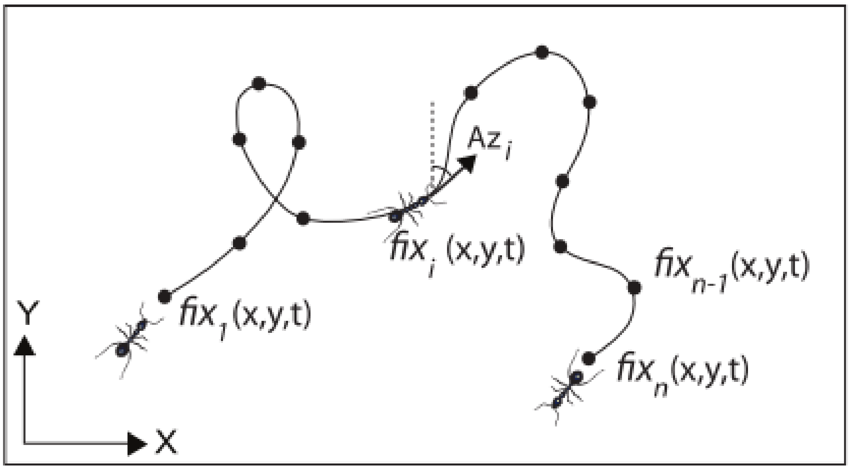


Figure 2.1 Trajectory of a moving object (an ant), representative of its movement path over time (Dodge, 2011).

To be concrete, we give a few examples to demonstrate the aforementioned concepts and motivate some applications.

- Taxi floating car data (FCD) collected by taxi companies for fleet management and dispatch system. With the GPS-enabled devices installed in the taxis (often linked to the meters), taxis' coordinates as well as their speed, direction, occupation status, can be collected in a specified time interval.
- Besides GPS, position logs can be obtained from mobile phones which mark users' locations referenced to the cells in the telecommunication network. These mobility data streams with recording users entering a cell – *(userID, time, cellID, in)* – users exiting a cell – *(userID, time, cellID, out)* (Fosca Giannotti & Pedreschi, 2008). Note that the users' locations are not explicitly given but need to be estimated by referencing to the locations of the cell towers.
- Zebra fish's movement data derived from video sequences using video tracking software at 30 frames per second (Soleymani, Cachat, & Robinson, 2014). The data are used for study on fish's behavior under different dosing conditions.

As can be seen form these examples, spatial trajectory provides a concept model to facilitate general study on the data of such kind and, in general, comprise three components such as space, time, and moving objects (Andrienko, Andrienko, Bak, Keim, & Wrobel, 2013). Space refers to a set of places or locations, in which location can be referenced in various manners, e.g., geographic coordination. Time, often seen as indexes of the locations, can be simply a universal time or relative time moments, e.g., elapsed time, abstract time stamps. Moving objects are in most cases only reflected using a unique identifier, but it is a fundamental component for the individual pattern discovery and collective behavior discovery.

With the advent of more reliable and low-cost object-tracking technologies, trajectory data can be collected at an unprecedented scale at a routine basis and thus have stimulated diverse and fast growing research to model, manage, analysis these data. Although trajectory data confirm the concept model of spatial trajectory, they do not necessarily obey a common data model or data format. In other GIS literatures, there exists some other terms bearing the same meaning of what we use in this thesis. Table 2.1 attempts to clarify the correspondences among different terms.

Table 2.1 Clarification of spatial trajectory related terminology.

This thesis	Equivalent terminology
moving object	moving entity, mobile object, dynamic object
spatial trajectory	trajectory, mobile trajectory, GPS trajectories, GPS trace
raw data	FCD, mobility data, movement data

This work focuses on movements that reveal mobility in urban environment. Therefore, we discuss the predominant means of transportation that facilitates and shapes the movements of such kind.

2.1.2. Urban Road Network

Urban transportation network refers to the infrastructures that facilitate urban mobility of human, vehicle, goods, etc. In a modern city, it’s often composed of a variety of mono-modal components such as motorized road network for private car driving, pedestrian way network for walking, and public transit networks including underground, suburban and tram lines for passenger transporting (L. Liu, 2011). The transportation networks considered in this work are mostly the ones that use carrier type of road, namely *urban road network*. However, the discussion can be generalized to other carrier types that share similar network models for the mobility application.

The study of road network is of widespread interest in the GIS community. In a GIS system, road network is modeled as points (e.g., road intersections) and lines (e.g., roads) while retaining their topological relationships, geographic positions and shapes. Besides the connectivity among roads, each road is characterized by *attributes* such as road classification (e.g., national highway, provincial highway, and county highway in China), traffic regulation (e.g., minimum speed limit, maximum speed limit, prohibit of U-turn), Point of Interests (POIs, e.g., school, restaurant, hotel) and so on (Gong, 2011). And for routing purposes, a graph representation needs to be constructed from a road network. Figure 2.2 shows the map representation of a sample road network and its underlying road network data model.

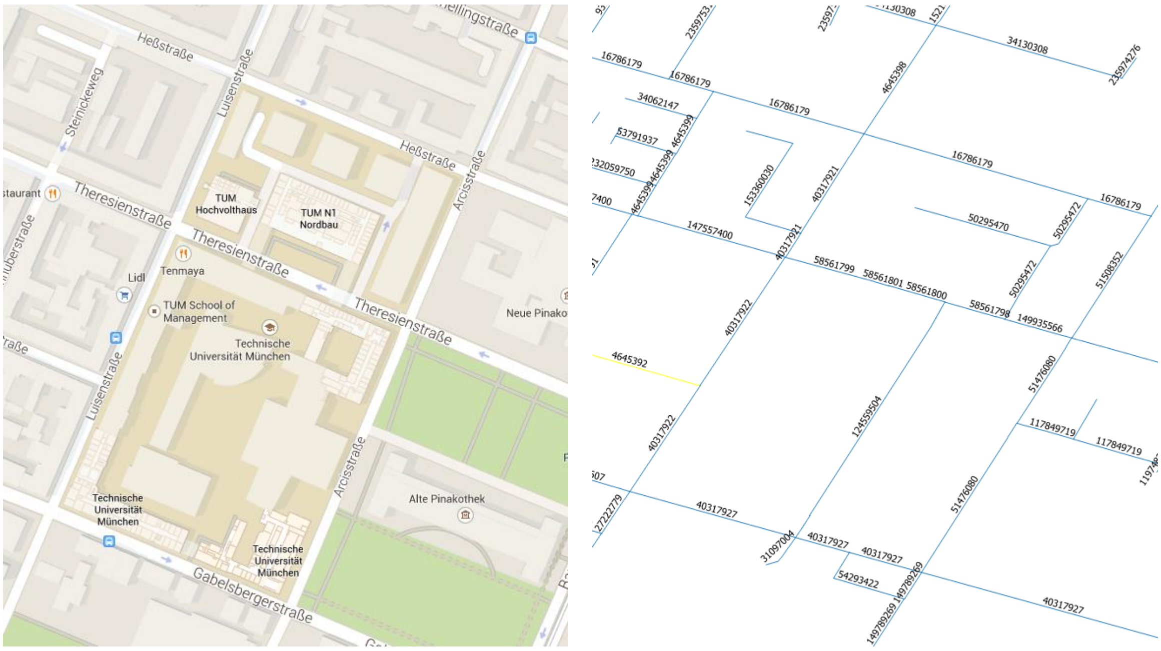


Figure 2.2 Road network around TUM main campus: (left) mapping of roads around TUM campus (source: Google Maps[4]), (right) plain visualization of geometries of roads (road network data from OSM).

There exists a number of road network datasets collected from either public or private organizations. Most of these datasets are tailored for specific applications (e.g., navigation, traffic engineering) and thus differ in geometry, accuracy, actuality and resolution for the roads in the same geographic area. Therefore, there are dedicated research to develop automatic methods of road network conflation in order to provide an integrated data service. (M. Zhang, 2009) performed extensive evaluations on four road network datasets including ATKIS[5], Tele Atlas, NAVTEQ and OpenStreeMap[6] (OSM), which we use here as an example to rationalize the choice of road network dataset for the study on spatial trajectory.

- ATKIS (Amtliches Topographisch-Kartographisches Informationssystem, the official topographic information system in Germany), produced via map digitization and object extraction from remote sensing imagery, is a general topographic dataset that serves as an information basis on top of which application-dependent data can be added (Volz, 2006). It contains a road layer which is composed of geometries and general-purposed attributes of road centerline with an accuracy of ±3m.
- Tele Atlas, acquired by TOMTOM[7], is data vendor that provides a fully attributed geospatial dataset for navigation, location-based services (LBS), and general mobile and internet mapping applications. The Tele Atlas road network data is acquired through both map digitization, field measurement. As one of the leading data vendors in the market, it provides an accuracy that is less than 10m in built-up area while 25m outside built-up area in Europe.

---

[4] https://www.google.de/maps/@48.1494453,11.5688217,17z?hl=en

[5] http://www.adv-online.de/Geotopography/ATKIS/

[6] https://www.openstreetmap.org/

[7] http://www.tomtom.com/

- NAVTEQ, now merged to HERE[8], is a leading data vendor that provides digital navigable maps on a global basis. Similar to Tele Altas, NAVTEQ, road network data contains both geometries and rich navigation-related attributes which are captured through map digitization and field measurement.
- OSM is a free, editable map of the whole world built by volunteers (“OSM wiki,” 2015). The OSM road network data is created via GPS-enable field measurement using a variety of consumer mobile devices and road digitization from satellite imagery. As OSM adopts a free structure for data acquisition, the road network data contains the geometries and an arbitrary number of attributes.

The selection of road network data for study on spatial trajectory data largely depends on the availability of the data, quality of the data (e.g., coverage, positional accuracy, richness of routing relevant attributes) and so forth. Though being criticized for its heterogeneous quality, OSM has enjoyed a steady growth throughout the years and has been adopted for many governmental and commercial usage. Furthermore, OSM road network data has a relatively good quality in the mega cities (Y. Wang, Zhu, He, Yue, & Li, 2011), which make it a conceivably choice for this work.

2.2. Labeling Spatial Trajectory

Due to the essential role of movement in nature and social system, a wide range of research efforts have been made to carried out on spatial trajectory. And thus making spatial trajectory related research a multidisciplinary/interdisciplinary field that can be applied to movement ecology, behavioral studies, transportation, and so forth (Dodge, 2011). Among these application domains, it often requires to *label* individual point in the trajectories with statuses in query such that the physical measurement can be better interpreted to understand the target movement, and we call these tasks *labeling spatial trajectory*.

For example, map matching needs to assign each data point in the location sequence to the road that moving object traveled on, location-based activity recognition identifies the activities (e.g., at home, at work, at bar) occurred at each location in the trajectory data, and transportation mode detection reveal the transportation modes (e.g., walk, cycling, driving, bus, subway) being used at each data point. In these tasks, each location-based observation is assigned to task-specific labels, i.e., roads in map matching, activity types in activity recognition and transportation modes in transportation mode detection.

Labeling spatial trajectory serves a number of useful purposes in the context of cartography and geographic information science (GIS): 1) Data semantic enrichment. Transportation mode detection enables high-level query such as “How many transportation modes do passenger use during a day?” 2) Data quality enhancement. Map matching calibrates the sparse and noisy location observations on to road network which leads to more accurate location data.

In the following sections, a bigger scope of spatial trajectory related research with a focus in computation is introduced which helps to identify the interrelationships between labeling

---

[8] https://www.here.com/.

spatial trajectory and its neighboring topics. Then a detailed discussion of labeling spatial trajectory is given.

2.2.1. Knowledge Discovery in Trajectory Data: A Retrospective Overview

The diverse and fast growing research on trajectory data have led to a wide range of publications in the past decades. In general, labeling spatial trajectory falls in the category of trajectory data mining and knowledge discovery for its close relationship with classification. In order to sketch the scope of this focal research topic, we follow the line of works in Geographic Knowledge Discovery (GKD) and spot on several major milestones in the development of movement research in GIS community in recent years, including international research collaborations, research seminars and book publications.

GKD is a special case of Knowledge Discovery in Database (KDD) which deals with geospatial data. KDD is an interactive and iterative process that is designated to identify valid, novel, useful and understandable pattern in data (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). As illustrated in Figure 2.3, a typical KDD process includes several steps including investigation of the application domain, selection of a target dataset, data cleaning and preprocessing, data transformation, application of data mining methods, interpretation of the discovered pattern, in which data mining methods serve as a critical step to identify the patterns in the data. Note that patterns in general refer to various forms, e.g. a particular rule in a classifier or a linear component in a regression model. Following Fayyad's definitions, Miller and Han shed lights on GKD and argued that it is a nontrivial special case which requires systematic investigation due to high while interrelated dimensionality, spatial dependency and heterogeneity, complexity in spatial objects and diverse data types of the geospatial data (Miller & Han, 2001). In their book collection, Smyth contributed his early vision of the opportunities of applying data mining techniques to mobile trajectories based on the framework of “stored behavior - predicted behavior” (Smyth, 2001).

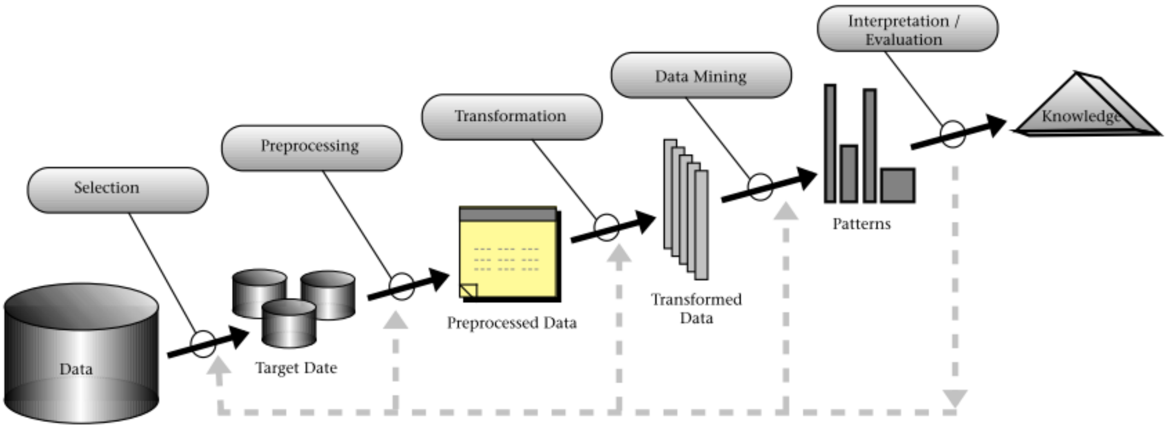


Figure 2.3 An overview of steps that compose the KDD process (Fayyad et al., 1996).

Fostered in the project GeoPKDD – *Geographic Privacy-Aware Knowledge Discovery and Delivery*[9], the book (Fosca Giannotti & Pedreschi, 2008) put forwards the research in trajectory data mining with a special focus on privacy issues. Being motivated by the practice of mining mobile phone log data, they proposed a three-step process for knowledge discovery in mobility data, namely trajectory reconstruction, knowledge extraction and delivery of obtained information. Trajectory reconstruction is to reconstruct trajectories for individual moving objects from the raw mobility data such as mobile phone log data, taxi FCD, etc. Knowledge extraction refers to adapting data mining techniques to trajectories such as clustering, frequent pattern discovery and classification. As for knowledge delivery, interpretation of obtained information as well as designing appropriate representation and visualization to facility user's explorative reasoning are of major concern.

With the thrive of GPS-enabled mobile devices, the COST Action IC0903 MOVE[10] carried on the focus of knowledge discovery regarding massive moving objects and echoed the six-year series of Dagstuhl Seminars[11] under the theme of *Representation, Analysis and Visualization of Moving Objects* (Bitterlich, Sack, Sester, & Weibel, 2008; Gudmundsson et al., 2012; Sack, Speckmann, Loon, & Weibel, 2010). The series attracted researchers with various backgrounds and foster discussions on topics of data modeling, management of trajectory data, movement ecology, pattern discovery in movement data, visual analytics and so forth. Practical issues concerning benchmarking of movement data analysis also received consistent attention.

Another notable effort beyond the GIS community is the comprehensive work addressing the development of computational methods for spatial trajectories that facilitate the applications in transportation and social networking in the urban context (Y Zheng, 2015; Yu Zheng & Zhou, 2011). The book proposed a technical framework that illustrates the various tasks and applications regarding computing with spatial trajectories. The framework put topics into two category, namely foundation and advanced topics. Foundation includes the tasks of preprocessing, indexing and retrieving trajectory data prior to and within database. And advanced topics mainly focus on application-oriented tasks such as activity recognition, trajectory analysis for driving, privacy issues, trajectory pattern mining and location-based social networks. An updated framework devoted to trajectory data mining is given in (Y Zheng, 2015).

2.2.2. Labeling Spatial Trajectories

Bearing in mind the examples listed before, labeling spatial trajectories refers to the tasks that requires performing point-wise label assignment. The term *labeling*, in this thesis, refers to a more general categorization other than *annotation, tagging* which are more often addressed in the literatures (Parent & Spaccapietra, 2013). The distinction is that labeling spatial trajectories may serve multiple purposes rather than solely semantic enrichment of trajectory data (e.g., trajectory annotation). This categorization provides a new perceptive to

---

[9] http://www.geopkdd.eu

[10] http://www.move-cost.info/

[11] https://www.dagstuhl.de/en/program/dagstuhl-seminars/

study the common characteristics of labeling tasks other than being restricted in ad hoc solutions.

Labeling tasks can be classified into two types based on the specific label set in the tasks, namely localization and behavioral classification.

- ***Localization*** refers to the task of inferring the actual positional information of the points in the trajectory. In this task, labels are often modeled as a set of candidate positions where the moving objects were observed. For example, labels in map matching are a set of candidate roads that were traveled on (note that the observation may deviate from the roads due to noise in the positioning process).
- ***Behavioral classification*** shares the same purpose with trajectory annotation and is designed to infer the states/labels of the moving objects, which often vary across specific applications. For example, labels in transportation mode detection are transportation modes that need to be classified.

There exist several differences between these two types of labeling tasks. Firstly, labels in behavioral classification bear clearer semantics than those in the localization tasks. Secondly, localization tasks often have larger label spaces, which may introduce huge computational cost in finding the most likely assignments. Therefore, these distinctions require non-trivial treatments in specific solutions.

Furthermore, multiple policies can be used in labeling spatial trajectory regarding the granularity ranging from point-based, segment-based to trajectory-based. Point-based policy means the designed models or algorithms assign labels to individual points. Segment-based policy needs to perform segmentation first which groups consecutive points in the trajectories into meaningful segments and assume that they share the same labels. Trajectory-based policy means that all points in the trajectory share the same label. These policies differ in the temporal scale of observations to be considered, and thus reveals varying degrees of flexibilities in the label assignments. Point-based method encourages straightforward point-by-point processing workflow while often suffers from being inflexible in evaluating the pattern embedded in the varying number of neighboring points. Segment-based policy may embody the risk of being too “aggressive” in the step of segmentation.

To be concrete, we compare several labeling tasks in terms of size and semantic of label set, applied methodologies and the input datasets in Table 2.2 below.

- As it can be seen in the table, the tasks of localization have much larger label sets than the tasks of behavioral classification. And the sizes could increase dramatically depending on the spatial extension of the moving space, e.g., indoor localization (Krumm & Horvitz, 2004) considers much less candidate locations than city-wide localization in road network (Hunter & Herring, 2009). Furthermore, the distinguishability among the labels also differentiate the two tasks. In localization, labels are considered as specific locations. The *distance* between two labels often reveal the mobility of the moving objects in the study space and thus the *label transition* can be represented using the traveling cost (e.g., all the tasks in localization model the label transition as a cost function of routing in the reference graph such as skeleton graph for indoor walking space, road network.). However, labels for behavioral classification may reveal varying degree of difficulties in label assignments. For examples, stop versus moving are easier to be classified since only two *features*[12] are needed (Krumm & Horvitz, 2004). And for multiple labels, a few labels are harder to classify than others (Shamoun-Baranes et al., 2012).

- A variety of methods were applied in the example tasks, ranging from tailored search algorithms to probabilistic models with well-established theoretical grounds. Most popular methods in the list fall into the class of *graphical model*, which correspond to the *dependency analysis* in the conventional data mining methods (Fayyad et al., 1996). These methods, in general, investigate two probabilistic modeling efforts, namely the label-observation dependencies and label-label dependencies. Label-wise dependencies include dependency between neighboring labels, e.g., user's motion status changes from stop to moving (Krumm & Horvitz, 2004) or dependencies among multiple labels in arbitrary positions in the trajectory, e.g., (Lin Liao, Fox, & Kautz, 2005b) introduced a soft constraint implemented as summation aggregation among the label assignment for activity recognition. Rather than explicitly modeling labels' temporal dependencies in the model, segmentation-based methods explore the dependencies by employing a segmentation procedure first, e.g., (Yu Zheng, Liu, Wang, & Xie, 2008) first segmented consecutive GPS observations into groups and then performed segment-based classification to identify different transportation modes. The third class of methods tailors a greedy search algorithm to find the optimal segments under predefined spatial-temporal criteria (Buchin, Kruckenberg, & Kölzsch, 2012).
- Trajectory data also exhibit a number of distinctions in the study examples. Firstly, these data are collected via various positioning techniques with different positioning limitations in the moving environment (e.g., WiFi signal readings are used in the indoor localization), which requires different preprocessing and features for labeling tasks accordingly. Secondly, the sizes of the dataset in terms of number of location samples pose a significant difference among the tasks. And efforts have been rarely made on the determination of an appropriate size of the dataset so that it's sufficiently large for modeling learning and validation. Thirdly, a varying number of moving objects are being evaluated ranging from only one moving object to hundreds of them. And there's also a task that studies the learnability of common behaviors among different users (Lin Liao et al., 2005b). Finally, all localization tasks use an auxiliary dataset which serves as reference for positioning and help to generate the label set. However, the auxiliary dataset can be considered as an optional side information in behavioral classification which makes no compromise in model's performance (Yu Zheng et al., 2008).

---

[12] Note that, in this thesis, the term feature is used equivalent to variable, predicator which follows the usage in the machine learning literatures. Reader shouldn't confuse the meaning that is specified in the Open GIS consortium as an abstract entity.

Table 2.2 Comparison of example labelling tasks.

Labeling Task	#Label	Labels	Methodology			Trajectory data	Auxiliary data
			model/algorithm	seg[13]	#feature		
Localization							
Indoor location inference (Krumm & Horvitz, 2004)	317	locations	HMM	NO	4	4586 WiFi signal readings from 10 walks with sampling rate of 36Hz	corridor graph
Path Inference (Hunter, Abbeel, & Bayen, 2013)	$\lvert\mathcal{R}\rvert$[14]	roads	CRF	NO	10	Dataset 1: Supervised learning using 700,000 samples of 10 taxi in 2 days with sampling interval of 1sec	road network
						Dataset 2: Unsupervised learning using 600,000 samples of 600 taxis in 1 day with sampling interval of 60sec	
Map matching (Newson & Krumm, 2009)	$\lvert\mathcal{R}\rvert$	roads	HMM	NO	2	4605 samples of 1 car with sampling rate of 1Hz	road network
Behavioral classification							

---

[13] seg is abbreviated for segmentation.

[14] $|\mathcal{R}|$ refers to the size of road network $\mathcal{R}$ which is often very large.

Pedestrian motion inference (Krumm & Horvitz, 2004)	2	stop, moving	HMM	NO	2	8870 WiFi signal readings with sampling rate of 36Hz	–
Mobility detection (Sohn et al., 2006)	3	stop, walking, driving	Boosted logistic regression	YES	7	GSM traces of 3 mobile phone users in 78 days with sampling rate of 1Hz	–
Geese trajectory segmentation (Buchin et al., 2012)	2	flight, stop	Greedy search	NO	4	location samples of mitigating geese in 4 months with maximal sampling interval of 2hr	–
Oystercatcher behavior classification (Shamoun-Baranes et al., 2012)	3/8[15]	fly, forage, body care, sit, stand, handle, walk, aggression.	Decision tree	NO	17	16434 GPS observations and associated 972406 accelerometer observations (1 GPS associated with ca. 60 accelerometer measurement) of 3 oystercatchers. 702 GPS observations are labeled in the field work.	GADM[16]
Taxi status classification (Zhu et al., 2011)	3	parking, occupied, nonoccupied	Decision tree + HSMM	YES	26	25million GPS samples of 600 taxis with sampling interval of 1 min	road network, POI
Transportation mode detection (Yu Zheng et al., 2008)	4	walk, car, bus, bike	Decision tree	YES	6	45 users' GPS logs covering 20,000km	–
Activity recognition (Lin Liao, Fox, & Kautz, 2005a)	7	work, sleep, leisure, visit,	Relational Markov Network	NO	5	Dataset 1: single user GPS logs in 4 months (400 visits to 50 different places)	road network, POI

---

[15] In total 16 states are introduced, but they are aggregated to 3 and 8 states in the model development.

[16] Geographic database of global administrative areas. http://www.dadm.org

	pickup, on/off car, other	Dataset 2: five users GPS logs in one week (25-35 visits to 10-15 places)

2.3. State-of-the-art of Study Tasks

Having introduced labeling spatial trajectory, we now zoom in to specific tasks with the purpose to unveil the common insights. These tasks cover both types in labeling spatial trajectory, namely localization and behavioral classification. The trajectory data are collected using the same positioning techniques and reflecting the same type of movements. In addition, we are more interested in the domain transportation as it provides huge potential of applications which may benefit from our study. Therefore, we choose low sampling rate map matching of taxi GPS trajectories for localization and inferring taxi status for behavioral classification in this work.

In the remainder of this section, we discuss the two study tasks in terms of problem statement, challenging issues, and state-of-the-art.

2.3.1. Map Matching

Map matching was first raised to provide accurate location information in an in-vehicle route guidance system (Collier, 1990), in which location-based observations (i.e., position, direction, speed) are estimated from on-board sensors (e.g., wheel rotation counters) through dead reckoning. Driven by the demand on location-based services and fast growing interests in knowledge discovery in trajectory data, map matching has been an active research topic throughout the years. However, the scope of the research has moved beyond vehicle route guidance system and it now serves as a fundamental technique for a broad range of real-world applications and research tasks (Yu Zheng & Zhou, 2011), such as travel time estimation, fleet management, route choice study, etc. Though each of these applications could raise specialized requirements for the problem, the map matching tasks share the common goal of associating the position data with the road data. More specifically, the process identifies the road segment in the road network data for each position data and the position on the road where the position data is recorded.

The fundamental need of investigating map matching is caused by the fact that the location measurements are often noisy due to the inherent inaccuracy in the positioning sensors and complicated positioning environments (e.g., signal delay and blocking in urban canyon for GPS), and thus the location data often deviate from the road center line. Note that most of map matching methods assume that a high quality road network dataset is given, which may not be true all the time (Quddus, Ochieng, & Noland, 2007). A straightforward idea is to snap each observation point to the closest road. Unfortunately, finding the nearest road often fails in the complex urban road networks. Krumm et al. (2007) reported the failure of map matching relying on the nearest roads in a number of local road structures including cross over, spur/spaghetti intersection, parallel roads, bypass and so forth (Krumm, Letchner, & Horvitz, 2007) (See Figure 2.4). Even though nowadays positioning sensors could achieve relative high accuracy, there are still problems with map matching in the real world data (H Wei, Wang, Forman, & Zhu, 2013). Therefore, a specialized method is necessary to tackle these problems.

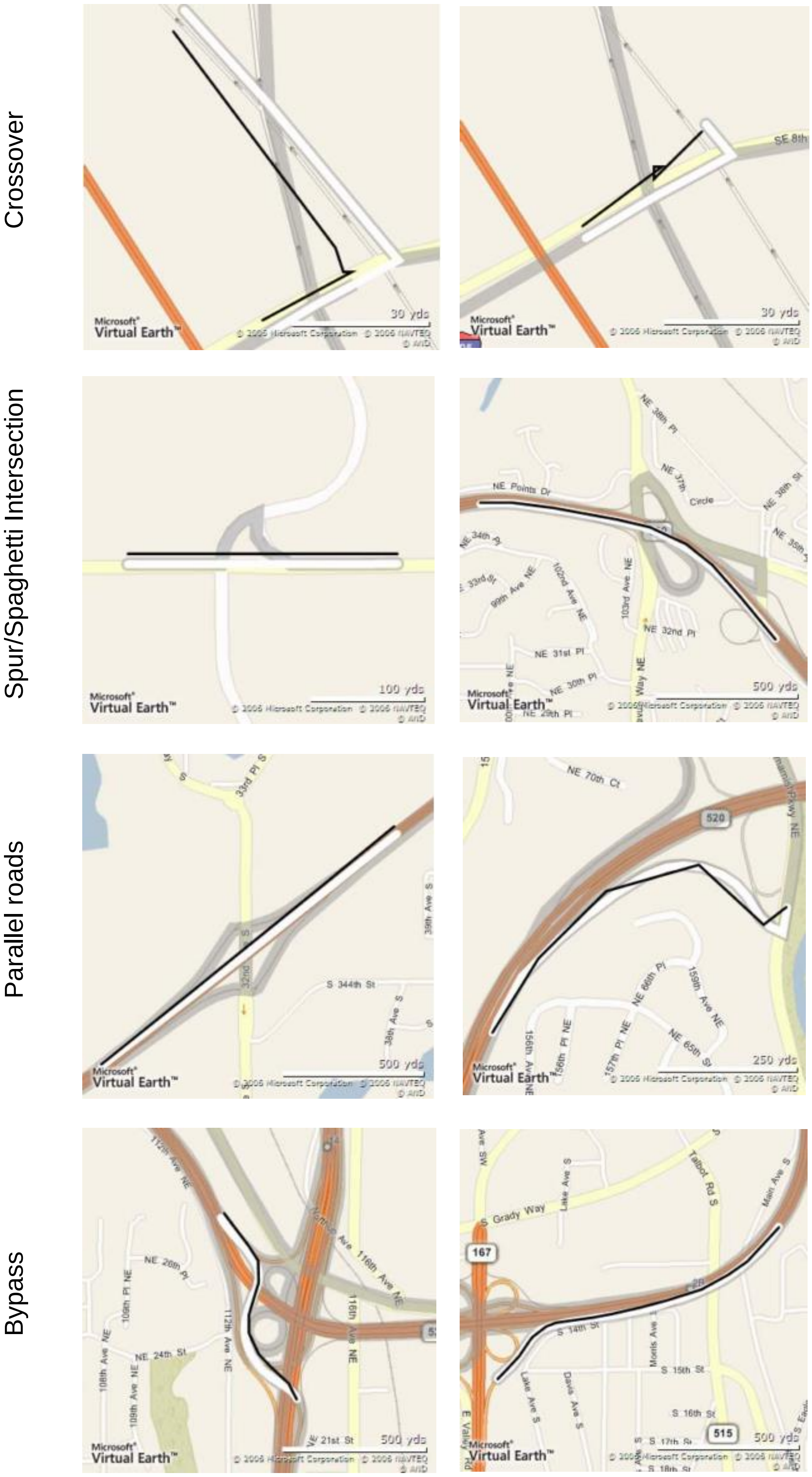


Figure 2.4 Failure cases of finding nearest roads for map matching in crossover, spur/spaghetti intersections, parallel roads, bypass (Krumm et al., 2007). Black thin lines indicate trajectory of GPS observations, white thick roads are correct matching results obtained from Krumm's method, and grey thick roads are results of finding nearest roads.

In addition to the noisy measurement, more challenges have been addressed in numerous literatures throughout the years. The major research focuses in recent years are summarized as follows.

- **Accuracy**. Different levels of accuracy are required in the applications. For ATT services (navigation and road guidance, distance-based pricing, etc.), high sampling rate (e.g. 1 Hz-30Hz) is used and demands horizontal position accuracy of 10m for each position estimation (Quddus et al., 2007). However, road-level accuracy, namely only the correspondence between position data and road segment is required in analyzing historical position data in road network, e.g., route choice analysis (Frejinger, 2008), traffic flow analysis (Giovannini, 2011), mobility pattern discovery.
- **Runtime Efficiency**. Efficiency becomes critical in real world application, which requires map matching either in real time or at a massive scale. A wide range of techniques has been used to tackle this issue with different phases and aspects of the map matching. For instance, use adaptive search range for GPS position errors to reduce the number of candidate road, build spatial index to facilitate the spatial query, simplify the road network to reduce search space(K. Liu, Li, He, Xu, & Ding, 2012), employ parallel computing to speed up the process at either algorithm level or program level. Many of the efforts are highlighted in the GIS CUP2012 (Ali, Krumm, Rautman, & Teredesai, 2012).
- **Robustness**. The position data may reveal varying degree of difficulties for map matching in terms of *nosiness* (inaccuracy of the position measure) and *sparseness* (sampling interval), which depends on positioning techniques, data collection protocols, etc. Hence, general map matching methods are required to maintain consistent performance of accuracy and efficiency when the data deteriorate. In particular, low-sampling rate GPS data in urban context attract most attentions in recent development of map matching.
- **Online/off-line.** Online/off-line refers to two different scenarios of map matching. Online task aims at generating matching results of current position data when only historical data are available. And off-line task offers fully observed position data for the trip. Therefore, online processing often accompanies the navigation task while off-line is considered for post processing such as traffic flow analysis. This leads to a concern of tradeoff between accuracy and latency in the development of map matching approaches. Proposed strategies include fixed/sliding windows, finding convergence point (Goh, Dauwels, & Mitrovic, 2012), and dynamically determine the output point with specified cost (G. Wang & Zimmermann, 2014).
- **Incomplete map data**. Map data, namely the road network data, is another crucial input for map matching. In addition to the positional inaccuracy, map data may also suffer incompleteness, i.e., missing minor road segments or newly constructed roads, incorrect driving directions. This is addressed by examining the matching results against a confidence threshold and thus identifying the portions of trajectory data for missing roads (Pereira, Costa, & Pereira, 2009; Torre, Pitchford, Brown, & Terveen, 2012).

- **Other variants**. There exist other efforts that variates from conventional map matching tasks, such as matching position data beyond road networks (Chen & Bierlaire, 2013), jointly tackle map matching and other tasks (e.g., behavior detection(Lin Liao, 2006), travel time estimation(Li, Ahmed, & Smola, 2015), modeling trajectory data uncertainty(K. Zheng, Zheng, Xie, & Zhou, 2012), joint matching and map building(Torre et al., 2012)).

The aforementioned focuses, either solely or jointly, have shaped the development of map matching methods. In particular, the level of accuracy has led to two directions of research. As for the point level accuracy, appeared in early years for navigation, methods that are capable of dealing continuous variables are considered, e.g., Kalman fitler, Particle filter; while for road level accuracy, methods for discrete variables are favored, e.g. HMM. Meanwhile, some focuses are addressed jointly. For instance, efficiency gain can be achieved either in the overall design of the method, i.e. global matching or incremental matching, or in a single phase of the overall process.

An extensive literature survey of map matching methods is carried out in (Quddus et al., 2007) which suggested four categories for its kind, namely *geometric*, *topological*, *probabilistic* and *advanced* methods. Geometric methods use geometric properties in terms of distance, direction to identify likely matching pair. Topological methods consider the connectivity between road segments. Probabilistic methods explicitly model the error regions of the position measures so as to assign probability/weight to individual candidate that intersects with the regions. Advanced methods rely on methods such as Kalman filter, particle filter, fuzzy logic, Hidden Markov Model (HMM), etc. Another categorization, *global matching* and *incremental methods*, is suggested in (Brakatsoulas & Pfoser, 2005). In global matching, the entire trajectory is used to determine the matching output while only a chunk of trajectory data is used in incremental methods.

For modern map matching research, more and more efforts are focused on the road-level accuracy on low sampling rate trajectory data. And from the methodology's perspective, most recent methods fall into the category of advanced methods. Unfortunately, there exists no commonly agreed state-of-the-art methods (Ali et al., 2012), probably due to the lack of benchmarking dataset, varying characteristic of the available datasets and diverse research motives. However, HMM based method and its variants are the most cited state-of-the-art in recent map matching literature, which often make use of both geometric and topological properties of the data while measure the affinity between position data and road network in terms of likelihood or weight. Therefore, a more focused review on HMM-based methods and statistical methods is provided in the following section.

To ease the representation, these methods are presented in three groups, namely HMM based methods, HMM variants, and empirical probabilistic models.

HMM based methods

HMM is a statistical model for segmenting and labeling sequence data. It models the joint probability of the observation sequence and state sequence. Given the initial probability of states, probability of individual observation conditioned on states (observation probability), probability of current state conditioned on previous state (transition probability), then the (hidden) state sequence can be solved with maximum likelihood. The solution is solved via

Viterbi algorithm. Then a basic HMM based map matching method is used to model the states and design the observation and transition probability.

(Newson & Krumm, 2009) model the states as road segments and use these states generate the GPS observations. The observation probability is set to the Gaussian distribution of the distance between observation and the nearby road, while the transition probability is computed as exponential distribution of the difference between the length of the shortest path and the distance between successive observations. The two parameters are estimated via statistical tests on samples draw from the test data.

(Goh et al., 2012) penalized the above observation probability with speeding factor and train a Support Vector Machine (SVM) with Radial Basis Funciton (RBF) for the transition probabiltiy to combine the distance discrepancy and the momentum change of the traveled path found via A* algorithm. The training uses 3000 path instances with binary label, that is being either actual path or not. To reduce the output latency, a varying sliding window (VSW) is designed for online traffic sensing.

(Raymond & Morimura, 2012) model the states with shape points in the road data rahter than the observation's projection on the road, which is claimed to have improve the suboptimality in viterbi algorithm introduced in the conventional modleing. And the travel distance between successive road points is used to determine the transition probability.

(Ren, 2012) uses an exponential distribution with a topological index for the transition probabilty in the map matching of GPS data. Meanwhile, a method based on the movement pattern recognition and a monocular visual odometry are explored as supplements to assure uninterrupted pedestrian navigation services.

(Song, Lu, Sun, Huang, & Chen, 2012) adds a multiplicator of speed limit to the observation probabilty and empirically tune the parameter for varying sampling rates to reduce HMM breaks (due to very small transtiiton probability when actual path's length is much larger than the trajectory distance. And the multi-threading technology is used to improve the runtime efficiency.

(Torre et al., 2012) consider the matching with missing roads in the roadnetwork. In the transition probability, information that includes max out-degree of the road, backtracking for U-turn, etc. is derived following a rule base. The method is feasible to recognize missing roads using a move forward/backwards machanism within viterbi decoding controled by a predined cutoff distance.

(Oran & Jaillet, 2013) use a cumulative proximity weight rather the common choice of the shortest distance and the parameterization for robust accuracy performance over varying sampling intervals, which gain a small margin (~1.5%) in the test.

(Osogami & Raymond, 2013) finds a multiobjective path for the transition path using a convex combination of travel distance and turns and Maximum Entropy Inverse Reinforcement Learning for parameter estimation on travel routes only.

(G. Wang & Zimmermann, 2014) improves the online Viterbi decoding algorithm using ski-rental model to control tradeoff between accuracy and latency. By exploring the uncertainty in the current states, the output window size can be determined dynamically, and thus achieve an error- and latency-bounded performance compared to other online strategies.

(Assam & Seidl, 2014) uses Gaussian distribution of tangent distance between top-m gps points pattern and road point pattern. Its transition probability is a weighted distance in which the weight is estimated using a statistical test of likely road geometry transitions.

HMM variants

(Li et al., 2015) develop a HMM model for interpolation and extrapolation on either location or time. The model jointly estimates the traveled path, travel time and speed. In particular, the motion between successive observations, the probability of turns made at intersection, and the traveled time with inverse Gaussian distribution are considered. The model is trained using an efficient inference algorithm over millions of trajectory data. Extenstive experiments on multiple datasets show reasonable accuracy of model with full model setting.

(Lou et al., 2009a) proposes a ST-Matching method which comprises spatial analysis and temporal analysis. The spatial analysis is used to compute the Gaussian distribution of the closet distance from GPS to its nearby roads and the ratio of distance between neighboring observations to the shortest travel distance between them. Temporal analysis is used to compute the cosine distance between actual average speed and typical speed constraint on the shortest path. Then the multiplications are summed over the trajectory. A sliding window strategy is used for online processing. The method outperforms the Average-Fréchet-Distance –based method in the test on low sampling rate.

(Rahmani & Koutsopoulos, 2013) used an adaptive search region based on the characteristic of the local roads and design a comprehensive cost function of A* algorihtm for path finding. It extended the work of (Lou et al., 2009b) by considering non link addictive criteria such as overall path frechet distance in overall path finding.

(H Wei, Wang, Forman, Zhu, & Guan, 2012) proposed a HMM equivalent formulism with interchangeable term derived from either (Lou et al., 2009b) or (Newson & Krumm, 2009). The term sampling interval is used in the max weight formulism to achieve a robust accuracy on varying sampling interval. The global weight is optimized by tuning two parameters beforehand to fit the training data.

(Srivatsa, Ganti, Wang, & Kolar, 2013) investigated the fitness of Markovian assumption in trajectory modeling using Chapman-Kolmogorov equation and found that it doesn’t hold, especially for the ones with specific destinations. Based on the analysis, an optimal path-finding algrotim is used instead of viterbi decoding for overal likelihood maximization. The algoirthm finds the closet path among top-K shortest paths by iterative trials on taking alternative path at the longest roads from the previous trail.

(H Wei et al., 2013) combined global maximum weight and global geometric method e.g., Fréchet distance. Following a global geometric method by constructing a free space between graph and trajectory, a previous formulation (H. Wei et al., 2012) is used in the dynamic programming in order to find the optimal path subjected to Fréchet distance.

(Tao & TIMMERMANS, 2013) applied Bayesian Belief Network (BBN) to combine multiple decision factors for matching. More specifically, a tree structure graphical representation binary classifier for individual GPS position is built which incorporates six decision variables, e.g., PDOP, DirectionDiff, DistToRoad, Connectivity, AngleDiff, RoadAzimuth. The model is trained with a small set (<1000 samples). The map matching also starts with an origin-detection process to calibrate the origin point using user profile (associated in the test data).

(Hunter et al., 2013) developed a path inference filter using a chain strucutre Conditional Random Fileds (CRF) with a small set of features for map matching. The model is suggested to be superior than HMM by covercoming the selection bias problem in transiton path selection. With benefit from the discriminative power of the CRF, the model is designed to fit the data better than HMM with a richer, non-independent feature set.

Empirical Probabilistic Model

(Giovannini, 2011) studied map matching for traffic flow analysis and proposed a four-step solution, including data aggregation to reorganize, modify, remove errorneous raw GPS data, affinity-based data matching in which the affinity is computed as the product of position and direction error distribution based on Cauchy distribution to retain the sensitivity on the tails in the distribution. With the data projection, each datum is identified to have several candidate matches. With these alternatives, a refined A* algorithm is used to find optimal paths with the shortest travel time while satisfying the constraint from the data recording. Eventually a global optimal path is found, using solely the travel time cost. Being different from other map matching methods, the weight of individual candidate road is not used in the global path finding.

(Bierlaire, Chen, & Newman, 2013) developed a probabilistic measurement model of smart phone data for map matching. The model captures the dependency between the observed position sequence and a hypothetical path over continuous position space along the path using integrals rahter than summation in HMM. The topology of the road network, DDR (domain of data relevance) is used to reduce solution space for an efficient integral computation. A traffic model based on speed patterns, i.e., stop, low speed, regular speed traveling is used to describe motions between succesive observations. Furthermore, a path generation algorithm is developed to find path and update the likelihood iteratively. (Chen & Bierlaire, 2013) extended the probabilistic measurement model from (Bierlaire et al., 2013) for multimodal map matching of smartphone data which were derived from sensors of GPS, Bluetooth, accelerometer.

(Sarlas, 2013) employed a route choice model for transition path identification. This model computes the probability of the candidate paths, which are selected among a list of shortest paths with added randomness.

(Westgate, 2013) investigated map matching for travel time estimation using Bayesian approach. By directly modeling the persistent bias in GPS data, the traveled path is modeled as missing data with GPS error and its unchanged bias in the likelihood of the data, which explores three statistical characteristics, namely individual GPS readings, multinomial logit choice model for unknown traveled paths as a function of traveled time, and lognormal distribution for the travel time between successive GPS observations. The probability is computed using the Metropolis-within-Gibbs framework. A test on simulated data show that the model with both GPS bias and independent error outperforms the reduced method in true and false positive rates.

As can be summarized from the aforementioned research works, the development of map matching reveals the follwing characterstics. Firstly, HMM-based methods have dominated the field and demonstrated a superior accuracy. The modeling of the transition probabiity is the most challenging step with the HMM and shows significant impact on the overall accuracy of the model. Unfornately, the real world route planning is a complex decision-making procedure which is not yet full understood. Therefore, current practice often requares a lot of enginneering efforts in designing the probability which needs combined intuitions and heuristic rules. Moreover, the parameters governing the probabiltiy or the weight measure is eitheir estimated from the data or predefined based on empirical evidence. Secondly, test datasets were different, not much effort was made in preprocessing the data, which also leads to confusion in applying and evaluating the proposed methods. Thirdly, map matching is still earning much attention in the community with an increasing need in processing large scale trajectory data for variosu accademic tasks.

2.3.2. Inferring Taxi Status

Inferring taxi status means to classify taxi's occupancy for each position log in the trajectory data. More specifically, a taxi turns its status to *occupied* when it picks up passengers and switches to *vacant* when it drops them off. Taxis tend to reveal different traveling behaviors in these two statuses. For example, occupied taxis have specific destinations so that they are more likely to take the fastest paths which ensure more profits, but vacant taxis tend to slow down and search around in the local streets for passengers[17]. Therefore, the capability of identifying taxi trips of being occupied can be useful for many applications such as intelligent routing service that incorporates regular traffic information (Yuan, Zheng, Zhang, & Xie, 2010), identify the pick-up/drop-off hotspots in the city for better taxi service recommendation, and even estimate the traffic demands for better urban planning. To acquire this information, it's straightforward to manually mark the starting points and ending points with meter installed in the taxi, and integrated the data with the GPS recordings. Unfortunately, there exist some GPS datasets without the meter data and thus give rise to the study of taxi status inference (Ganti, Srivatsa, Ranganathan, & Han, 2013; Zhu et al., 2011).

Only a few literatures have directly addressed taxi status inference problem. Therefore, some related works in mining taxi mobility data, activity recoginition, transportation mode detection are also selected for methodolgy comparison.

(Zhu et al., 2011) investigated the problem of inferring from GPS trajectories the taxi status of being occupied, vacant and parking. First, a parking place detection algorithm is developed to find the parking point sets in the trajectories, which uses of a density-based algorithm for candidate point sets and a supervised model to reduce the false detection of traffic jam. Then the non-parking trajectories are classified via a two-phase inference model. For individual GPS observations, a decision tree with probabilistic outputs is used and fed

---

[17] The service could vary in different cities, e.g., instead of searching passenger on the street, the taxis in Munich tend to wait at specific sites and pick up passengers with a reservation at specific locations.

with features extracted from trajectory alone, historical trajectories, road network and POIs. And for observation sequences, a Hidden Semi Markov Model (HSMM) is used to capture various duration patterns in the output sequences of the decision tree. Experiments show that parking status is relatively easy to recognize, while for the other two statuses, even the training with five times so much test data an accuracy of only 75% is achieved.

(Ganti et al., 2013) introduced the distance/time stretch factor for taxi pick-up/drop-off point inference following the heuristic that taxis take the shortest path when they are occupied. Then a HMM model is developed to incorporate this feature in the emission probability computation. The output of HMM model is post-processed by a clustering algorithm for pointwise decision on the final outputs. The parameters of the model, stretch factor and window size, are empirically chosen for the corresponding test datasets. Despite its simplicity, the model outperformed baseline method by a factor of 2 in the extensive experiments.

(Phithakkitnukoon, Veloso, Bento, Biderman, & Ratti, 2010) studied the problem of predicting the number of vacant taxis given the location and time. A predicator based on Naïve Bayes classifier is built independently for each cell in the study area, which accounts the time of the day, the day of the week and the weather condition. The work discusses the error-based learning for parameter estimation and evaluates the adequacy of data using mutual information. The study case reveals that traffic demands vary across the urban area and the regions with larger demands often have higher variances.

(L. Liao, Fox, & Kautz, 2007) developed an acitvity model using hierachical Conditional Random Fields for place extraction and activity recognition from GPS traces. The model is built with three layers, the lowest layer consists of GPS readings (all matched to the road patch), the middle layer contains activity nodes, and the top layer consists of significant places. This graphic structure enables the model to capture complex dependencies among different abstract layers, but with a high compulational load for inference and parameter estimation. Therefore, approximate inference and learning algorithms are used for efficient reasoning.

(L. Zhang, 2014) investigated the problem of classificaton of six transportation modes from the GPS tarjectories. First, a trajectory is segmented into sub-trajectories by identifying stops using a greedy search algorithm with prefined rules. Then a multi-stage classification method is developed to detect transporation modes recursively. In the first stage, fuzzy-logic is used to detect walk, bike and the mode with motorised vehilces. Then a supervised SVM is used to classify the rest of the modes. The sequential dependency among the modes is also explicitly applied after the classificaton in the first stage.

(Yu Zheng et al., 2008) proposed a method to learn the transporation mode from GPS trajectories. The method empolys a segmentation procedure, an inference model, and a post-processing procedure. During the segmentation, the change points in the trajecotry are detected, following a set of predefined rules with multiple threshold parameters. Then the segmented sub-trajectories are classified using both structured prediction (e.g., CRF) and single output classifiers (e.g., Decision Tree, SVM) for comparisons. Some of the latter ones are used in postprocessing so as to enforce a transportation mode transition in the post-processing. The experiments show that the segmentation based on change points outperforms the segmentation based on uniform duration and uniform length. The inference based on Decision Tree outperforms the inference based on CRF.

It can be noticed from the selected literatures that the task of taxi status inference is not well solved. (Ganti et al., 2013) reported that less than 80% recall is achieved on a realworld dataset[18] with an expected error range of 10m, and the performance drops dramatically when the sampling rate decreases. The inference of taxi status raises a number of intriguing issues.

First, there is a label uncertainty. The fundamental idea underneath the task is the mobility patterns (i.e. speed, direction) have the adeque information to infer the status/label in query, e.g., occupied/vacant, activities (work, sleep, leisure, visiting, etc), transportation modes (walk, bike, car, bus, tram). The real difficultiy relies in the distinguishablity of status. For example, walking is easier to identify than the driving mode by bus or car (L. Zhang, 2014; Yu Zheng et al., 2008). And for taxi status inference, simply applying common indicators such as speed and direction can not yield statisfying results (Ganti et al., 2013) even though it’s only a binary classificaition for the individual position logs.

Secondly, current practices intend to incorporate the heuristics in the inference methods and various approaches such as mutli-stage inference which recursively solves the labeling tasks (L. Zhang, Thiemann, & Sester, 2010), unified inference framework with complex structure (L. Liao et al., 2007), have been tested. Each of these methods enjoys certain advantages over the others. To ensure the overall performance, most works tend to employ the multi-stage strategy, e.g., segmentation with post-processing procedure to explicitly leverage the empirical insights.

Thirdly, despite the reasoning power of inference framework (e.g. graphcial models), model building can still be challenging as 1) contributions to the overall performance of the inference model may not be evaluated directly (Yu Zheng et al., 2008), especially when preprocessing and post-processing are invovled; 2) finding the most relevant feature variables may achieve a better performance than complex inference framework (Ganti et al., 2013); 3) the application of the structured prediction method (e.g., HMM, CRF) may suffer from a careless design of the input in terms of irregular temporal scale (e.g., length of the segment) from the preprocessing steps.

---

[18] Shanghai Jiao Tong University. SUVnet-Trace Data. http://wirelesslab.sjtu.edu.cn

CHAPTER 3.

Discriminative Models for Labeling Spatial Trajectories in Road Networks

---

Labeling spatial trajectories refer to the tasks involving location sequences such as localization in road networks (map matching of high sampling rate trajectories) (Newson & Krumm, 2009), route reconstruction (map matching of low-sampling-rate trajectories) (Hunter et al., 2013; Lou et al., 2009a; Yang & Meng, 2014), trajectory segmentation (Sankararaman, Agarwal, Molhave, Pan, & Boedihardjo, 2013) and activity recognition (L. Liao et al., 2007), etc. These labeling tasks need to handle uncertainty, whether due to the imprecise observations, partial observability, nondeterminism, or a combination of them all. There are numerous sources of uncertainty in spatial trajectories ranging from inherent errors of the positioning devices to the pragmatic aspect that only discrete statuses are recorded for the continuous movements (Trajcevski, 2011).

This chapter first investigates the uncertainty in labeling spatial trajectories in the urban road network, and then introduces the probabilistic graphical model, which lays the foundation of the thesis to resolve the uncertainty in labeling spatial trajectories.

3.1. Uncertainty in Labeling Spatial Trajectories

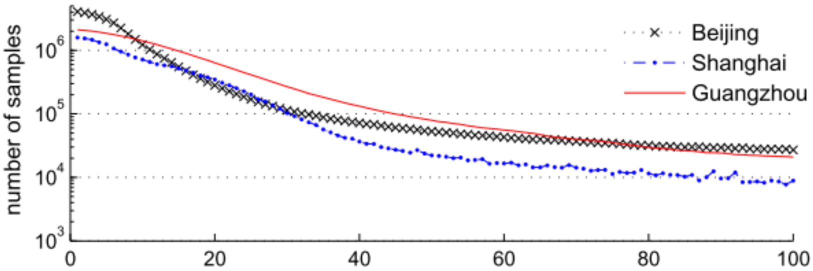
The uncertainty in the labeling tasks is discussed by addressing three types of sources, namely *imprecise positioning*, *sampling* and *nonlinear behavioral dynamics*. The first two root in the process of data collection while the last one unfolds the inherent nature of moving objects' behaviors.

3.1.1. Imprecise Positioning

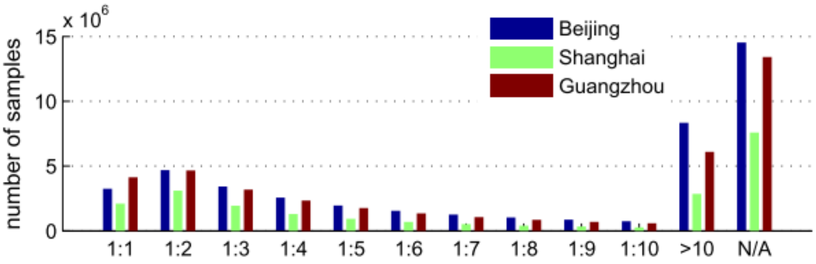
Moving objects can be located using a variety of positioning techniques, such as Global Positioning System (GPS), network-based techniques, etc. Regardless of the fact that nowadays GPS's Standard Positioning Service (SPS) provides a Global horizontal accuracy of 2.849 meters at a 95% confidence level (FAA, 2014), the positioning information often bears an uncertainty because a GPS receiver only approximates the actual position of the respective sensor or object due to physical limitations and measurement errors of the sensing hardware (Lange, Weinschrott, Geiger, & Blessing, 2009). Moreover, positioning with GPS-enabled devices in the urban context has been suffering from signal blocking and multipath effects which are still unsolved issues (Bourdeau, Sahmoudi, & Tourneret, 2012; Groves, 2011).

Besides the unsatisfactory positioning techniques, research practices often need to deal with noisy location data which may be:

- Legacy datasets that are collected with less accurate positioning devices.
- Crowd-sourced location data using a variety of unknown positioning devices with varying positioning accuracies.
- Inconsistent location data collected in complex urban environments (Figure 3.1 illustrates the estimation of varying positioning accuracies in three mega cities in China).
- Inaccurate reference data, e.g., outdated road network (Quddus et al., 2007).



(a) great-circle distance to the nearest road (m)



(b) distance ratio between 1st and 2nd matches in 100 m

Figure 3.1 Distance between location observations and the nearest roads (Y. Wang et al., 2011). A comparison of estimated positioning accuracies of taxis GPS data in three cities, Beijing, Shanghai and Guangzhou, in China. (a) Log-scale histogram of the count of distinct location observations according to their distance to the nearest road. (b) The ratio between the difference to the nearest and the second nearest roads within 100 meters.

Many research works have been done to model the uncertainty of the position information which can be categorized to *pdf-based models* and *shape-based models* (Lange et al., 2009). The pdf-based models employ two-dimensional probability density functions (e.g. two-dimensional Gaussian distribution) to describe positions under uncertainty at the specified moment. The shape-based models describe uncertain spatial extent of positions using geometric shapes with probabilities. In the context of labeling spatial trajectories, the uncertainty can be addressed using a similar notion of pdf-based models. But rather than using solely the position information of one observation, more information as well as more neighboring observations are used to resolve the issue collectively.

3.1.2. Sampling Rate

In practice, moving objects are observed at discrete time intervals, that is, trajectory data are the discrete samplings of the continuous movements. The sampling is necessary for following reasons:

- Efficient tracking and management of moving objects

  Efficient tracking and management aims to reduce the cost of collecting the trajectory data. The cost may refer to computation power in city-scale vehicle fleet management, on-board data storage and transmission cost in vehicle tracking, battery consumption of mobile applications, etc. In these cases, the sampling achieves cost reduction by filtering out the redundant location observations. Two strategies can be used for sampling, namely *time-based* and *distance-based*, both retain the observations at a specified time/distance interval.

- Users' engagement in the Location-based social network service (LBSNS)

  Another data source of spatial trajectories is the LBSNS, such as location-enabled tweets[19], check-ins in Facebook[20] and Foursquare[21], Microsoft's GeoLife[22]. Unlike the tracking and management, these trajectory data are collected voluntarily thus sampling unnecessarily complies with a universal rule among the diverse users. As a result, the sampling intervals could range from seconds to days based on individual user's engagement in the service, i.e., intensive user engagement leads to high sampling rate and vise versa.

Sampling introduces another uncertainty to the labeling tasks, which results in an information loss in the trajectory data. And the degree of the uncertainty varies when different sampling intervals are applied. As an example, Figure 3.2 shows four spatial trajectories of the same movement in the road network but with different sampling intervals. The 10s trajectory has the highest sampling rate (i.e. shortest sampling interval), thus it captures the finest details of the movement. The 30s, 60s, 120s trajectories are derived from the 10s using the time-based sampling strategy, which can only sketch the movement at coarse scales. In general, the larger the sampling interval is, the more uncertainty the data embodies. It can be verified in the same example (see Figure 3.2) that the route choices at the end of the 120s trajectory (bottom right sections) are totally “filtered out”. Note that the statement only holds if the moving objects share the equivalent mobility in the space they travel. In the context of road network, the mobility accounts for the connectivity of the local road networks and the speed limits on individual roads.

---

[19] https://support.twitter.com/articles/122236-adding-your-location-to-a-tweet#

[20] https://www.facebook.com/help/461075590584469/

[21] https://support.foursquare.com/hc/en-us/articles/201065340-Check-ins

[22] http://research.microsoft.com/en-us/projects/GeoLife/

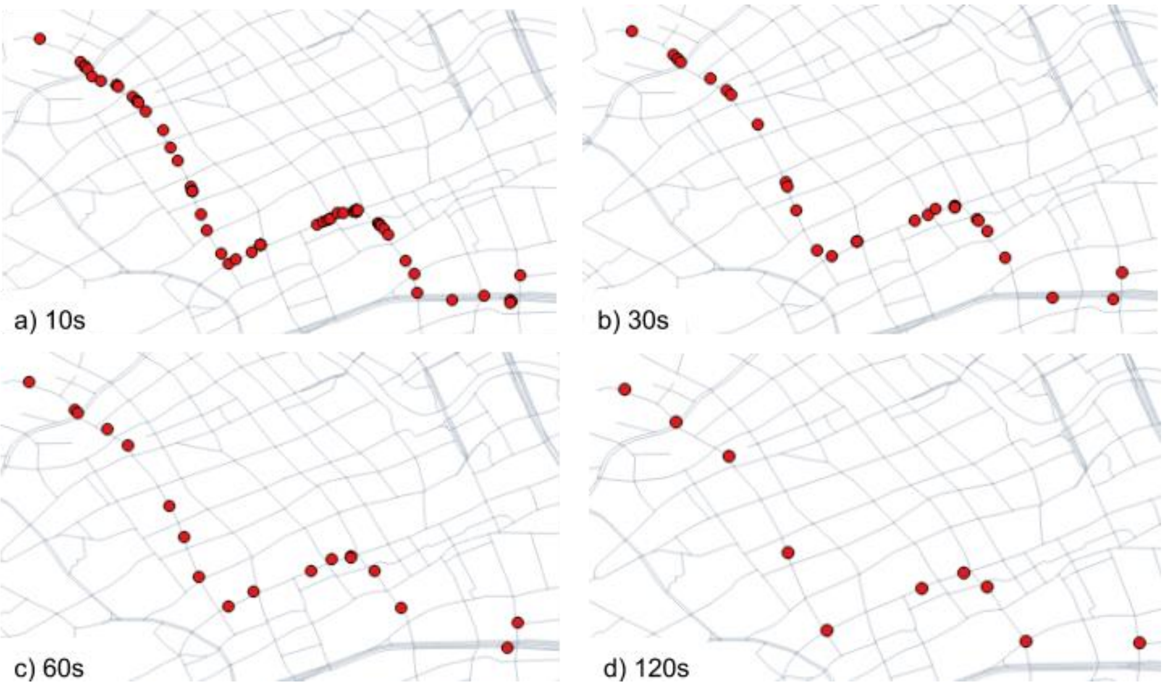


Figure 3.2. Trajectory data of the same movement in road network with varying sampling intervals.

3.1.3. Behavioral Dynamic

Inferring moving objects' behavior thus to extract semantics from raw location observations is a common goal of various labeling tasks. *Behavior*, in the context of labeling spatial trajectories, bears the meaning of the range of actions made by individual moving objects in conjunction with their moving space under various stimuli. Behaviors of interest could range from route choices made by drivers to the types of transportation modes (e.g. walk, bike, driving).

The basic assumption used in behavioral classification of spatial trajectories is that there exists a consistent mapping between moving objects' mobility states and their changes and the behavior of interest. Thus recognition of mobility patterns is crucial to the classification tasks. The mobility status can be described in a set of the *movement variables,* such as speed, sinuosity, turning angle, etc. Figure 3.3 shows the temporal dynamics of the movement variables, speed and turning angle, in contrast to the service status (i.e., binary code with 1 for *occupied* that the taxi is with passengers and 0 for *non-occupied* that the taxi is without passengers) of the taxi #10058. The data are obtained directly from the on-board GPS-enabled device in the taxi. Note that they can also be estimated from the location observations, in which case the scale issue should be considered for unbiased estimations (Laube & Purves, 2011).

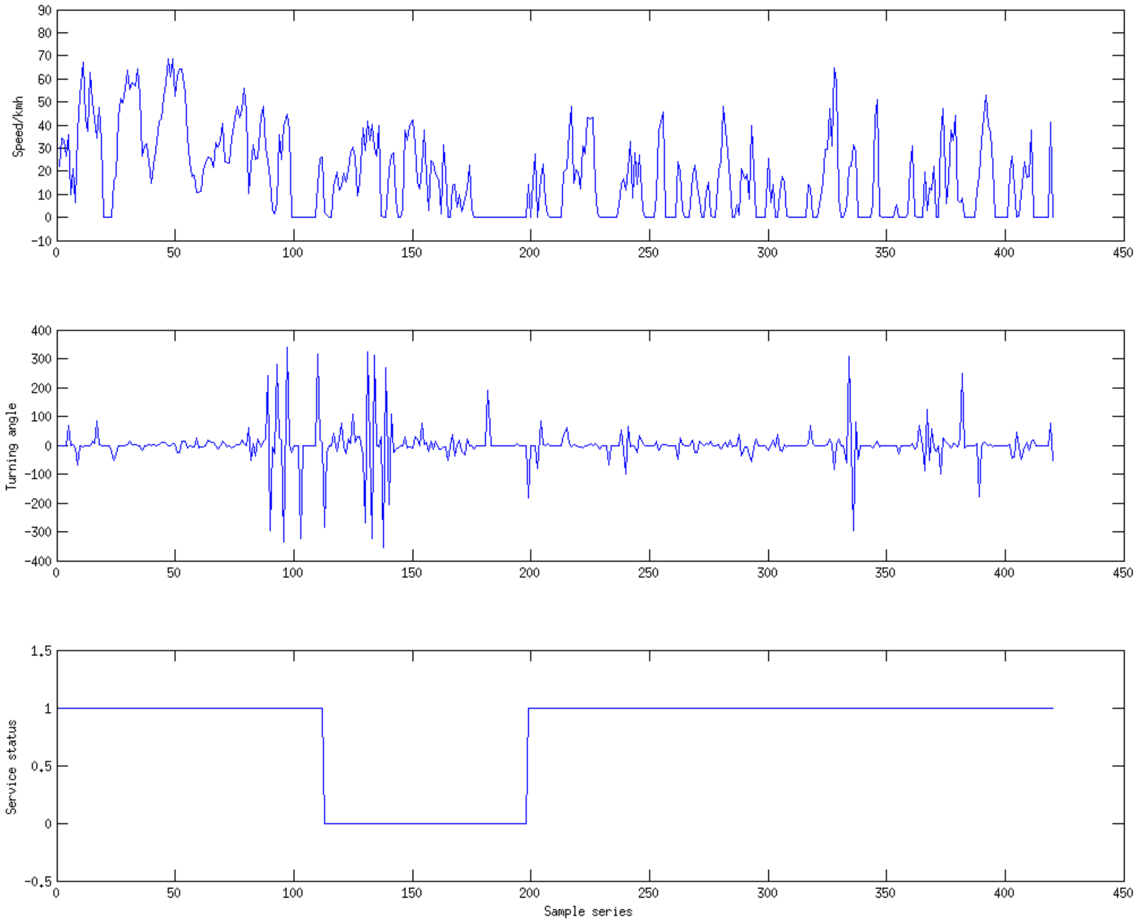


Figure 3.3 Temporal dynamics of the mobility status in terms of speed (top), turning angle (middle) of a sample series of the taxi #10058 and its status (bottom). The sample series starts at 12:25 and ends at 15:19 in 2010-4-1.

The behaviours of moving objects often reveal a dynamic nature as the movement variables change in a nonlinear way. This complexity gives rise to the uncertainty in the mapping between mobility pattern and the behaviour of interest. Take the case shown in Figure 3.3 as example, the taxi travels with irregular speed patterns as well as multiple in-between stops (probably at the traffic lights), which doesn’t’ align well with the changes of its status in the temporal dimension. The *turning angle* may have indicated the first switch of status for the increased turning angles (probably caused by the driving manoeuvres when approaching the destination) but failed to capture the next status switch. A possible interpretation, in this case, could be that the two exceptionally long stops (two speed line sections with constant zero values) can better match the switching events as they occur right before the taxi drop-off/pick-up passengers.

The above status inferring example illustrates the uncertainty associated with behavioural dynamics and motivate the need to find a relevant representation of the mobility pattern of moving objects for behavioural classification. As it can be shown in the later sections, the representation is the key issue to achieve good performance in the labelling tasks.

3.2. Discriminative Models for Sequence Labeling

The need for labeling sequence data, or predicting multiple variables that depend on each other over a set of definite states, arises in a wide variety of problems in several scientific fields. In information extraction, the task of Name-Entity Recognition (NER) require to tag text elements with pre-defined categories such as names of persons, organization, locations, etc. In computer vision, image patches are labelled with their semantic classes (since image data can be modeled as pixel sequence). And in geoscience, moving objects are localized, their behaviors classified, the evolving physical states of the geographic phenomena are predicted.

Probabilistic models describe data that can be observed from a system, and they are often used to infer unknown quantities and make predictions on unseen data (Ghahramani, 2012). In particular, probabilistic modeling for sequence labeling is to build probabilities of paired observation and label sequences in order to maximize the number of correct label assignments in the output sequence. Graphical models, a marriage between probability and graph theory, are well studied and understood for such problems.

In the remainder of this section, the fundamentals of probabilistic modeling for labeling sequence data including modeling, feature extraction, inference and learning are introduced.

3.2.1. Probabilistic Graphical Models

Probabilistic graphical models are a powerful framework which combines uncertainty (probabilities) and logical structure (independence constraints) to compactly represent complex, real-world phenomena (Koller, Friedman, Getoor, & Taskar, 2007). To realize that, a graph representation is used to explicitly address the dependency among the random variables which characterize different perspectives of a target problem with uncertainties. In this graph, nodes account for random variables and edges between the nodes claim the dependency between the corresponding variables. Many probabilistic models such as Hidden Markov Models, Kalman filters can be described using this general modeling language.

The motive of endorsing graphical representation in probabilistic modeling for multiple random variables is the compact yet powerful expressivity that it induces. Real world applications normally involve jointly modeling dozens or even hundreds of variables, i.e., a 100×100 image requires 10000 variables. And it can be daunting to describe them naively (a distribution with n binary random variables would need 2^n numbers). In contrast, a graphical representation describes a distribution in a compact way by exploring its underneath structure and allows it to be constructed and utilized effectively (Koller & Friedman, 2009). The local structures of the graph, cliques formed by a subset of variables that are fully connected, assert the *conditional dependencies* among the random variables. Meanwhile, the distribution represented by the graph can be broken down into a product of *factors*, each of which is defined on a much smaller possibility space rather than the one over all the variables. These dual perspectives of a graphical representation, namely a set of *conditional dependencies* and the *factorization* of the distribution, are found to be equivalent which are most useful in the modeling and design inference algorithms respectively (Sutton, 2012).

Graphical models comprise two classes of models, *Bayesian networks* (or directed graphical models) and *Markov networks* (or Markov random fields, undirected graphical models), see Figure 3.4 for illustrative examples. Bayesian networks use a directed graph in which edges have directions associated with them, while Markov networks use an undirected representation. These two classes of models share the merits of the graphical models but differ in the dependencies they can encode and the factorization that they induce (Koller & Friedman, 2009). Unlike Bayesian networks, it's not that intuitive for Markov networks to correspond a local structure in the graph to either probabilities or conditional probabilities. Markov networks utilize the notion of energy (origins from statistical physics) defined on the cliques in which nodes are fully connected in the graph, and derive the probability by normalizing the sum of the energy. Detailed discussion on this is given in the later section.

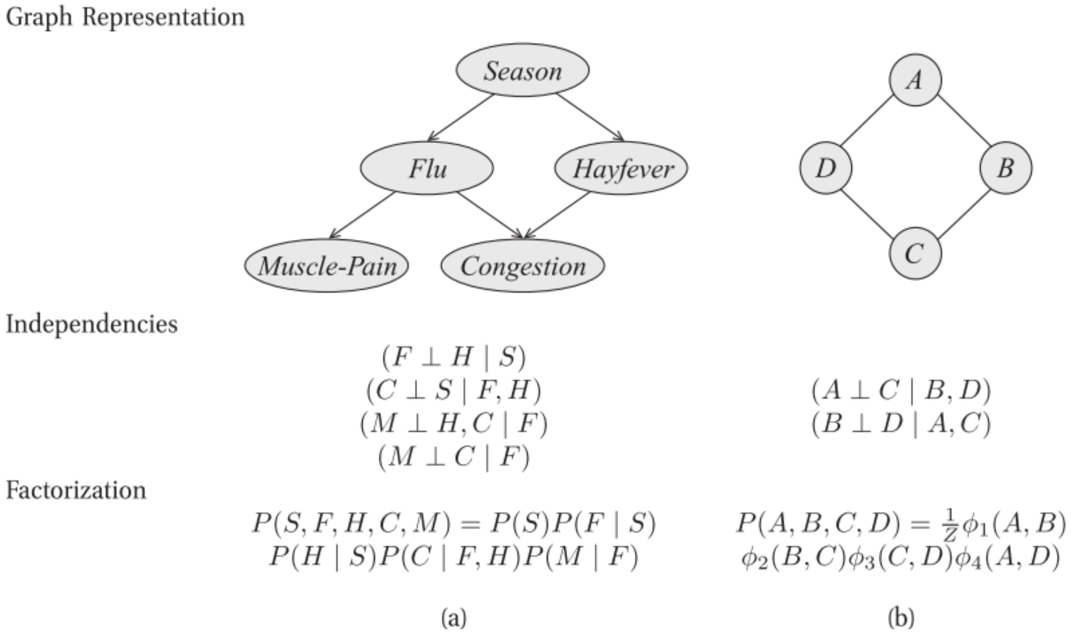


Figure 3.4. Different perspectives on probabilistic graphical models (Koller & Friedman, 2009): (a) medical diagnosis using a Bayesian network to infer the causal relationships among diseases (Flu, Hayfever) and symptoms (Muscle-Pain, Congestion), (b) a sample Markov network.

3.2.2. Generative versus Discriminative Classifiers

Modeling the probability of multiple random variables over a discrete set of states, namely predicting the output states/labels $\mathbf{y}$ given multiple observations $\mathbf{x}$ in classification, the models fall into two categories: generative models and discriminative models. Generative models use join probabilities of observations and output variables $p(\mathbf{x}, \mathbf{y})$, which intend to describe how the observations can be generated by the class variables. Discriminative models construct the conditional probability of the output variables given observations $p(\mathbf{y}|\mathbf{x})$. Though these two might be converted to each other using Bayes's rule,

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \underbrace{p(\mathbf{y}|\mathbf{x})}_{\text{Discriminative}} = \underbrace{p(\mathbf{y})p(\mathbf{x}|\mathbf{y})}_{\text{Generative}} \tag{3.1}$$

they are distinct approaches in practice and both have potential advantages in practice (Sutton, 2012). Table 3.1 gives a few examples for these two categories.

	Single output	Sequence output
Generative	Naive Bayes, Restricted Boltzmann Machine	Hidden Markov model
Discriminative	Logistic Regression, Support Vector Machine, Neural Networks	Conditional Random Fields

Table 3.1 Generative versus Discriminative Models.

Since it's tempting to know which class of models enjoys better performance of classification (the count ratio of correctly classified examples among all in the test), comparisons are often made by investigating a *generative-discriminative* pair of models such as naive Bayes and logistic regression for discrete input, Normal Discriminant Analysis and logistic regression for continuous input. (Ng & Jordan, 2002) appealed for such a purpose. Ng and Jodan argued with both theoretical and empirical evidences that the two classes of models may outperform each other with varying example sizes, the so-called "two-regime" behavior. That is, logistic regression creates fewer asymptotic errors (indicating a better theoretical performance), but it can only outperform the naive Bayes when the size of training examples has reached a certain threshold. And the empirical results reveal no general knowledge of how large the thresholds should be for different domains. Furthermore, they suggests that, in practice, the cold-start performance of logistic regression can often be improved via regularizations and a hybrid classifier that inherits merits from both models should be considered. And the superior performances hold for classifier for sequence output, such as Conditional Random Fields versus its generative counterpart Hidden Markov Model in many applications.

Discriminative models also enjoy advantages in the modeling stage and they may very well get along with rich, overlapping input variables, or say features. As shown in the Eq. (3.1), discriminative models don't need to model the interdependencies among the input variables $\{\mathbf{x}\}$ for $p(\mathbf{x})$, which is otherwise a difficult task. For example, in map matching of GPS trajectories, it's not straightforward to model the dependency between the width of the road and the number of the traffic lights on it when you attempt to incorporate contextual information of the road networks the moving objects are traveling in. And it could be troublesome to model these dependencies in generative models, which may either require to model the dependencies explicitly, thus raise the difficulty of tractability or retreat to simpler independence assumptions with may influence the performance. More detailed discussions can be found in (Sutton, 2012).

As for the graphical representation, it's natural to represent the generative models with the form $p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$ using directed graphical model, while discriminative models use more often undirected graphs.

3.2.3. Conditional Random Fields

The Conditional Random Fields (CRF) is an undirected graphical model used to compute the probability of a label sequence conditioned on the observation sequence (Lafferty, McCallum, & Pereira, 2001), namely segmenting and labeling sequence data. The model was first proposed for natural language processing (NLP) and outperformed previous methods of Hidden Markov model (HMM) and Maximum Entropy Markov Model (MEMM) on the task of Penn treebank part-of-speech (POS) tagging[23]. The superior performance as discussed in the paper, was achieved by retaining the discriminative nature of MEMM while solving the *label bias problem* of its kind. CRF was soon successfully applied to a variety of problems in NLP (Sha, Pereira, & Science, 2003), computer vision (He, Zemel, & Carreira-Perpinan, 2004), computational biology (Bernal, Crammer, Hatzigeorgiou, & Pereira, 2007), and so on.

To formulate CRF, let $\mathbf{x} = (x_1, \ldots, x_T)^\top$ denotes the observation sequence of length $T \in \mathbb{N}$, in which each entry $x_t \in \mathbb{R}$ is an observation at position $t = 1..T$ in the sequence, $\mathbf{y} = (y_1, \ldots, y_T)^\top$ denotes the associated label sequence that takes values from a finite set $\mathcal{Y}$, i.e. a simple case of binary classification $\mathcal{Y} = \{0, 1\}$. Then, a general CRF can be given as follows

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{y}_c, \mathbf{x}_c) \tag{3.2}$$

where c indexes a subset of variables $\{\mathbf{y}_c, \mathbf{x}_c\}$ in which variables fully depend on each other, ψ_c is an associated *potential* function of the variable set c that maps the inputs to a non-negative value, and $Z(\mathbf{x})$ is a normalization function of input $\mathbf{x}$ defined as

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{y}_c, \mathbf{x}_c) \tag{3.3}$$

and the $\sum_{\mathbf{y} \in \mathcal{Y}}$ means the sum of all possible labels settings. Thus, the CRF defines a probability that factorizes on $|\mathcal{C}|$ factors of the inputs and labels. In a graphical language, the model can be described using an undirected graph with $|\mathcal{C}|$ cliques. Note that equation (3.3) only gives a modeling framework that shows how the probability should be factorized in the graph and the final model is obtained by specifying the potential functions.

The potential function Ψ_c is often rewritten as $\exp(-\epsilon(\mathbf{y}_c, \mathbf{x}_c))$. The function $\epsilon(\mathbf{y}_c, \mathbf{x}_c) = -\ln \Psi_c(\mathbf{y}_c, \mathbf{x}_c)$ called an energy function has an origin in statistical physics and is used to describe the probability of a physical state that depends inversely on its energy (i.e. configuration of a set of electrons) (Koller & Friedman, 2009). The energy function can be specified using the weighted sum of a set of predefined feature functions $\{f_{c,k}\}$. Each feature function maps the values of the variables that it takes a real number, a *feature*. The *weights* $\{\omega_{c,k}\}$ indicate how compatible the features are in the specific classification task. Then it yields

$$\Psi_c(\mathbf{y}_c, \mathbf{x}_c) = \exp(\sum_{k=1}^{K_c} \omega_{c,k} f_{c,k}(\mathbf{y}_c, \mathbf{x}_c)) \tag{3.4}$$

By substituting Ψ_c in Eq.(3.2), a CRF yields a log-linear form of

---

[23] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \exp(\sum_{k=1}^{K_c} \omega_{c,k} f_{c,k}(\mathbf{y}_c, \mathbf{x}_c)) \tag{3.5}$$

The subscripts of feature functions indicate that for each clique, in the graph representation, a unique feature set can be used. But it is also allowed to encourage similarity among the cliques that have the same structure by specifying the same set of feature functions and *tying parameters*, namely use identical weights for the same feature functions in the cliques. This results repeated local structures in the graph.

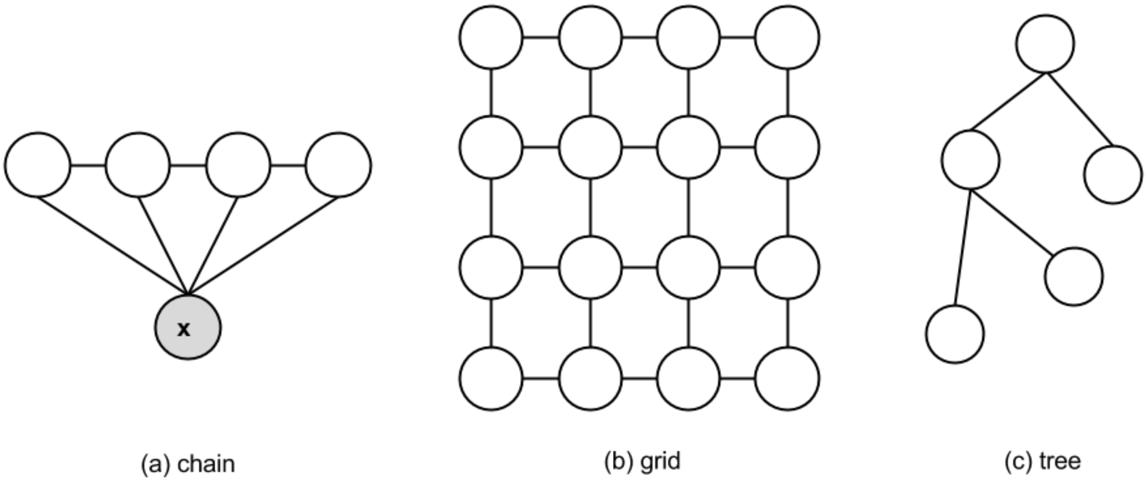
One typical implementation of CRF is to use a chain structure, in which dependency assumptions are made only between current label variable y_t and its preceding neighbor y_{t-1} and the same set of features with tied parameters are used. It yields

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp(\sum_{k=1}^{K} \omega_k f_k(y_t, y_{t-1}, \mathbf{x}_t)) \tag{3.6}$$

where

$$Z(x) = \sum_{y \in \mathcal{Y}} \prod_{t=1}^{T} \exp(\sum_{k=1}^{K} \omega_k f_k(y_t, y_{t-1}, \mathbf{x}_t)) \tag{3.7}$$

for which the graphical representation is shown in Figure 3.5 (a).



(a) chain

(b) grid

(c) tree

Figure 3.5. Different graphical structure CRF. (a) chain, (b) grid and (c) tree. Note that only the graphical structures for output variables are depicted for grid and tree.

3.2.4. Feature Extraction

Besides the CRF framework, feature functions should be defined in order to concretize the probabilistic models. The features, i.e., outputs of the feature functions, represent the raw data (e.g., word sequence, image pixels, GPS trajectory) in binary, categorical or continuous values. Note that a feature bears the same meaning to a variable, an attribute, or a predictor across the communities of statistics, machine learning and data mining. For instance, in the task of localizing a GPS point to local road network, features can be the binary indicator of the road being an arterial road or a residential street, the sphere distance between the GPS point and a nearby road, or a categorical transformation of the real-valued distance to qualitative values {VERY_CLOSE, CLOSE, FAR, VERY_CLOSE}.

Finding a good data representation is very domain specific and related to available measurement (Guyon & Elisseeff, 2006). Feature engineering is the procedure that serves such a purpose in classification tasks, which could be further specified as two tasks - feature construction and feature selection. Feature construction is the procedure to transform the input raw data into a set of features (or feature data), which is a key step conditioning subsequent classification procedure. Thus, it has been a long-term endeavor in machine learning community to explore new feature instances and methods to derive them for specific tasks. The task can be carried out either manually with human expertise or automatically. See (Sutton, 2012) for examples of how to design features of CRF for NLP tasks.

Often, to improve the classification result of the CRF, more features should be used (the feature set can be easily expanded using automatic feature induction (McCallum, 2003)). However, this leads to a dilemma that using more features also increases the risk of overfitting. Therefore, feature selection, to find the most relevant feature subset, is of great interest. Besides improving classification accuracy, it could also yields runtime data reduction in the memory, less complexity in parameter estimation (more features could result in more parameters to be estimated which will be explained in later section), and parsimonious models for easier statistical interpretations. Generally speaking, feature selection methods fall into three categories, filters, wrapper methods and embedded methods.

3.2.5. Inference on Graphical Models

Applications of probabilistic models can be realized using two different types of inference tasks, *probability query* and *finding most likely assignments*. Probability query computes $p(\mathbf{y_k}|\mathbf{x}_k)$, namely the conditional probability of a subset random variables $\mathbf{y_k}$ given a subset of observed random variables $\mathbf{x_k}$, e.g., query the probabilities of certain diseases given the symptoms of the patient. Finding most likely assignments is to find the assignments of the random variables with the maximum probability, for which a common case is to find the joint assignments of all the output variables $\mathbf{y}$ with the maximum probability given observed random variables $\mathbf{x}$ by computing $\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$.

The complexities of the inference tasks largely depend on the graphical structure of the probabilistic models. For simple graphs without loops such as chain and tree, both inference tasks can be computed exactly with efficient algorithms. However, for more general models with loop in the graph, e.g., grid structured model for image data, exact inferences are often intractable or suffer from a slow computational performance. Since real world applications or complex models often require intensive computation on a very large dataset, efficiency become a critical issue thus making exact algorithm less appealing in the practice. Therefore, inference tasks on complex model often resort to compute the approximate probabilities using either sampling-based methods, e.g., MCMC (Andrieu, Freitas, Doucet, & Jordan, 2003) or variational methods, e.g., Belief propagation (Yedidia, Freeman, & Weiss, 2001), for which the quality of approximation has to be evaluated. Note that inference often serves as a subroutine in learning the model since finding the optimal parameters requires the iterative evaluation of the model at each step. Thus, efficient training can often benefit from the selection of an efficient inference algorithm.

3.2.6. Discriminative Learning

Learning graphical models, in general, is to construct a model $p(\theta)$ from a set of data instances $\mathcal{D} = \{d_i\}_{i=1}^{N} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ for an underlying probabilistic distribution p*. The goal of learning might vary due to a range of different purposes, such as density estimation, task specific learning, and knowledge discovery (Koller & Friedman, 2009). Therefore, there exist various metrics to evaluate the learned model $p(\theta)$ in contrast to the truth distribution p*. As for discriminative models, the expected *conditional log-likelihood*

$$\mathbb{E}_{d_i \sim p^*}[\ln(p(\mathbf{y}|\mathbf{x};\theta))] \tag{3.8}$$

is often used to measure model's capability of predicting $\mathbf{y}$ given $\mathbf{x}$, where $\mathbb{E}_{d_i \sim p^*}[\cdot]$ denotes the expected value of the given quantity (computed from the model) with data instance d_i sampled from the distribution p*. The higher value the expectation gets, the better the model approximate the truth distribution.

Since the truth distribution is unknown for real-world applications, the expectation is often approximated by averaging over a sufficiently large data set $\mathcal{D}$. And by exploiting the standard assumption that the data instances are *independent and identically distributed* (IID), it yields to compute

$$\mathbb{E}_{\mathcal{D}}[\ln(p(\mathbf{y}|\mathbf{x};\theta))] = \frac{1}{N}\sum_{i=1}^{N} \ln(p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)};\theta)) \tag{3.9}$$

This numerical criterion serves as an *objective function* to quantify the preference of different models in terms of finding the optimal parameters θ*. Therefore, learning can be formulated as an optimization problem. The choice of learning algorithms depends on the specification of the model. Thus, we will address this issue accordingly in the subsequent chapters.

CHAPTER 4.

A Chain Structured Model

This chapter discusses the probabilistic modeling of spatial trajectories in road network using CRF. Map matching is performed to demonstrate the modeling effort, for which the author will show later that the derived model can also be applied to other applications.

Map matching is to recover the original route from a sequence of GPS observations (see Figure 4.1 for an example). More precisely, given a sequence of observations of a moving object, map matching finds the corresponding label sequence, namely the roads that are traveled on. The basic attributes of the GPS observations collected by positioning sensors include latitude, longitude and timestamp, while extra information such as instant speed, acceleration, heading direction etc. can also be obtained from the sensors. Another input data is the road network, which comprises the geometric representation and the navigational data such as speed limit, driving directions, etc. For vehicles' trajectory data, only roads for driving are required, while other transportation networks are needed in the multimodal routing scenario (Chen & Bierlaire, 2013).

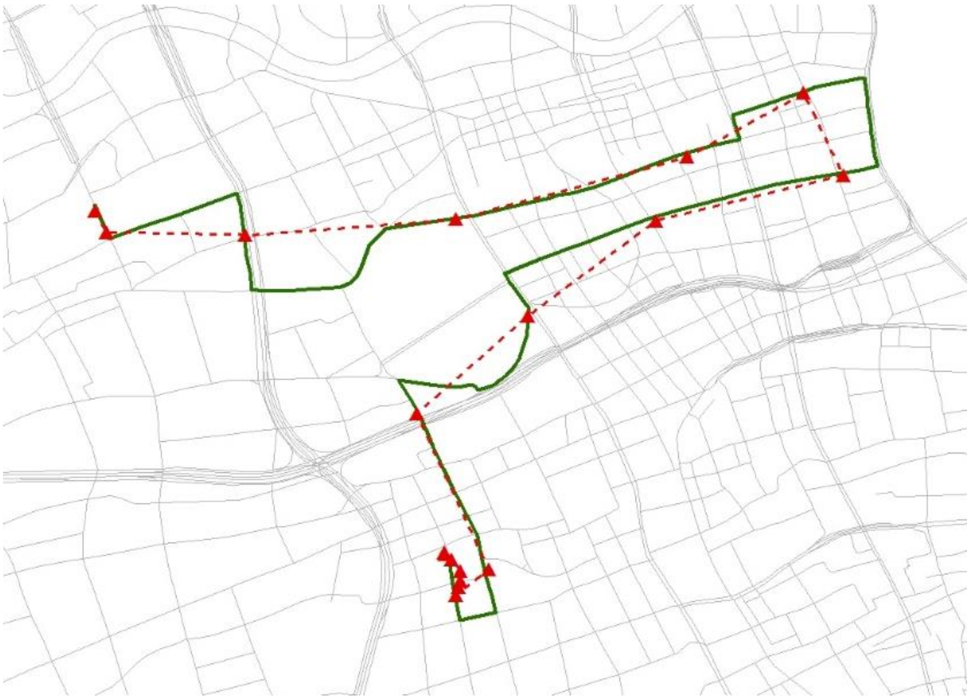


Figure 4.1 An example of map matching of spatial trajectory (red dash line) in road network (grey lines). The ground truth driving route is marked in a green line.

4.1. Chain Structured CRF

In map matching, two tasks need to be solved, namely localization of the individual GPS observations in the road network and finding the transition paths between two subsequent observations. Localization finds the actual roads where the vehicles' locations are observed. Finding the transition paths is to determine the actual path the vehicle takes in the road network. As discussed in Section 3.1, the GPS locations can be quite noisy in the urban area, which makes the localization based on finding nearest roads often unsuccessful in urban road network. This may be a reason for the standard approach to jointly solve localization and finding transitions in recent map matching research (Lou et al., 2009a; Newson & Krumm, 2009; Yang & Meng, 2015). Making use of the latent observations of the transition paths in the road network largely improve the matching accuracy for trajectories with high sampling rates, because it eliminates many infeasible candidates by imposing topological constraints. However, the performance drops dramatically when the sampling rate decreases, which imposes huge information loss in finding the transition paths. To address these challenges, a CRF-based probabilistic model is proposed.

The CRF model addresses the two tasks by employing a chain structure that comprises two types of nodes in the graphical representation, *point nodes* and *path nodes* (see Figure 4.2). These two types of nodes correspond to the alternating observations of the locations and the *transition paths*, and they jointly determine the output road sequence. Note that the transition paths are implicitly observed from both the trajectory data and the road network. Observations are paired with point nodes with edges for dependencies, while edges between point nodes and path nodes impose a feasibility constraint that is explained later in this section.

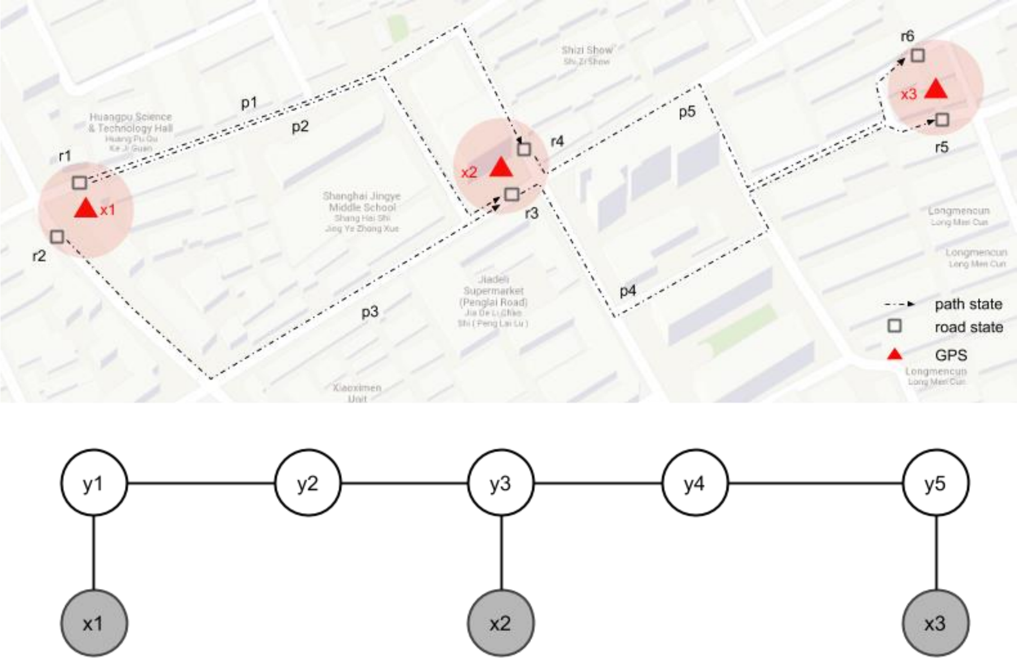

r1
r2
x1
p1
p2
p3
x2
r3
r4
p4
p5
x3
r5
r6
path state
road state
GPS
Huangpu Science & Technology Hall
Shanghai Jingye Middle School
Jiadeli Supermarket (Penglai Road)
Shizi Show
Longmencun
Xiaoximen
y1
y2
y3
y4
y5
x1
x2
x3

Figure 4.2. A chain-structured CRF for 3 GPS observations. The map on top illustrates a simplified situation of identifying roads and paths given GPS observations in the road network. This requires 5 random variables $y_1 \in \{r_1, r_2\}$, $y_2 \in \{p_1, p_2, p_3\}$, $y_3 \in \{r_3, r_4\}$, $y_4 \in \{p_4, p_5\}$, $y_5 \in \{r_5, r_6\}$ to build the CRF.

Thus, nodes y_1, y_3, y_5 linking with observations (shaded nodes) are point nodes while nodes y_2, y_4 are path nodes.

More precisely, given a sequence of location observations $\mathbf{x} = \{x_1, .., x_T\}$ of length $T \in \mathbb{N}$ and the road network $\mathcal{R} = \{r_i\}$. A sequence of random variables $\mathbf{y} = \{y_1, .., y_{2T-1}\}$ of length $2T-1$ are used for the sequence labeling. Let $t = 1..T$ be the position index in the sequence, observations, point nodes and path nodes are defined as follows

- Observations
  $\{x_t\}$ is the set of variables that represents observations of the moving object. Each variable x_t can be either a scalar or a multi-dimensional vector of a variety of sensory measurements of moving objects' behavior, e.g. location, instant speed, direction.
- Point nodes
  The point nodes $\{y_{2t-1}\}, t = 1..T$ are the random variables that resolve the uncertainty of the roads being traveled on. Thus, each node y_{2t-1} is connected to the observation x_t in the graph. The set of roads $\{r_i\}_{2t-1}$ associated with point node y_{2t-1} are called its *point states* in the road network $\mathcal{R}$.
- Path nodes
  The path nodes $\{y_{2t}\}, t = 1..T-1$ are the random variables that address the uncertainty in the transition paths. Each path node y_{2t} is defined on a set of transition paths $\{p_j\}_{2t}$ between subsequent observations x_{2t-1}, x_{2t+1}, which are called the path states of the path node y_{2t} in the road network $\mathcal{R}$. Note that a path state is modeled as a sequence of roads, i.e. $p_j = \{r_1 \ldots r_i\} \subset \mathcal{R}$.

Both point states and path states are generated from the observation sequence $\mathbf{x}$ and the road network $\mathcal{R}$. For each point node, the point states may include all the roads in the road network. However, it's often impractical since the size of the road network can be very large, especially for the mega cities. For example, the road network data of Shanghai, China from OSM has $109,271$[24].roads This leads to a very large point state space which raises huge challenges of computation complexity in terms of space and time in the model inference. Fortunately, it's often unnecessary to consider all the roads in the road network thanks to the accuracy achieved by nowadays' positioning technologies. Therefore, only those roads within the "confidence" range of the positioning sensor need to be included as the point states. Note that due to heterogeneity of the positioning conditions and the density of road networks, point nodes could have varying numbers of point states (see Figure 4.3).

---

[24] Statistic is made on data from https://mapzen.com/metro-extracts/ in Dec, 2014.

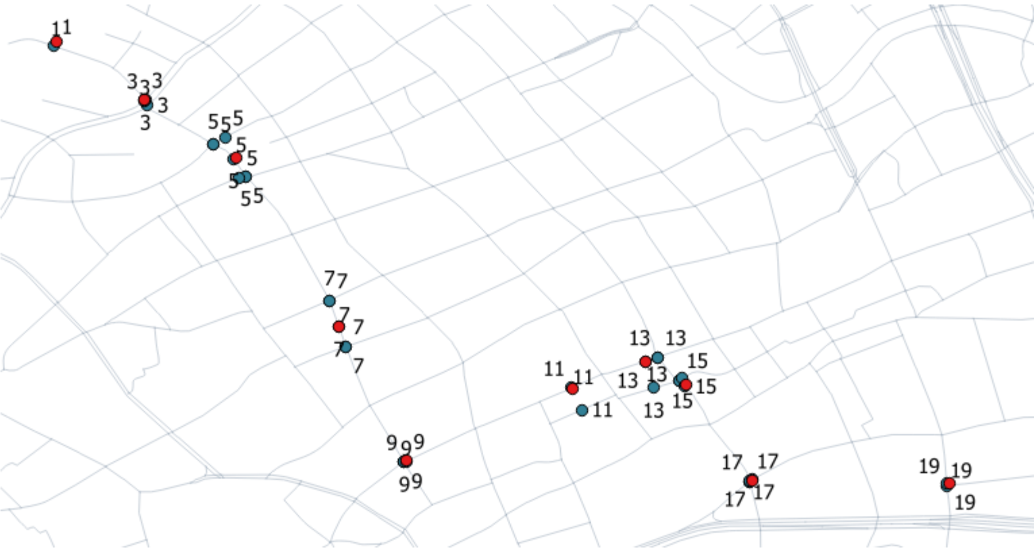


Figure 4.3 Mapping of a sequence of location observations (red dots) and their projections on the associated point states (blue dots). The 1st Node has only 1 point states while the 5th node has the most, 7 point states.

For each path node, the path states $\{p_j\}$ can also be restricted to a finite set of feasible transition paths in the road network, for which “feasible” bears the meaning of satisfying all mobility constraints, e.g., driving directions on single roads, turn restriction at the crossing, maximum driving distance under speed limits in the sampling intervals. Unfortunately, it can still yield a huge number of alternative paths in the road network. Moreover, the number grows dramatically if the sampling rate decreases in dense areas of road network (see Figure 4.4). There are two strategies to address large sets of path states, eliminate redundant path states in state generation and apply efficient algorithms for model inference. Both strategies will be discussed in the following sections.

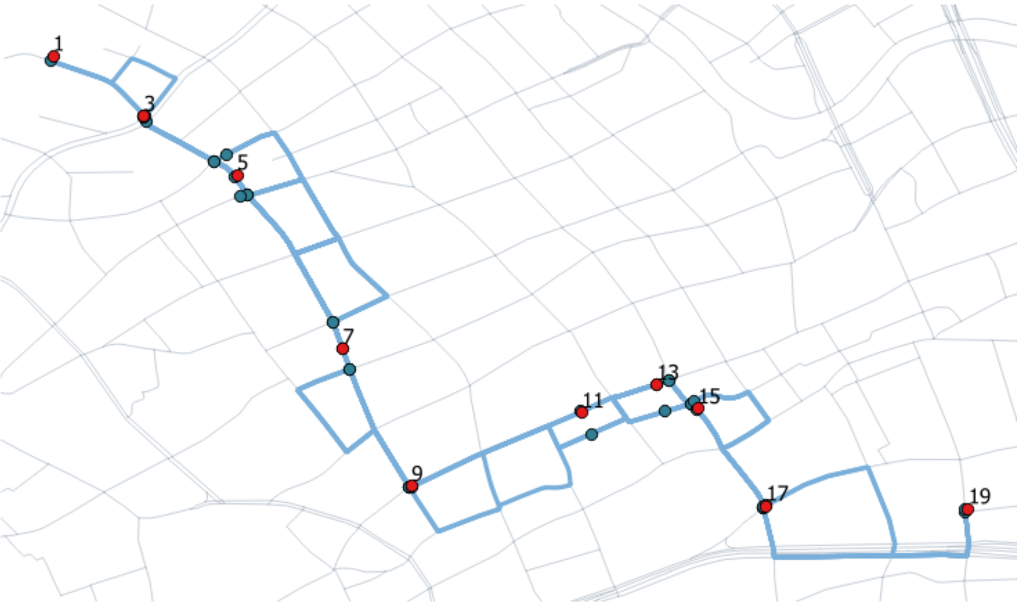


Figure 4.4 Mapping of path states (blue lines) of a sequence location observations (red dots). 35 alternative path states are found between the 5th node and the 7th node.

The rationale of using path nodes to explicitly model transition paths is that it allows the model to evaluate more than one transition path between two point states/roads. This may avoid an early elimination of ground truth for low sampling rate trajectories, for which ground truths may be the results of multi-objective routing rather than simply fastest path finding. Furthermore, modeling a transition path as a node rather absorbing it into the edge can reduce memory in use for model inference.

To complete the graphical model, edges are introduced to claim the dependencies among the nodes. As already shown in Figure 4.2, edges are used only between subsequent nodes yielding a chain structured CRF. The chain structure implements the *First-order Markovian assumption*, which assumes that the next node only depends on the current node and not on the sequence of preceding nodes. As it will be shown in later sections, this simple structure enjoys nice properties for model inference and learning. Given the graphical representation, the probability of labels $\mathbf{y}$ conditioning on the observations $\mathbf{x}_{1:T}$ is proportional to the product of the clique potentials

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &\propto \Psi(y_1, \dots, y_{2T-1}, x_1, \dots, x_T) \\ &= \Psi_1(y_1, y_2, \mathbf{x})\Psi_2(y_2, y_3, \mathbf{x}) \dots \Psi_{2T-1}(y_{2T-1}, \mathbf{x}) \end{aligned}$$

where Ψ_t is the potential function at position t, which can take neighboring labels and arbitrary observations as input. For the simplest case, it may be further assumed that the potential is time invariant (i.e. same potentials are used for the cliques at all positions in the chain), thus yielding a normalized quantity

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{2T-1} \psi(y_t, y_{t+1}, \mathbf{x}) \tag{4.10}$$

where normalization constant $Z(\mathbf{x})$ is derived by sum over all possible label sequences:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{2T-1} \psi(y_t, y_{t+1}, \mathbf{x}) \tag{4.11}$$

Note a dump label $y_{2T} = \text{End-Label}$ is used for the notation convenience. The CRF also has an exponential form of

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp(\sum_{k=1}^{K} \omega_k f_k(y_{2t-1}, \mathbf{x}) + \sum_{s=1}^{S} \mu_s g_s(y_{2t}, \mathbf{x}) + \ln h(y_{t-1}, y_t)) \tag{4.12}$$

where $f_k(\cdot)$ is *point feature function* defined on point nodes that expresses the compatibility between the observations and their point states (i.e. nearby roads), $g_s(\cdot)$ is *path feature function* defined on path nodes that reveals the utility of the path states (i.e. the transition paths), $\{\omega_k\}$ and $\{\mu_s\}$ are weights of the these feature functions that can be learned from the example trajectories data. The last term $h(\cdot)$ is a binary function that enforces the connectivity between the point states and path states. $h(\cdot)$ is set to 1 if the input point state shares either the starting road or the ending road of the input path state and 0 otherwise. This term can be seen as a fix to the loose modeling of the point states and the path states. And the design of the feature functions will be discussed in detail in Section 4.3.

4.2. State Generation

States encode the knowledge of interest to be extracted from the raw observation data. In some prediction tasks, states are defined already in the formulation of problem (e.g., binary states of *occupied* with passengers and *non-occupied* without passengers in the classification of taxis' states), while others require the generation of states for individual observation sequences (e.g., prediction tasks involve sequential localization such as map matching).

States and their generation are briefly explained to justify the modeling concerns in the previous section. In practice, however, this process is more important as it might appear to be, and it could affect both the accuracy and the efficiency of the prediction model. Therefore, the details are discussed in the setting of map matching in this section.

4.2.1. State Generation Workflow

As explained previously, two types of states are needed accordingly for two types of nodes in the CRF model, namely point states and path states. Point states resolve the uncertainty of localizing single GPS observations while path states recover the partial route choice between two GPS observation.

Theoretically, it's straightforward to consider all feasible states in the model thus to avoid the risk of missing true candidates. A common practice (Hunter et al., 2013; Lou et al., 2009a; Newson & Krumm, 2009) is to first search point states for all point nodes using a buffer with predefined radii and a threshold of the state's number and then using routing algorithm to find path states among two sets of the point states of subsequent point nodes for the path node. However, the problem arises when processing lowing sampling rate observation sequence in the mega-city road network:

- Running routing algorithm is time consuming in the large road network and the efficiency is dominated by the size of the input road network (Zhan & Noon, 1998). The size can be measured in terms of the number of nodes and edges in the graph representation of a road network;
- A large number of point/path states are generated, in which huge redundancy can be identified which thus turns to be a waste of computation in the later model inference.

In order to resolve the dilemma between the computing efficiency and the risk of missing true candidates, a state generation workflow (see Figure 4.5) is designed based on the above observations, which comprises four steps:

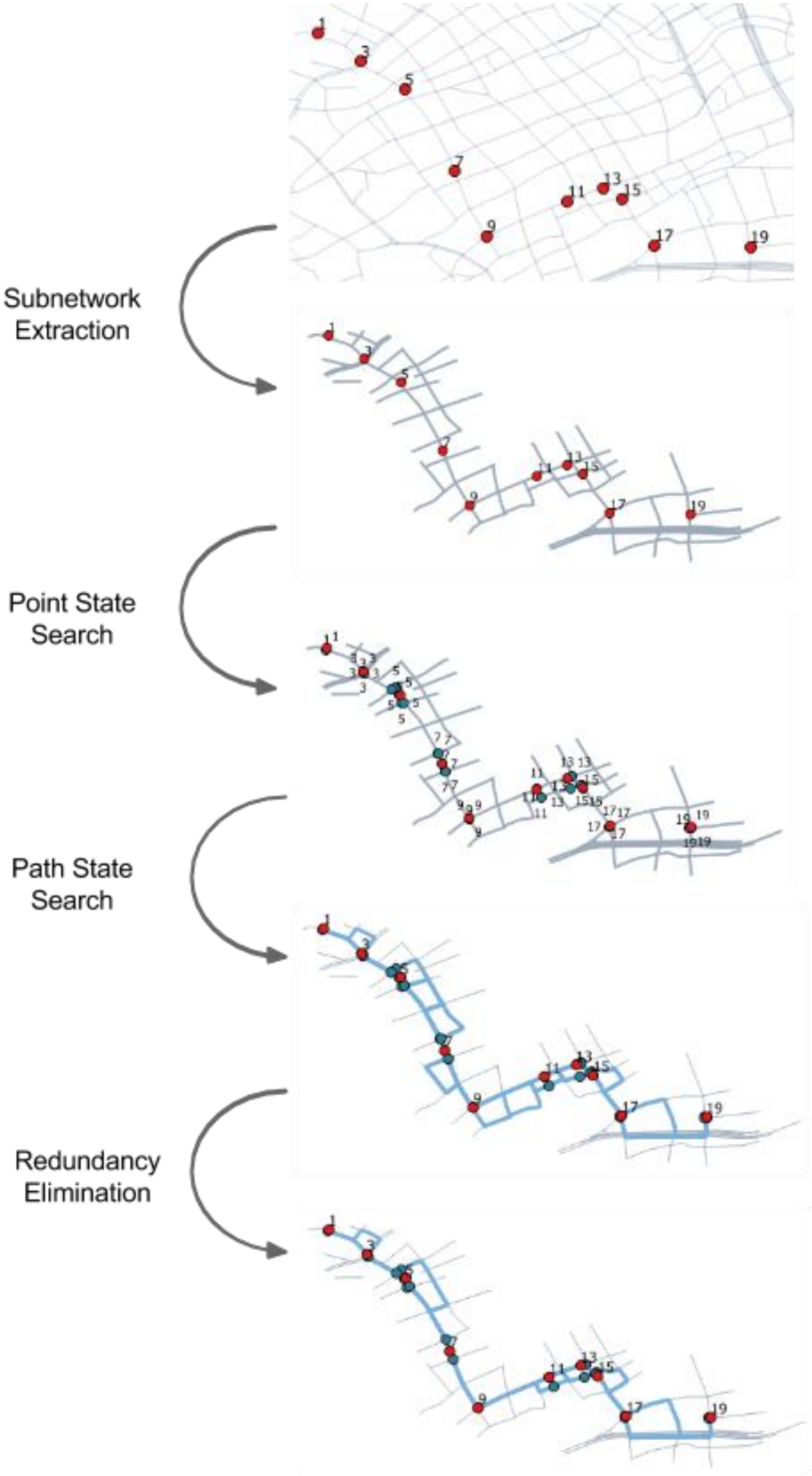


Figure 4.5. State generation workflow.

Step 1 Subnetwork Extraction

Extract a subnetwork of a much smaller size from the overall road network for current GPS trajectory using a feasible range. The feasible range can be implemented in two ways: 1) Free space feasible range. A polygonal range defined by a buffer query on the location point sequence. The radii are set to the maximum moving distance obtained from the maximum speed and sampling interval (Brakatsoulas & Pfoser, 2005); 2) Network-based feasible range. The feasible range is obtained based on the moving distance in the road network which takes into account the speed limit of the roads (K. Liu et al., 2012). Either way can produce relatively small subnetwork that counts as the upper bound of the uncertainty of the moving objects in the road network.

Step 2 Point States Search

Search point states for each GPS observation in the subnetwork with buffer query and remove co-located point states. The buffer query is carried out with predefined radii that represent the accuracy of the positioning devices. The redundancy in point states is defined as the co-location of two point states being at the end of the road segments, which is caused by the modeling of the road network (see Figure 4.6). For each co-located point cluster, only the one associated with the source road is kept which retains the larger possibilities of routing choices.

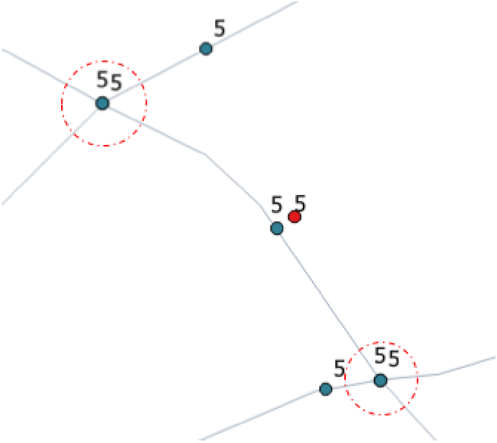


Figure 4.6. Example of point state (blue dot) redundancy of a GPS point (red dot). 4 redundant states are marked within red circles.

Step 3 Path States Generation

Search path states for each point state pairs in the subsequent point state sets in the subnetwork. The search is implemented using a top-K routing algorithm, Yen's algorithm (Martins & Pascoal, 2003), with predefined parameter K to control the complexity for certain point state pair.

Step 4 Redundancy Elimination

Redundancy Elimination using forward-backward message passing. The last step is to ensure the topological connectivity of all states, i.e. each point/path states should be able to proceed/backtracking to the states associated with the last/first GPS observations. The necessity of this procedure is that the previous search operation is carried out in a local scope that tends to violate the topological integrity. Rather than perform this connectivity checking on a single state basis, a dynamic programming algorithm is proposed for this purpose, which is discussed in the next section.

4.2.2. Redundancy Elimination

Redundancy elimination is to remove the point/path states that fail to connect to the states associated to the first and last point nodes. The global operation on the state sequence eliminates the redundancy introduced in the node-wise local search, thus reduces the computational burden on the later model inference, which involves enumerating the states for summation throughout the observation sequence (See Section 4.4). The problem can be illustrated in a simplified example (see Figure 4.7). In the example graph, each column of nodes represent the states associated to the point/path node at position t, links among them represent the available connections. In this case, two groups of states fail to stratify the condition of full connection, e.g. the state at the top of position 3 fails to connect to any of the states in the last position, namely position 4, and the states at bottom of position 3 and 4 fail to connect to the starting position. Note that the number of the states in each column does not have to be equivalent and tends to be very large in the practical cases.

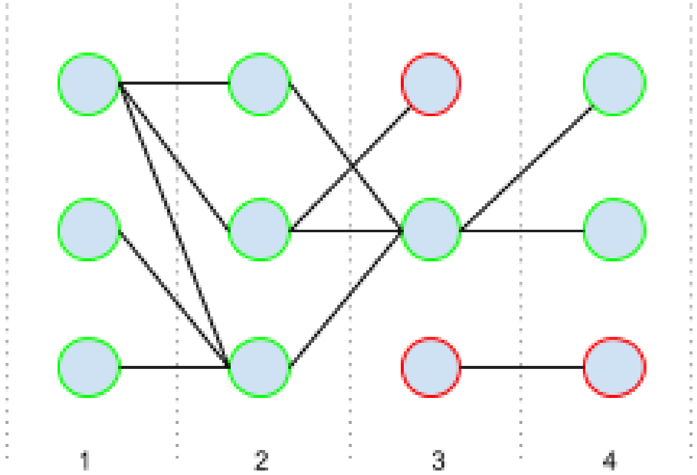


Figure 4.7. State transition graph of a 4 node sequence. Redundancy elimination yields to retain fully connected states (green nodes) while remove partially connected states (red nodes).

The graph illustration also lays the foundation of the proposed algorithm. Rather than checking the full connectivity for all states individually with a worst case computation complexity of $O(TM^T)$ for a sequence with length of T and state number of M at each node, a linear complexity can be achieved using dynamic programming on the above state transition graph.

Let $\mathcal{S}_{\text{ALL}}$ be the set of all the states (i.e., including both point states and path states) of a CRF model, S be a $M \times T$ binary path transition matrix constructed using all the path nodes in the CRF which contains $2T-1$ nodes and M states at each node, binary entry S_{it} at row i and column t of S is assigned to 0 if the path state is redundant and 1 otherwise, binary $e_{ij}^{(t)}$ is the associated weight of the link between S_{it} and S_{jt+1} is assigned to 1 if the two states have valid connection and 0 otherwise. Note that for a model that comprises nodes with vary numbers of states, M is set to the maximum state number. Thus, the algorithm is given as follows

```
Algorithm: REDUNDENCYELIMINATION
    Input:   $\mathcal{S}_{\mathrm{ALL}}$
    Ouput:   $\mathcal{S}^*_{\mathrm{ALL}}$
    Build path transition matrix $S$ from $\mathcal{S}_{\mathrm{ALL}}$;
    Set $S \leftarrow \mathbf{0}$;
    Set $S_{i1} \leftarrow 1$, $i = 1..M$
    // Forward pass
    for $t = 1..T-1$
        $S_{jt+1} \leftarrow \max_{i=1..M}(S_{it}e^t_{ij})$, $j = 1..M$;
    // Backward pass
    for $t = T-1..1$
        $S_{it} \leftarrow S_{it}\max_{j=1..M}(S_{jt+1}e^t_{ij})$, $i = 1..M$
    Remove path states in $\mathcal{S}_{\mathrm{ALL}}$ if their associated $S_{ij} = 0$;
    Remove point states in $\mathcal{S}_{\mathrm{ALL}}$ if they have none of the associated path states;
    Return $\mathcal{S}^*_{ALL} \leftarrow \mathcal{S}_{\mathrm{ALL}}$;
```

The algorithm is executed in two passes on the transition graph. For each pass the message "the current state links with any of the states at previous node" using $\max$ function to aggregate the binary codes, which result 1 if connected or 0 otherwise. Thus, the forward pass propagates the message "connected with any state at the first node" and the backward pass propagates for the last node. Only those nodes connected to both first and last node states remain 1 in the end. The link value $e^{(t)}_{ij}$ can be computed based on the rules that enforce path state transition behaviors, e.g. the value can be set to 0 if 1) two path states are connected; or 2) two paths are connected but result in a U-turn. Note that more rules can be added to restrict the desired transition behaviors. For observation sequences that have outliers with no fully connected path states, the algorithm would eliminate all the states of the sequences. To fix this, we simply assign the states of all the outliers to 1.

The effect of applying redundancy elimination varies from trajectory samples and the local road network. Take the example trajectory illustrated in Figure 4.5, more than 10% of original states are eliminated which can also be visually identified in the last two steps in Figure 4.5.

4.3. Features Extraction

Two types of feature functions are used in the model, namely point feature functions $\{f_k\}$ and path feature functions $\{g_s\}$. These features are arbitrary, real valued functions of the observations and the labels that allows the CRF to incorporate domain knowledge, such as the moving status of a vehicle could indicate whether it's traveling on the highway, the chosen paths should possess certain utilities that the drivers preferred.

In this section, we first given examples of features considered in the model, then discuss the issue of parameter tying.

4.3.1. Features

Two examples of feature functions are described as follows to show how features encode information.

Distance error feature. $f_k(y_{2t-1}, x_t) = \mathbb{I}(y_{2t-1} = r_i)\mathrm{dist}(x_t, r_i)$, this point feature informs the deviation of the GPS observation from the nearby road r_i, where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the expression in the bracket holds and 0 otherwise, $\mathrm{dist}(\cdot)$ is the distance from the GPS location of observation x_t to the road r_i. For a random variable y_{2t-1} over a set of finite roads $\{r_i\}$, e.g. y_1 over three roads $\{r_1, r_2, r_3\}$ at position t in the trajectory, three features are needed

$$f_1(y_1, x_1) = \mathbb{I}(y_1 = r_1)\mathrm{dist}(x_1, r_1)$$

$$f_2(y_1, x_1) = \mathbb{I}(y_1 = r_2)\mathrm{dist}(x_1, r_2)$$

$$f_3(y_1, x_1) = \mathbb{I}(y_1 = r_3)\mathrm{dist}(x_1, r_3)$$

Thus, this set of features allows the CRF to evaluate individually the deviation of the observation from the set of road labels being considered. Furthermore, if a univariate Gaussian distribution over the deviation $d_t = \mathrm{dist}(y_{2t-1}, x_t)$ rather than the deviation itself should be encoded (e.g., to capture the randomness in the positioning device which is often Gaussian), then the features take the form

$$\begin{aligned}\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(d_t-\mu)^2}{2\sigma^2}) &= \exp(\frac{-1}{2\sigma^2}d_t^2 + \frac{\mu}{\sigma^2}d_t + (\frac{-\mu^2}{2\sigma^2} - \log\sqrt{2\pi\sigma^2}))\\ &= \exp(\omega_k f_k + \omega_{k+1} f_{k+1} + \omega_{k+1} f_{k+2})\end{aligned}$$

And following features are used in the model

$$f_k = \mathbb{I}(y_{2t-1} = r_i)d_t$$

$$f_{k+1} = \mathbb{I}(y_{2t-1} = r_i)d_t^2$$

$$f_{k+2} = \mathbb{I}(y_{2t-1} = r_i)$$

Note that in practice, only one bias term, f_{k+2}, is used for one label.

Path length feature. $g_s = \mathbb{I}(y_{2t} = p_i)\mathrm{leng}(p_i)$, this path feature simply describes the travel distance in the transition path p_i, where $\mathrm{leng}(\cdot)$ is the sum of the total length of the roads within the path and the travel distance between the end roads.

A full list of features used in CRF for map matching is reflected in the following table.

Name	Type	Description
distance error	node/point	distance between location of observation x_t and the nearby road. To induce a Gaussian distribution of the feature, its square term is also used.
direction error	node/point	difference between direction of observation x_t and the tangent direction of its closest point at the nearby road. To induce a Gaussian distribution of the feature, its square term is also used.
length difference	node/path	difference of length of travel path and the distance between the locations of two subsequent observations x_t and x_{t+1}
length ratio	node/path	ratio of length of travel path and the distance between the locations of two subsequent observations x_t and x_{t+1}
length	node/path	length of traveled path
avg. speed limit	node/path	average speed limit of the roads in the path
min. travel time	node/path	travel time on the path with speed limits
# left turn	node/path	number of left turns the driver makes
# right turn	node/path	number of right turns the driver makes
highest road class	node/path	the highest road class in the path. The road class is an ordered attribute in the road network data, and a higher class indicates higher speed limits and better road conditions e.g. more lanes, wider and better road surface. See (“OSM Key:Highway,” 2015)
lowest road class	node/path	the lowest road class in the path
road class change	node/path	the number of changes of the road class
cosine speed	node/path	cosine distance of the speed limits and the speed of the observations x_t and x_{t+1}
time constraint	node/path	difference between the actual time and the minimum travel time in the path
avg. link length	node/path	average length of the roads in the path
# link	node/path	the number of roads in the path
path size	node/path	this attribute describes the correlation between the path state and other path states for the path node. See definition in (Frejinger, 2008)
path size (time-based)	node/path	this attribute is based on travel time for path state correlation
transition constraint	edge	indicator of feasible transition between two path states, with 0 indicating feasible and –inf otherwise

Table 4.1. Features used in the CRF for map matching.

4.3.2. Parameter Tying

As shown in the above examples, features reflect our domain knowledge of the specific problem and can be induced in a manual fashion. However, it’s often difficult to know what features fit best to the classification problem in advance, especially when little prior knowledge is given, features can be also induced in an automatic way (McCallum, 2003).

Thus, the number of features can grow quickly which also leads to a large number of parameters. For example, M feature functions with $|\mathcal{R}|$ road labels in road network $\mathcal{R}$ would need $M \times |\mathcal{R}|$ parameters, for which $|\mathcal{R}|$ is often very large. In map matching, it's often unnecessary to specify unique parameters for each road (the feature functions learn a universal knowledge for the entire road network rather than one single road). Therefore, the parameters of the features can be tied across the positions in the trajectory and it requires at most $M \times N$ parameters with N corresponds to the largest size of candidate label sets at each positions. Note that $N \ll |\mathcal{R}|$ thus it largely reduces the computation complexity of parameter estimation.

4.4. Inference on Chain

Efficiency is the most critical concern when choosing algorithms for the inference tasks on probabilistic graphical models. The two essential parts of the computation is discussed in this section.

1. **Normalization**

Computing the normalization given in Eq.(4.11) requires to sum over all possible label sequences. For graphical models with general structures the computation is intractable (Koller & Friedman, 2009). For example, for a chain with T random variables with M labels, the computation complexity of brute force method, namely enumerating every possible label sequence, is $O(M^T)$. Fortunately, an efficient computation can be achieved for the chain structured graphical model using dynamic programming algorithms. In this section, we discuss the normalization function using backward variable elimination.

By definition in Eq.(4.11), the normalization is given as

$$\begin{aligned} Z(\mathbf{x}) &= \sum_{\mathbf{y}} \prod_{t=1}^{T} \psi(y_t, y_{t+1}, \mathbf{x}) \\ &= \sum_{\mathbf{y}} \psi(y_1, y_2, \mathbf{x}) \psi(y_2, y_3, \mathbf{x}) \ldots \psi(y_{T-1}, y_T, \mathbf{x}) \psi(y_T, y_{T+1}, \mathbf{x}) \end{aligned}$$

Rather than computing the sum from outside-in, we can push the $\sum_{y_t}$ inside the product bypassing the potentials that don't depend on the summation and cache then the intermediate results.

$$\begin{aligned} Z(\mathbf{x}) &= \sum_{y_1} \cdots \sum_{y_T} \psi(y_1, y_2, \mathbf{x}) \ldots \psi(y_T, y_{T+1}, \mathbf{x}) \\ &= \sum_{y_1} \cdots \sum_{y_{T-1}} \psi(y_1, y_2, \mathbf{x}) \cdots \sum_{y_T} \psi(y_{T-1}, y_T, \mathbf{x}) \psi(y_T, y_{T+1}, \mathbf{x}) \\ &= \sum_{y_1} \cdots \sum_{y_{T-1}} \psi(y_1, y_2, \mathbf{x}) \ldots \beta_{T-1}(y_{T-1}) \end{aligned}$$

where

$$\beta_{T-1}(y_{T-1}) = \sum_{y_T} \psi(y_{T-1}, y_T, \mathbf{x}) \psi(y_T, y_{T+1}, \mathbf{x})$$

This term is a vector that stores the sum over the labels of y_T, thus eliminates the variable in the rest of the computation. By eliminating the variables backwards, the normalization finally yields

$$Z(\mathbf{x}) = \sum_{y_1} \beta_1(y_1) \tag{4.13}$$

In the computation, $T-1$ vectors are used to cache the computation results at position $t = 1..T-1$

$$\beta_t(y_t) = \sum_{y_{t+1}} \psi(y_t, y_{t+1}, \mathbf{x}) \beta_{t+1}(y_{t+1}) \tag{4.14}$$

$$\beta_T(y_T) = \psi(y_T, y_{T+1}, \mathbf{x}) \tag{4.15}$$

$$\beta_t(y_t) = \sum_{\mathbf{y}_{[t+1..T]}} \psi(y_t, y_{t+1}, \mathbf{x}) \prod_{t'=t+1}^{T} \psi_{t'}(y_{t'}, y_{t'+1}, \mathbf{x}) \tag{4.16}$$

Where the notation $\sum_{\mathbf{y}_{[t+1..T]}}$ indicates the sum over the chunk of the label sequence $\mathbf{y}$ starting from $t+1$ to T. This reduces the computation complexity to $O((TM^2)$ for T random variables with M labels, which is linear with respect to the length of the label sequence. Note that the chain structure ensures the variable elimination process to proceed along the variable list thus the expensive label enumerations are broken into a much smaller scale of two neighboring random variables.

2. Marginal Probabilities

As discussed before, there are two inference tasks that we are interested in, particularly in the context of CRF for spatial trajectory, computing the marginal probabilities $p(\mathbf{y}_{y_t=\text{label}}|\mathbf{x})$, and finding the most likely assignments. And similar to the normalization function, the computation of marginal probabilities is used as a subroutine in both inference and parameter estimation procedure. In this section, it's shown that the quantities can be efficiently computed using variable elimination in a forward-backward fashion.

Marginal probability of variable y_t taking label can be given as

$$\begin{aligned} p(\mathbf{y}_{y_t=\text{label}}|\mathbf{x}) &= \sum_{\mathbf{y}_{y_t=\text{label}}} p(\mathbf{y}|\mathbf{x}) \\ &= \frac{1}{Z(\mathbf{x})} \sum_{\mathbf{y}_{y_t=\text{label}}} \prod_{t=1}^{T} \psi(y_t, y_{t+1}, x) \end{aligned} \tag{4.17}$$

To compute the quantity given the normalization constant, only the right most term needs to be computed. Using the similar variable elimination trick for computing the normalization constant, we expand the product and cache the intermediate results yielded from both sides, i.e.,

$$\begin{aligned} p(\mathbf{y}_{y_t=\text{label}}|\mathbf{x}) \propto & \sum_{\mathbf{y}_{[1..t-1]}} \prod_{t'=1}^{t-2} \psi(y_{t'}, y_{t'+1}, \mathbf{x}) \times \psi(y_{t-1}, y_t = \text{label}, \mathbf{x}) \\ & \sum_{\mathbf{y}_{[t+1..T]}} \psi(y_t = \text{label}, y_{t+1}, \mathbf{x}) \times \prod_{t'=t+1}^{T} \psi(y_{t'}, y_{t'+1}, \mathbf{x}) \end{aligned} \tag{4.18}$$

Recall that we've used backward variable $\beta_t(y_t)$ (see Eq. (4.16)), and similarly we can define a forward variable which caches the computation results from the head of the label sequence.

$$\alpha_1(y_1) = 1 \tag{4.19}$$

$$\alpha_t(y_t) = \sum_{y_{t-1}} \alpha_{t-1}(y_{t-1})\psi(y_{t-1}, y_t, \mathbf{x}) \tag{4.20}$$

$$\alpha_t(y_t) = \sum_{\mathbf{y}_{[1..t-1]}} \prod_{t'=1}^{t-1} \psi(y_{t'-1}, y_{t'}, \mathbf{x})\psi(y_{t-1}, y_t, \mathbf{x}) \tag{4.21}$$

Thus, the marginal probability can be computed as

$$p(\mathbf{y}_{y_t=\mathrm{label}}|\mathbf{x}) = \frac{\alpha_{t-1}(y_t = \mathrm{label})\beta_t(y_t = \mathrm{label})}{Z(\mathbf{x})} \tag{4.22}$$

$\alpha_t(y_t)$ requires the same computing complexity as $\beta_t(y_t)$ which is also linear to the length of the label sequence. To implement, it only requires computing the sequence of the forward variables $\alpha = \{\alpha_1, \ldots, \alpha_T\}$ and the sequence of the backward variables $\beta = \{\beta_1, \ldots, \beta_T\}$, then the marginal probabilities can be computed by dividing the normalization constant.

3. **Finding Most Likely Labels**

Finding most likely labels is of direct interest in applications of CRF, e.g., it returns the most likely road sequence for map matching, which is to compute

$$y^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) \tag{4.23}$$

This requires no significantly different effort from the forward-backward algorithm discussed before, which also computes over all possible label sequences. The only operation is to replace the sum operation $\sum$ with max operation $\max$ while computing the quantities for each label assignment. The algorithm, called Viterbi, is discussed in detail in (Rabiner, 1989). The main steps are:

Step 1: In the forward pass on the label sequence, compute the maximum probability of the visited label sequence for each label at current position.

Step 2: In the backward pass on the label sequence, trace back the label entries with the maximum probabilities at each position.

4.5. Parameter Estimation

The inference requires learned weights of the feature functions thus to compute the clique potentials, which can be estimated by training CRF with labeled data. For example, in the context of map matching, labeled data is the GPS observation sequences with actual road sequences.

4.5.1. Maximum Likelihood

A common training scheme for CRF is to maximize the (conditional) log likelihood of the labeled data. With the i.i.d. (independent and identical distribtued) assumption on examples, the total likelihood is simply a product of the likelihood of individual examples (sum in the logarithm domain). Therefore, given a labeled data set $\mathcal{D}$ with N training examples $\{\mathbf{y}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{N}$, training the model (recall Eq.(4.12)) is to maximize

$$\begin{aligned}\sum_{\mathcal{D}} \ln p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = & \sum_{i=1}^{N}\sum_{t=1}^{T}(\sum_{k=1}^{K} \omega_k f_k(y_{2t-1}^{(i)}, \mathbf{x}^{(i)}) + \sum_{s=1}^{S} \mu_s g_s(y_{2t}^{(i)}, \mathbf{x}^{(i)})) \\ & - \sum_{i=1}^{N} \ln Z(\mathbf{x}^{(i)})\end{aligned} \tag{4.24}$$

in which the constraint term $\ln h(\cdot)$ is resolved to 0 since all point states and path states are connected in the labels.

From the perspective of optimization, the terms of point features f_k and path features g_s reveal no difference in the optimization procedure. Thus, we rewrite the log likelihood of the labeled data $\mathcal{D}$ with the parameter vector θ as

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{m=1}^{M} \theta_m q_m(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \ln Z(\mathbf{x}^{(i)}; \theta) \tag{4.25}$$

where $\theta = (\theta_m)_{m=1}^{M} = (\omega_1, \ldots, \omega_K, \mu_1, \ldots, \mu_S)$ is a real-valued vector which stacks all the $M = K + S$ parameters associated with the feature functions, and similarly $(q_m(y_t^{(i)}, x_t^{(i)}))_{m=1}^{M} = (f_1, \ldots, f_K, g_1, \ldots, g_S)$ is the stack vector for features. This formulation allows the following discussion in the framework of *log-linear model*.

In practice, a large number of features are often used to achieve better prediction accuracy. However, this also raises a risk of *overfitting*, i.e., the learned model manages to achieve a low prediction error rate on the labeled data yet fails to generalize it to unseen data. To resolve the dilemma of better fitting the labeled data and low generalization error, the log likelihood is often trained with a penalty term, so-called ℓ_2 norm, which is the negative sum of the quadratic parameters. Thus, training CRF with ℓ_2 penalized log likelihood is to maximize

$$\ell(\theta) - \frac{1}{2}\lambda_2 \sum_{m} \theta_m^2 \tag{4.26}$$

with respect to the parameters θ, where λ_2 is a non-negative hyper parameter that controls the amount of the penalty, i.e., the larger the value of λ_2, the greater the amount of penalty and 0 for no penalty.

In the log likelihood ℓ given in Eq (4.25), the first term $\theta_m q_m$ is linear w.r.t. the parameters θ, the second term $-\log Z(\theta)$ is a negative logarithm of a sum of exponentiated linear combinations of θ which thus is concave w.r.t. θ (Boyd & Vandenberghe, 2004). Furthermore, the negative ℓ_2 norm is a differentiable concave term. Thus, training the CRF model is to maximize an unconstrained concave objective function. The convexity ensures that the objective function has only one global optimum.

4.5.2. Gradient Ascent

A typical solution to maximize an unconstrained real-valued objective function is to use gradient ascent, which employs a line search strategy to iteratively approach the local maximum. Since the objective function used to train the CRF model is concave, the local maximum is the global optimum. After each iteration, the searching in the solution space goes a step more towards the direction in which the objective function increases. To determine how far to move along the direction, the evaluation of the log likelihood in Eq.(4.26) and its gradient is necessary

$$\begin{aligned}\frac{\partial \ell(\theta)}{\partial \theta_m} &= \sum_{i=1}^{N} q_m(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \frac{1}{Z(\mathbf{x}^{(i)};\theta)} \frac{\partial Z(\mathbf{x}^{(i)};\theta)}{\partial \theta_m} \\ &= \sum_{i=1}^{N} q_m(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \frac{1}{Z(\mathbf{x}^{(i)};\theta)} \frac{\partial}{\partial \theta_m} \sum_{\mathbf{y}} \exp(\sum_{m=1}^{M} \theta_m q_m(\mathbf{y}, \mathbf{x}^{(i)})) \\ &= \sum_{i=1}^{N} q_m(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}^{(i)};\theta) q_m(\mathbf{y}, \mathbf{x}^{(i)})\end{aligned}$$

in which the first term can be formulated as the expectation of m-th feature under empirical distribution, and the second term yields the expectation of the m-th feature under the distribution of CRF model. Thus, the above equation can be written as

$$\frac{\partial}{\partial \theta_m} \frac{1}{N} \ell(\theta) = \mathbb{E}_{\mathcal{D}}[q_m(\mathbf{y}^{(i)}, \mathbf{x}^{(i)})] - \mathbb{E}_{\theta}[q_m(\mathbf{y}, \mathbf{x}^{(i)})] \tag{4.27}$$

That is, the scaled gradient of m-th feature equals to the difference between the two expectations. When the gradient is zero, the two expectations are equal and thus the model fits the data best.

Computing the regularized likelihood requires the inference algorithm introduced in the previous section for the partition function. And computing the gradients needs the inference for the marginal probabilities. Note that the quantities depends on the input $\mathbf{x}^{(i)}$ thus it requires to run the inference whenever the likelihood is computed, which raises the efficiency concern of the inference on CRF. Furthermore, the gradient ascent is relatively slow in terms of the convergence rate; that is, it requires many more computationally expensive likelihood computations. This motivates the use of optimization methods with a fast convergence rate.

More efficient alternatives that also use a line search strategy are Newton's methods, which consider the trace of the search by including second-order derivatives of the function, Hessian matrix, in the updates of the parameters. This largely improves the rate of the convergence but requires to compute the inverse Hessian matrix which would take up a large memory space when the number of the parameters grows, thus leads to unstable intermediate computation result as most Hessians are poorly conditioned (Press, Teukolsky, Vetterling, & Flannery, 1992). Thus, quasi-Newton method, e.g. Broyden-Fletcher-Goldfarb-Shannon (BFGS) and its limited memory version, attracts a lot of interest as it approximates the Hessian rather than computing it directly, which is also empirically proven to be a success in the context of CRF (Sha et al., 2003). Other improvements are also investigated, such as conjugate gradient descent which constrains the directions of consecutive gradients to be orthogonal (Wallach, 2002), stochastic gradient descent that randomly selects one training example rather than scan over all of them in a single iteration thus yielding fast parameter updates (Vishwanathan, Schraudolph, Schmidt, & Murphy, 2006).

Among all the alternatives for solving the optimization problem in parameter estimation, we are interested in Limited-memory L-BFGS method for its superior performance in the NLP tasks (Sha et al., 2003).

4.5.3. Learning with Partially Observed Model

In practice, the data set $\mathcal{D}$ could have missing values. The problem arises when certain values were not collected for some examples in the data collection, or the variables cannot be observed. In map matching, missing values occur when the road network contains errors, e.g., missing roads with less traffic or incorrect road attributes for routing (double-way roads assigned with one way). An example of missing roads is illustrated in the Figure 4.8.

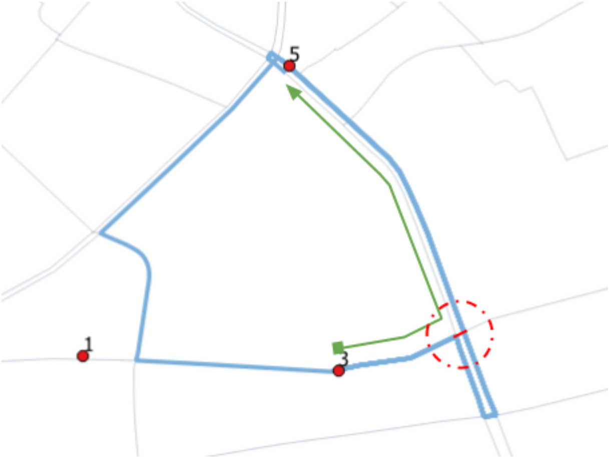


Figure 4.8. An example of missing values. Because of the missing road (red link in the red circle), the actual path (indicated by the green arrow) chosen by car cannot be covered by all the path states (blue lines) between 3rd node and 5th node.

A straightforward way to deal with missing values is to delete the partially observed examples from the training data, and to train only with complete data. However, the labelled data are often manually prepared, and it would be too expensive to remove the entire example data sequence for a small portion of nodes with missing values.

To address this issue, a common practice proposed in the literature (Koller & Friedman, 2009; Murphy, 2012; Quattoni & Wang, 2007) is to consider the variables with missing values separately using hidden variables in the formulation of the model and its log likelihood function which enjoys a similar learning scheme, e.g. gradient ascent (hidden means that the variables are not assigned with any values in the training data).

Let $\mathbf{y}_o$ be the vector of label variables observed by the model among all label variables $\mathbf{y}$, $\mathbf{y}_h$ be the vector of label variables with missing values in $\mathbf{y}$, $q_m(\mathbf{y}_o, \mathbf{y}_h, \mathbf{x})$ denotes the m-th feature on the sequence $\mathbf{y}$. Recall the model in feature stack form in Eq.(4.25), the partially observed model is given as

$$p(\mathbf{y}_\mathrm{o}, \mathbf{y}_\mathrm{h}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x};\theta)} \exp(\sum_{m=1}^{M} \theta_m q_m(\mathbf{y}_\mathrm{o}, \mathbf{y}_\mathrm{h}, \mathbf{x})) \tag{4.28}$$

And the log likelihood of the labeled data set $\mathcal{D}$ is

$$\ell_{\mathrm{partial}}(\theta) = \sum_{\mathcal{D}} \ln(\sum_{\mathbf{y}_{\mathrm{h}}} p(\mathbf{y}_{\mathrm{o}}^{(i)}, \mathbf{y}_{\mathrm{h}}|\mathbf{x};\theta)) = \sum_{\mathcal{D}} \ln(\frac{1}{Z(\mathbf{x};\theta)} \sum_{\mathbf{y}_{\mathrm{h}}} \tilde{p}(\mathbf{y}_{\mathrm{o}}^{(i)}, \mathbf{y}_{\mathrm{h}}|\mathbf{x};\theta)) \qquad (4.29)$$

where

$$\tilde{p}(\mathbf{y}_{\mathrm{o}}, \mathbf{y}_{\mathrm{h}}|\mathbf{x};\theta) \triangleq \exp(\sum_{m=1}^{M} \theta_m q_m(\mathbf{y}_{\mathrm{o}}, \mathbf{y}_{\mathrm{h}}, \mathbf{x})) \qquad (4.30)$$

is the unnormalized distribution. The term $\sum_{\mathbf{y}_{\mathrm{h}}^{(i)}} \tilde{p}(\mathbf{y}_{\mathrm{o}}^{(i)}, \mathbf{y}_{\mathrm{h}}|\mathbf{x};\theta))$ is the same as the partition function $Z(\theta)$, except that a subset of $\mathbf{y}$ is fixed at $\mathbf{y}_{\mathrm{o}}^{(i)}$. Similar to derivation of the gradient of the fully observed model, the gradient of the partially observed model is

$$\frac{\partial}{\partial \theta_m} \frac{1}{N} \ell_{\mathrm{partial}}(\theta) = \mathbb{E}_{\mathcal{D}}[q_m(\mathbf{y}_{\mathrm{o}}^{(i)}, \mathbf{y}_{\mathrm{h}}, \mathbf{x}^{(i)})] - \mathbb{E}_{\theta}[q_m(\mathbf{y}_{\mathrm{o}}, \mathbf{y}_{\mathrm{h}}, \mathbf{x}^{(i)})] \qquad (4.31)$$

where unlike $\frac{\partial \ell(\theta)}{\partial \theta_m}$, in both expectations, a subset of $\mathbf{y}$ are marginalized over $\mathbf{y}_h$.

4.5.4. Feature Selection via ℓ_1 Regularization

To achieve a better classification performance, a large number of features are used in the CRF. This yields a lower error rate on training data while raising the risk of high generalization error on test data. A common technique to tackle this problem is to add a penalty term to the objective function which penalizes learning large weights of feature functions in training. In this section, we discuss the other kind of regularization techniques, ℓ_1 regularization, and explain how to perform the feature selection with it.

ℓ_1 regularization adds an absolute term to the objective, which tends to reduce the weights to exactly zero in training. It has to solve

$$\arg\max_{\theta} \ell(\theta) - \lambda_1 \sum_{m} |\theta_m| \qquad (4.32)$$

where $\lambda_1 \geq 0$ again is used to tune the amount of penalty. The objective also remains convex while become non-differentiable at $\theta_m = 0$, which requires extra treatment to solve this optimization problem.

Having the advantage of producing a sparse model (having many parameter set to 0), optimizing ℓ_1 regularization has invoked a lot of interest in machine learning community. A variety of optimization methods are proposed to solve the problem. Since the convexity of ℓ_1-regularized objective ensures the finding of a unique optimal solution, those methods can be distinguished by how they handle non-differentiability of the objective function. Therefore, we mainly consider the efficiency in terms of running time while choosing optimization algorithms. Some comprehensive experimental reviews have been reported in (Schmidt, Fung, & Rosaless, 2009; Schmidt, 2010), which stimulated our interest in the Projected Scaled Sub-Gradient (PSS) methods for its fast convergence rate and consistent performance across different types of data set. We also find it more successful on GPS trajectory data.

Still, we have to choose the hyper parameters λ_1 and λ_2 which are difficult to determine in advance. As for λ_1, we tune the hyper parameters by evaluating the resulting error rates using a geometric sequence of decreasing from λ_{max} to 0, where λ_{max} is large enough to reduce all weights to zero. The justification of using a geometric sequence is that the target value is close to 0 and more trials are needed to approach it. And we use the same hyper parameter for ℓ_2 for comparison.

4.6. Chain CRF for Behavioral Classification

In this section, we show the use of a chain structured CRF for another type of sequential labeling task – inferring taxi status (i.e. occupied/non-occupied) from the spatial trajectories. In this task, binary states are given and thus state generation is not needed. This leads to one major difference in the semantics of the labels in map matching and status inferring that the labels in the latter task embody a more meaningful structure. Therefore, it’s interesting to study status inferring as a complementary of the map matching. In the remainder section, we explain the practical need of status inferring, and then discuss the modeling using a chain structured CRF.

4.6.1. Inferring Taxi Status

Inferring taxi status from taxis’ spatial trajectories is to determine the associated binary statues for each data point in the trajectory data, and we focus on the states describing whether a taxi is occupied by passengers in this thesis (see Figure 4.9 for examples). This information is useful for many applications, e.g., better understand the taxi demands across the urban area, recognize taxi anomaly for being non-occupied for exceptionally long period, identify occupied taxi trajectories for accurate traffic estimate (since non-occupied taxis usually slow down to look for passengers along the roads[25]). The practical need of solving this problem is that in some taxis the taximeters (i.e. the electronic device used to calculates passenger fares) are not linked to the positioning devices (e.g., GPS) (Zhu et al., 2011) and thus the status data are missing in the trajectory data. Therefore, it’s tempting to learn the mobility patterns from taxis with status information and use them to infer the taxi status where the data is missing.

The challenges of inferring taxi status in twofold: 1) the mobility pattern associated with the status is uncertain as illustrated in Figure 3.3 and thus it requires to develop more informative features; 2) for low-sampling rate trajectory, the geometric information of the trajectory is not available which often accounts for critical information to discover status related mobility pattern (Matsubara, Li, & Papalexakis, 2013). Previous study (Zhu et al., 2011) relies heavily on the map matching result and the POI data. However, we are interested to develop a solution using only the trajectory data and the road network. Note that a critical effort in status inferring is to identify the status transitions that corresponds to the activities of pickup/dropoff passengers, namely segment the sensor data sequence.

---

[25] This behavior may vary in different countries but it is the case in our test data collected in China.

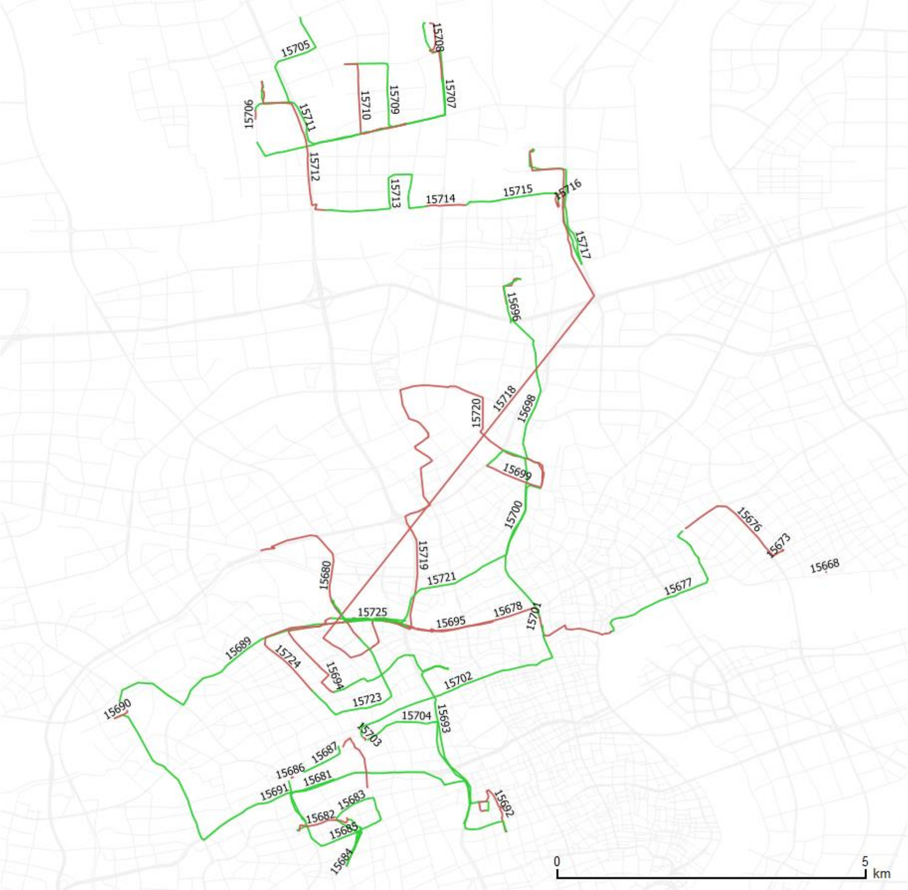


Figure 4.9 Mapping of service trajectories of a taxi in one day in Shanghai, China. Occupied trajectories are illustrated in green while non-occupied trajectories are illustrated in red. The trajectories are also marked with serial numbers which indicate the taxis' temporal activities throughout the day.

4.6.2. Model

Similar to the modeling for map matching discussed in previous section, probabilistic modeling with UGM requires the design of the graphical structure and a set of informative features. For taxi status inferring, we also use a chain structured CRF to model the sequential inputs but use a different definition of the label variables.

More precisely, given a sequence of location observations $\mathbf{x} = \{x_1, .., x_T\}$ of length $T \in \mathbb{N}$ and the road network $\mathcal{R} = \{r_i\}$, a sequence of random variables $\mathbf{y} = \{y_1, .., y_T\}$ of length T are used for the sequence labeling. Let $t = 1..T$ be the position index in the sequence, observations and nodes are defined as follows

- Observations
  Sharing the similar definition with map matching, $\{x_t\}$ is the set of variables that are represented as a multi-dimensional vector of a variety of sensory measurements of moving objects' behavior, e.g. turning angle, average speed.

- Nodes
  The nodes $\{y_t\}, t = 1..T$ are the random variables that resolve the uncertainty of the status of the taxis. Thus, each node y_t is connected to the observation x_t in the

graph. For each node y_t, there exists a set of binary status (i.e., occupied/non-occupied) of the taxi.

Using a chain structure, the probability of the label assignment sequence conditioned on observation sequence can be formulated as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \psi(y_t, y_{t+1}, \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \psi_1(y_t, \mathbf{x}) \psi_2(y_t, y_{t+1})$$

where ψ_1 is the unary potential defined on each observation-node pair using the prior knowledge of the taxis' mobility and spatial behavior such as slowing down to find passengers (relatively low average speed), less likely to find passengers on the highway. ψ_2 is the pairwise potential defined on the node-node pair for the label/state transitions (y_{T+1} is used as dummy term for brevity), $Z(\mathbf{x})$ is the normalization constant that sums over the label space of the node sequence. The exponential parametrization is given as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp(\sum_{k=1}^{K} \omega_k f_k(y_t, \mathbf{x}) + \sum_{s=1}^{S} \mu_s u_s(y_t, y_{t+1})) \tag{4.33}$$

where f_k and u_s are feature functions for unary potential and pairwise potential accordingly. Detailed specification of features used in the model is given in Table 4.2 Features used in the CRF for status inferring.

Name	Type	Description
window speed mean	node	Arithmetic mean of the speed of the GPS observations within the specified consecutive observations, *window*, depending on the taxi status. The function takes the form, $\mathbb{I}(y = \mathrm{Label})\mathrm{Mean}(t, W, speed)$, where $\mathrm{Mean}()$ is a window function centered at position t in the data sequence and computes the mean speed of the W consecutive observations. The window size is set to achieve an operation scope of 1mins, 2mins (e.g., for trajectory with 10s sampling interval, W is set to 7 to achieve 1mins).
window speed variance	node	Variance of the speed of the GPS observations in the specified window depending on the taxi status. The window size is set as the same as *window speed mean.*
window turning angle mean	node	Arithmetic mean of the turning angle of the GPS observations in the specified window depending on the taxi status. The window size is set to 2mins.
window turning angle variance	node	Variance of the turning angle of the GPS observations in the specified window depending on the taxi status. The window size is set as the same as *window turning angle mean*.
time of the day	node	Hour of the day of the GPS observation.
nonoccupied cluster index	node	Cluster index of GPS observations within the specified window for non-occupied observations. The cluster index is computed as the number of observations in unit area of the convex hull polygon that is generated by the set of observations. The window is set to 2mins.
status continuity	edge	Binary indicator that yields 1 if the previous node shares the same status. 0 otherwise.
speed change at pickup/dropoff	edge	Real-valued function that indicates the change in speed between two nodes. When they have different status labels, a taxi pickup or dropoff occurs.

Table 4.2 Features used in the CRF for status inferring.

Sharing a similar way of defining the feature functions as discussed in Section 4.3.1, the features for taxi status inferring emphasize the mobility pattern (i.e., mean, variance of the mobility variables such as speed, turning angle), temporal information and label transitions. As for the parameter tying, we share the parameters across the states for node features while explore a variety of tying strategies for label transitions.

Since the same graphical structure is used in the model for taxi status inferring, we invest no extra effort in choosing algorithms for both inference and parameter estimation as discussed in the previous sections.

4.7. Summary

In this chapter, the probabilistic modeling of spatial trajectory in road network is discussed under the framework of CRF, for which some highlights can be summarized as follows

1. A chain structured model is proposed in the context of map matching. The model allows an arbitrary number of transition paths which could be helpful for trajectory data with a low sampling rate..
2. With regard to the large number of states the model needs to evaluate, a state generation workflow as well as a redundancy elimination algorithm is proposed to reduce the computational complexity.
3. The issues of model inference and learning as well as the problem of missing values in the data are addressed.
4. ℓ_1 Regularization is proposed to learn a sparse model to achieve competitive labeling performance with much less model complexity.

CHAPTER 5.

Experiments and Implementations

In order to verify the feasibility and evaluate the performance of the proposed models, extensive experiments have been conducted. The experiments fall into two categories, namely localization and behavioral classification. In particular, map matching of GPS trajectories in road network and taxi status inferring, are tested on the real-world dataset.

This chapter first introduces the test data, spatial trajectories and road network used in the experiments. Then the implementations including feature extraction, probabilistic modeling, inference and learning, and labelled data preparation are discussed. Experimental results for both representative tasks are presented in the end as well as empirical studies of some most relevant research questions that arise in the practice.

5.1. Raw Datasets

Both map matching and taxi status inference are tested using the same real-world trajectory dataset, Shanghai taxi floating car data (FCD). The road network data in the corresponding area is extracted from OpenStreetMap[26] (OSM). The choices of test data are made for following reasons:

1. Shanghai, China, is one of the largest cities in the world and it has a highly developed and complex urban road network which may serve as an adequate test bed to demonstrate the power of the proposed models.
2. Taxi trajectory data enjoy a high spatial coverage of the urban road network and more consistent driving behavior (compared to normal drivers, taxi drivers are more experienced).
3. OSM road network for big cities is of relatively good qualities in terms of spatial coverage, completeness of attributes and it’s free to access.

The details of the test data are described in the remainder of this section.

5.1.1. Shanghai Taxi FCD

Shanghai Taxi FCD dataset stores the movements of ca. 7000 taxis in 82[27] days since 1 April 2010 to 30 June 2010, which use a sampling rate of 10s[28]. The raw data is provided in text files using the format of space separated values. Each text file stores all taxis’ GPS data

---

[26] www.openstreetmap.org

[27] Several days’ data are missing.

[28] A few records have shorter or longer sampling intervals.

within 24 hours that results 40 million records per day in average. The attributes of each record are listed in Table 5.1.

Name	Description	Example	Data Type[29]
date	date of GPS observation	2010-03-01	Date
time	time of GPS observation	20:37:16	time without timezone
company	abbreviation of taxi company's name	QS	character varying(4)
taxi id	number identifier of a taxi	18384	Integer
longitude	longitude in degree	121.531167	double precision
latitude	latitude in degree	31.22658	double precision
speed	instant speed in km/h	39.5	double precision
direction	instant heading direction of a taxi, range from 0 to 355, in which 0 indicate north and value increases clockwise.	245	Integer
occupied	binary code, with 1 for occupied and 0 otherwise	1	Integer
signal	binary code, with 1 for validate GPS record and 0 otherwise	1	Integer
server date time	date time that the record is saved in the server	2010-03-01 20:37:36	timestamp without time zone

Table 5.1 Attributes of Shanghai taxi FCD records.

To efficiently access such huge amount of the GPS data for query, analysis and mapping, the data are imported into the spatial database PostgreSQL[30] with extension PostGIS[31] that supports spatial queries. And the database management system (DBMS) is deployed on a Linux server configured with a SSD storage drive.

The FCD data often comes with errors, such as invalid attribute values (e.g., 25:34:16 for time, 100 for longitude), missing values, or even malformatted records. These records are removed from the data.

5.1.2. OSM Road Network

OpenStreeMap (OSM) is a free, editable map of the whole world that is built by volunteers. Due to its open nature and steadily improved data quality (Haklay, 2010), OSM has become increasingly popular for research and real-world applications.

OSM road network is extracted using the service Metro Extracts[32] which supports updated city-based extraction with multiple data formats, such as OSM PBF, shapefile, etc.

---

[29] Data type complies with the data types used in PostgreSQL

[30] http://www.postgresql.org

[31] http://postgis.net/

[32] https://mapzen.com/metro-extracts/

Then road data is converted into routable data format using OSM2PO33 thus to support routing analysis required in the labeling tasks. In the end, the routable road network is imported into the same DBMS. The attributes of the converted OSM road network is listed in Table 5.2.

Name	Description	Example	Data Type[34]
osm_id	identifier of the road	8621489	bigint
osm_name	name of the road	居家桥路	char varying
osm_source_id	identifier of the source node of the road	115443169	bigint
osm_target_id	identifier of the target node of the road	115443115	bigint
class	class code of the road according to OSM's 'highway' tag[35]. In general, a smaller number indicates roads with high speed limits such as highway.	32	int
length	Road length in km	0.2868539	double precision
kmh	speed limit in km/h	50	int
cost	travel time from source node to target node computed by length/kmh	0.0057370784	double precision
reverse_cost	travel time from target node to source node computed by length/kmh. For two-way road, reverse_cost is equal to cost, while it's set to 1000000 for one-way road.	0057370784	double precision
x1	longitude of source node	121.5555711	double precision
y1	latitude of source node	31.263319	double precision
x2	longitude of target node	121.557088	double precision
y2	latitude of target node	31.2610903	double precision
geom_way	polyline geometry of the road		geometry

Table 5.2 Attributes of routable OSM road network data.

OSM road network for Shanghai contains in total 10927136 roads and 77895 nodes, which is illustrated in Figure 5.1.

---

[33] http://osm2po.de/

[34] Data type complies with the data types used in PostgreSQL

[35] http://wiki.openstreetmap.org/wiki/Key:highway

[36] Statistics based on the data retrieved in Dec, 2014.

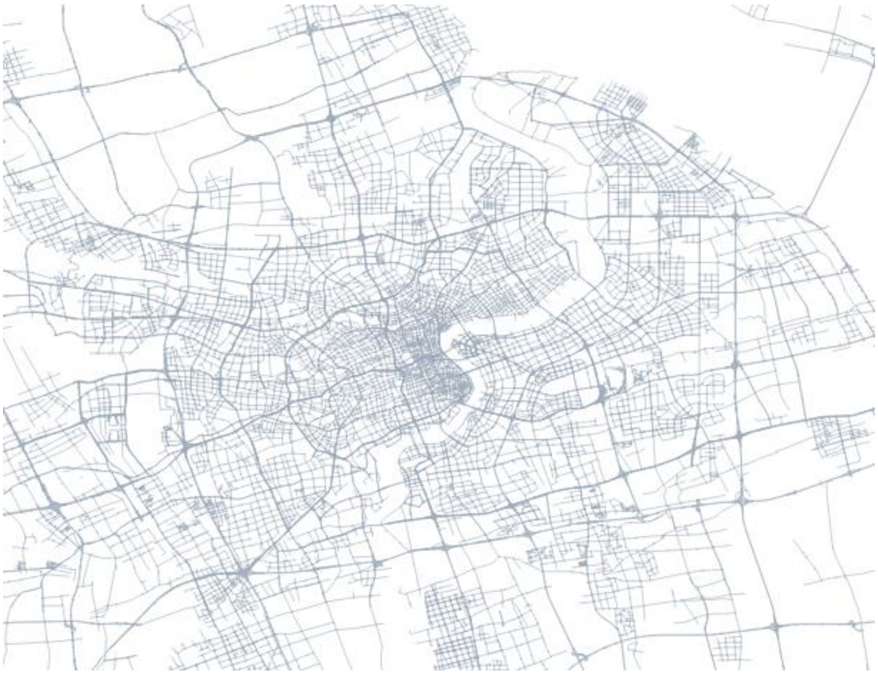


Figure 5.1 Shanghai OSM road network at scale 1:203,669.

5.2. Implementations

The implementations for the experiments involve three major tasks, namely labeling, feature extraction and model development (training and testing). Labeling is to identify the labels for each data instance, e.g. find the road sequences given location sequences. In the setting of supervised learning, labeled data is required for both training and testing. Feature extraction is to derive a numerical representation from the (labeled) dataset. And model development deals with the implementation of the proposed model.

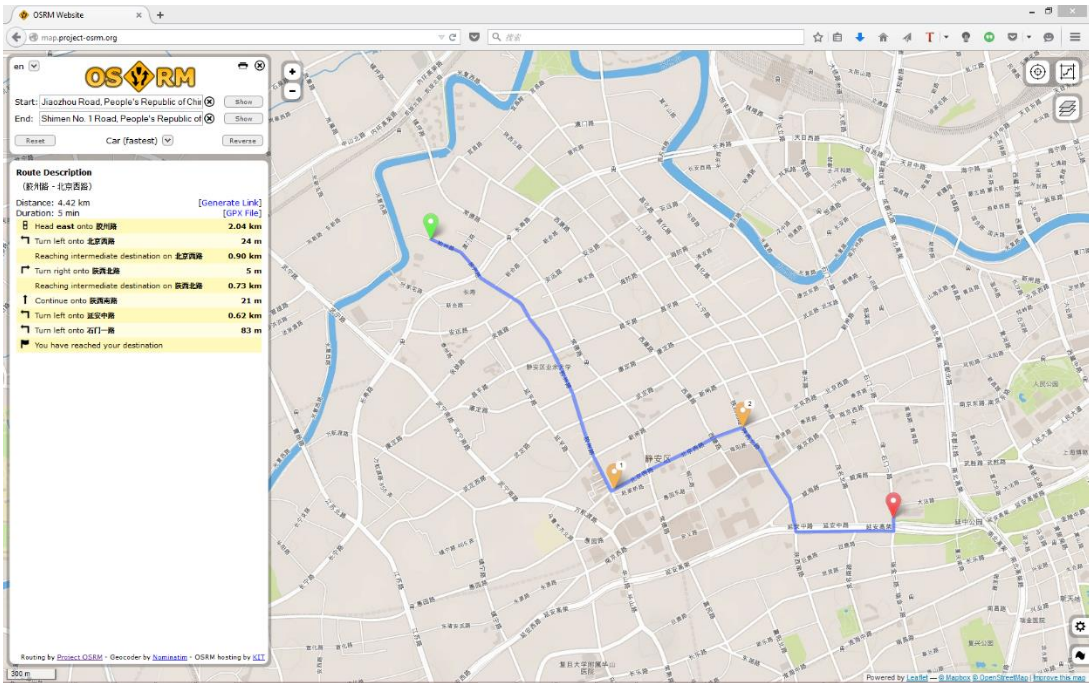
5.2.1. Labelling Using An Interactive Routing Tool

Labeling is a common task in developing learning machines, since labeled data is required in the experiment for performance test of either supervised or unsupervised learning. In our task, map matching doesn’t have the ground truth data and thus we have to manually prepare the labeled data. Labeling for map matching may refer to various efforts depending on the specific goals in the applications, e.g. finding corresponding road sequences, finding the actual positions on the roads, which requires different definition of the labels. Since we are more interested in the routing behaviors, we do the labeling only at the road-level.

Labeling for map matching is a non-trivial task because the manual solution without appropriate concern would yield erroneous labels and consume huge amount of time. In our first trial, a bare-hand solution is used that we do the labeling in ArcMap by manually checking one GPS point at a time, which takes 20 hours to match 1400 GPS points. Furthermore, the matched data contain many errors due to reasons such as skipping short road segments, missing GPS points.

Fortunately, there exists a practical way to overcome the deficiency of the bare-hand solution by taking advantage of an interactive routing tool. More specifically, we use Open

Source Routing Machine (OSRM)[37] to perform this task. Though OSRM is originally designed to find optimal (e.g. shortest/fastest) path in road networks, its interactive routing adjustment interface makes it eligible for the labeling tasks (many routes don't follow shortest/fastest path). Figure 5.2 shows the mapping of the results (i.e., road sequence for the given GPS trajectory) as well as the detailed turn-by-turn routing instructions display on the left. Having set the origin (green balloon) and the destination (red balloon), OSRM computes the optimal path automatically, and then we can alter the route by simply drag the route (blue line between the origin and destination) to match the given GPS trajectory. And each adjustment would be marked with a yellow balloon. As shown in the figure, two adjustments are made to recover the original route. Further post processing is needed to elaborate the label information in a machine-readable form[38].


OSRM Website
map.project-osrm.org
en
OSRM
Show
Show
Reset
Reverse
[Generate Link]
[GPX File]
1
2
静安区
300 m
Routing by Project OSRM - Geocoder by Nominatim - OSRM hosting by KIT
Powered by Leaflet — © Mapbox © OpenStreetMap | Improve this map

Start: Jiaozhou Road, People's Republic of Chi

End: Shimen No. 1 Road, People's Republic of

Car (fastest)

Route Description

(胶州路 - 北京西路)

Distance: 4.42 km

Duration: 5 min

Head **east** onto 胶州路	2.04 km
Turn left onto 北京西路	24 m
Reaching intermediate destination on 北京西路	0.90 km
Turn right onto 陕西北路	5 m
Reaching intermediate destination on 陕西北路	0.73 km
Continue onto 陕西南路	21 m
Turn left onto 延安中路	0.62 km
Turn left onto 石门一路	83 m
You have reached your destination	

Figure 5.2 Web-based interactive routing tool, OSRM, for labelling GPS trajectories for map matching.

This interactive routing tool manages to assist us in labeling 14000 GPS points in around 2 hours (100x boost in time efficiency) and also help us to gain the insight of a variety of routing cases that detour from the optimal paths. However, there still exists difficulty to ensure the labels' quality which needs manual fixes, e.g., routes that cannot be found using optimal path finding, routes that contain a U turn which cannot be found by the OSRM (even with the adjustment), erroneous road data (i.e. missing end roads, outdated road topology) in the OSM road network.

---

[37] http://project-osrm.org/

[38] Post processing is done using a python script that to retrieve the road sequence from the road network using the sampled location sequence exported by the OSRM. Since the sampling is done using the road network, it eligible to recover the road sequence by finding nearest neighbor.

5.2.2. Feature Extraction in Spatial Database

Feature extraction from GPS trajectories in road network involves non-trivial spatial analysis such as spatial queries, route planning, etc. Though a variety of spatial analysis tools can be used to perform these tasks, we choose spatial databases as the feature extraction platform for following reasons:

1. Full functionality. Spatial database such as PostgreSQL is well equipped with a wide range of spatial analysis tools to support spatial analysis tasks, e.g., PostGIS for spatial queries, pgRouting for route planning, etc.
2. Interoperability. Spatial database enjoys the flexibility to export geospatial data into a variety of formats for further use such as mapping and plotting, which are critical means to explore large datasets.
3. Flexibility. Spatial database embraces a natural language alike query interface, SQL. This is particularly helpful in the iterative development of features for the predictive model, which eases the engineering efforts in the debug-and-test cycle.

Taking advantage of the abovementioned properties, we develop a feature extraction module on top of PostgreSQL using PL/pgSQL, and thus we may generate feature data from raw trajectory data in database by performing SQL queries. Take the labeling task of map matching for example, the data model for feature extraction is illustrated using an Entity Relationship Diagram (ERD) with Craw's foot notation[39] in Figure 5.3. Note that, for taxi status inference, the table *Graph_State* only contains two binary records and depends no more on the table *Road*.

---

[39] https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

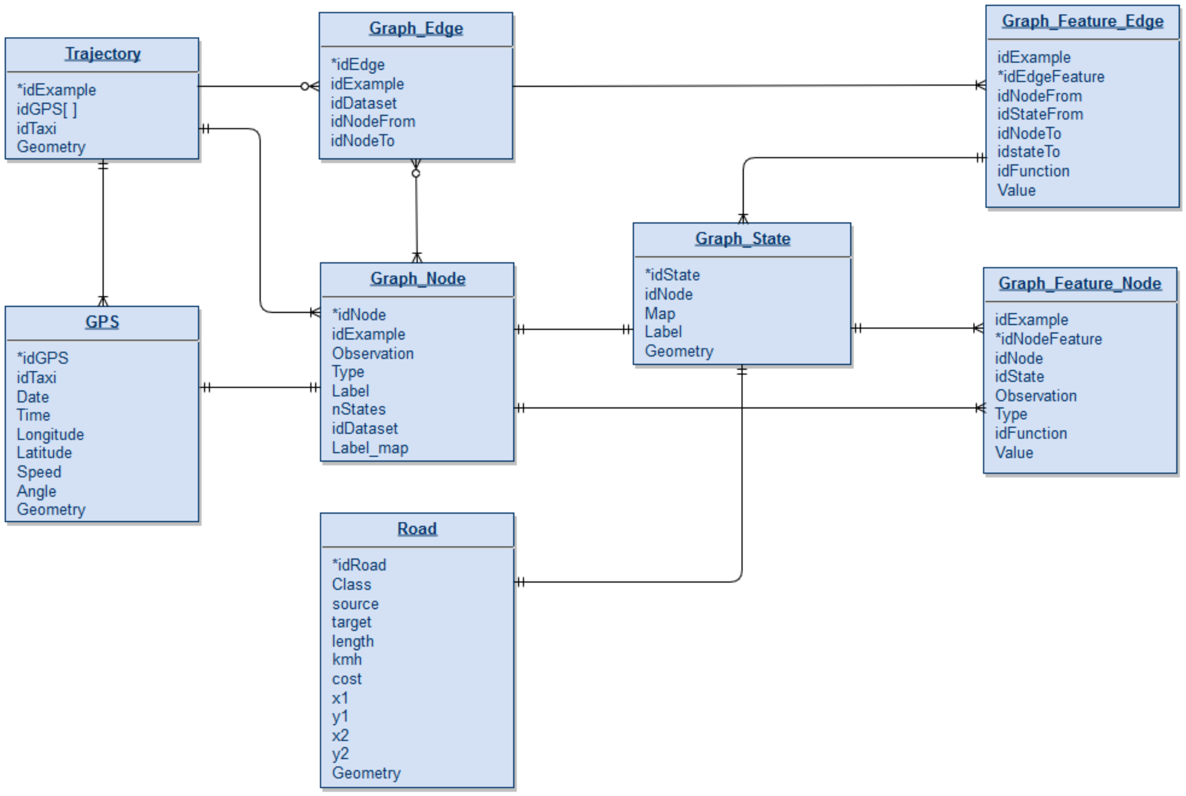


Figure 5.3 Entity Relationship Diagram of the database design for feature extraction.

5.2.3. Path: A Matlab Toolbox For Labeling Spatial Trajectories

As for the model development, we implement the proposed model using Matlab for its fast scripting capability for prototyping and the rich libraries for graphical models, e.g., Bayes Net Toolbox[40] (BNT) for directed graphical model, UGM[41] for undirected graphical model. More specifically, our implementation is built upon on Mark Schmidt's UGM, which is a set of well documented Matlab functions that implements undirected graphical of discrete data with pairwise potentials, i.e., decoding, inference, sampling and training.

UGM follows a similar principle of code design as Kevin Murphy did in BNT, a simple data structure for representing the graphical structure of the model using an adjacency matrix, all inference and optimization methods are designed to share the function signature so that they may work in a plug-and-play fashion. Though UGM is capable of building model with arbitrary graphical topology, it doesn't support high-order potentials (potentials with 3 or more than 3 random variables as the input) which makes it less favorable.

Thanks to UGM's scalable architecture, our implementation only needs to focus on the model building (i.e. initiating the graphical structure using adjacency matrix, parameters binding). Furthermore, UGM suffers a poor performance in terms of runtime efficiency, which becomes worse for large dataset since the inference sweeps the whole dataset for each objective evaluation in the optimization. In order to take the fully advantage of the multicore

---

[40] https://github.com/bayesnet/bnt

[41] http://www.cs.ubc.ca/~schmidtm/Software/UGM.html

computing platform, we adopted the parallelism in the data sweep code (i.e. for loop) using Matlab's specialized parfor loop implementation.

5.3. Labeling Task I – Map Matching of Low-Sampling Rate GPS Trajectories

We first evaluate our model using the task map matching of low-sampling rate GPS trajectories. The experiments involve data preparation, model training, testing and evaluation. In the following sections, we first introduce the experiment setup, then present the experimental results, and assess the results using specific sample cases.

5.3.1. Experiment Setup

The test dataset for map matching is extracted from the Shanghai Taxi FCD dataset, records GPS trajectories of 70 taxis in one day across the downtown area in Shanghai, China. It comprises 124 trajectories in total and 13767 GPS observations covering an overall length of 788 km after eliminating some erroneous trajectories, e.g. extremely short trips and trips losing long distance GPS observations. Spatial distribution of the trajectories in the test dataset and statistics of sample trajectories are demonstrated in Figure 5.4. The rationale of using a geographic constraint of passing through downtown area is to collect the most representative (complex routing scenarios) and challenging (dense road network) map matching cases in Shanghai while retain a relative small size of test data.

Since the data source doesn't provide the ground truth labels, we prepare the labeled data as discussed in Section 5.2.1. With the labeled high-sampling rate data (i.e. 10s sampling interval), we degrade the test data (10s sampling interval) to 120s sampling interval using an even sampling strategy, and thus yielding total labeled 1458 GPS observations. Furthermore, we randomly split it into a training set and a test set with a ratio of 7:3 (see Table 5.1 for details), the entire training set is used to estimate the hyper-parameters via 5-fold cross validation. These settings are applied to both ℓ_2 and ℓ_1 regularization.

	#Trajectory	#GPS observations	#Paths
Training set	87	1099	1009
Test set	38	479	436

Table 5.3 Specification of training/test set.

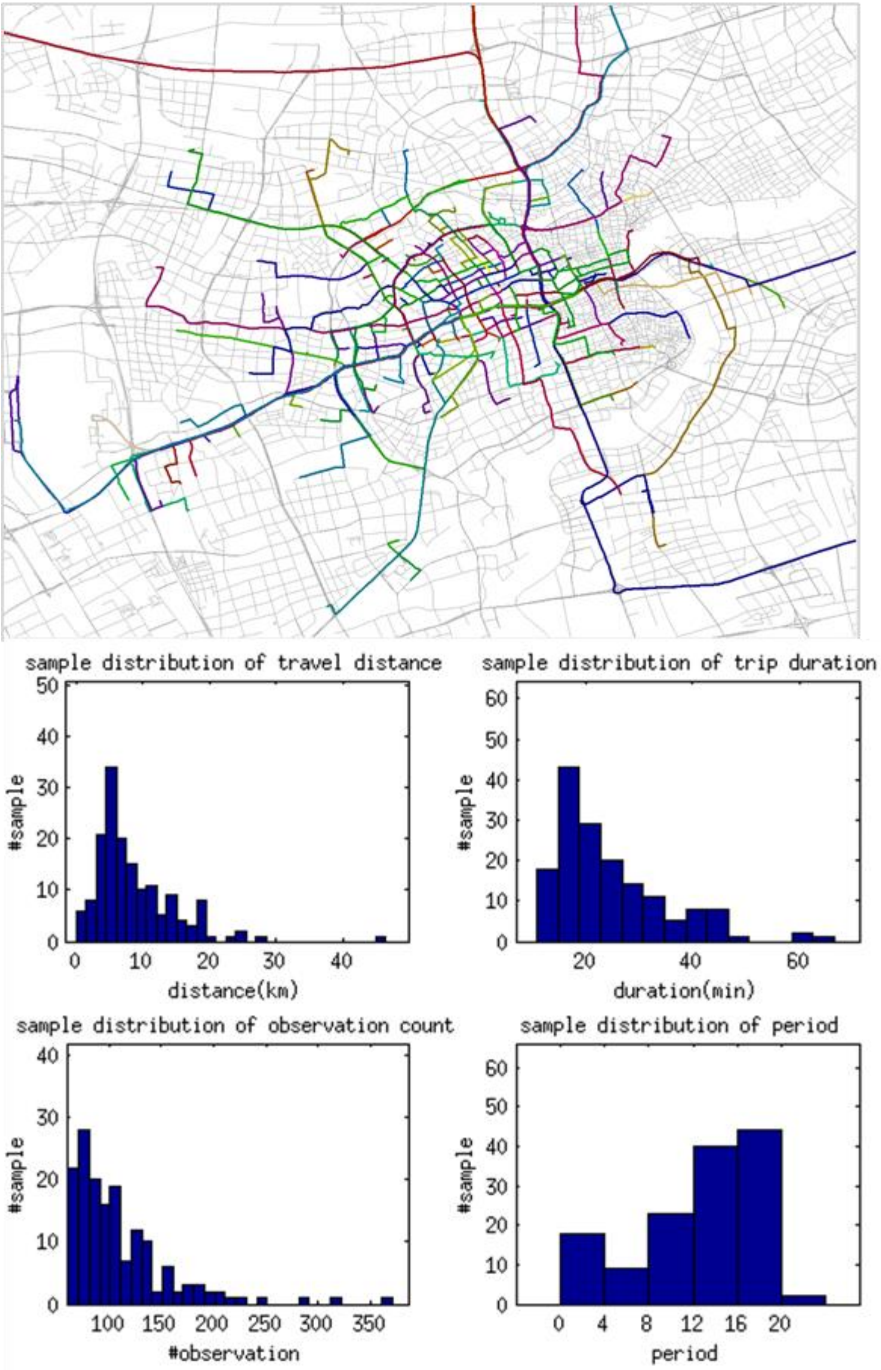


Figure 5.4 The spatial distribution of GPS trajectories in the test data for map matching (Top). The statistics of sample trajectories (Bottom): travel distance (upper left), trip duration (upper right), observation count (bottom left) and daytime period in hour (bottom right).

Having extracted the feature data as discussed in Section 5.2.2, preprocessing such as rescaling and standardization are compared with their application to the entire dataset before feeding them to our model. This step is crucial for both identifying the most relevant features and gaining a regular objective function surface for the optimization so as to avoid trapping the optimizer in certain dimension with extremely large weight. In the test, we find that using standardization gains a slightly improvement on both accuracy performance and runtime efficiency.

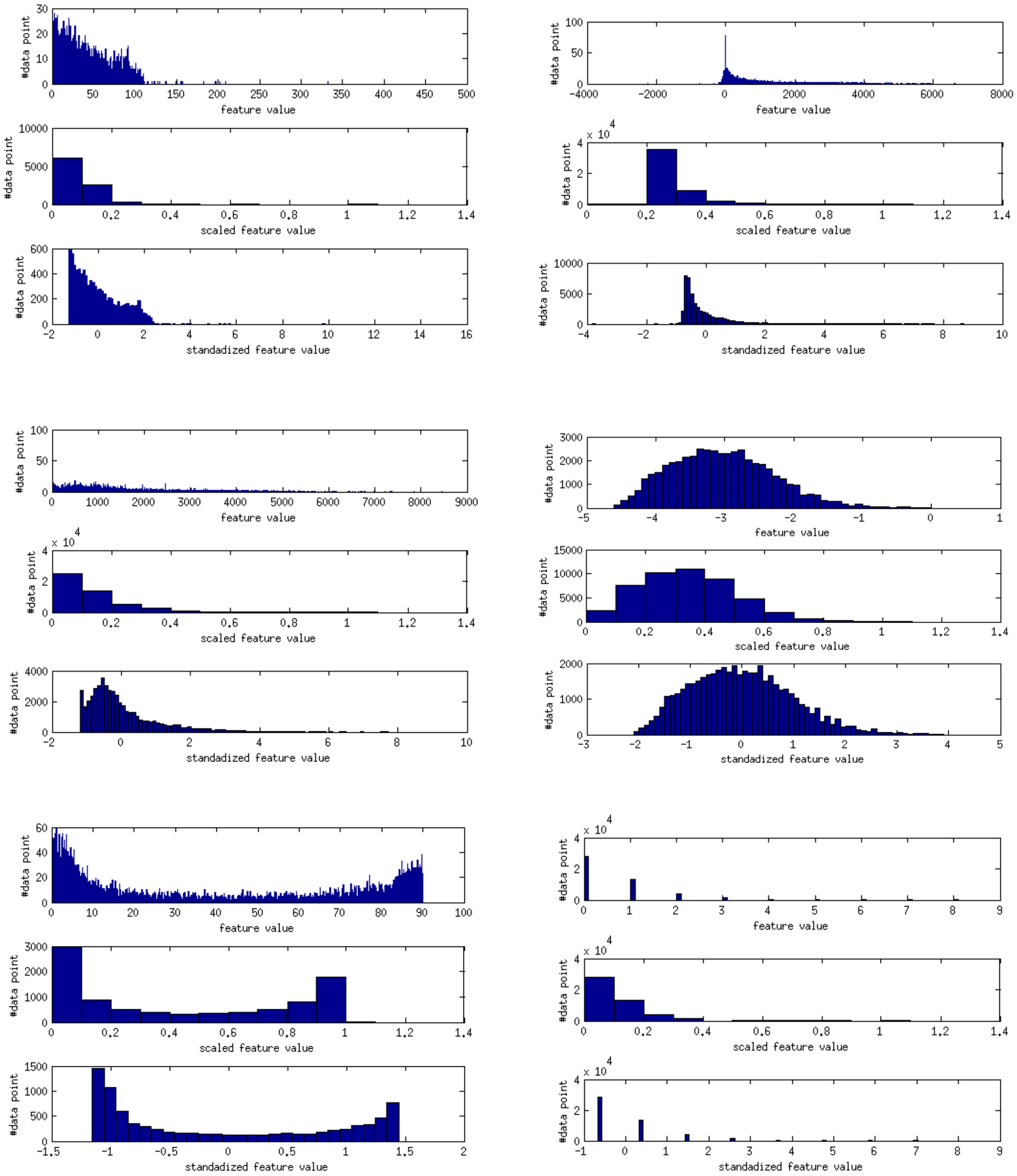


Figure 5.5 Distribution of the sample feature data. In each block, (up) is raw feature, (middle) is re-scaled feature, (bottom) is standardized feature.

5.3.2. Experiment Results

We use both *Error Rate* and *Overall Confidence* to evaluate the matching accuracy of the model on the given test dataset, which are defined as follows.

- Error Rate quantifies the matching accuracy in terms of the proportion of the count of incorrect matching instances among all instances with a range of [0, 1], the smaller the better, which can be computed as

$$\mathrm{ErrRate} = \frac{\#\ \text{Incorrect matching instances}}{\#\ \text{Matching instances}}$$

In the evaluation, we compare the error rate for point, path, and total (including both point and path) separately with the purpose of discriminating the model capability in matching points and inferring paths. Though these two subtasks are interrelated, it is still tempting to compare the error rates individually because inferring paths are considered more difficult than matching points.

- Overall Confidence is designated to reveal the capability that the model may discriminate truth candidates among false ones. Each matching instance (i.e. point or path) is often assigned with a confidence with a range of [0, 1], where higher confidence indicates better performance even when the two models achieve the same error rates. As for the entire dataset, we endorse the definition in ACM GIS cup 2012(Ali et al., 2012) but with focus solely on the confidence which yields

$$\text{OverallConf} = \frac{\sum \text{Correct Matching's Conf.} - \sum \text{Incorrect Matching's Conf.}}{\#\ \text{Matching Instances}}$$

  The metric takes the values in [-1, +1], where -1 means all label assignments are incorrectly assigned and 1 means all label assignments are correctly assigned.

A 5-fold cross validation is used to estimate the optimal hyper-parameter λ for the model with ℓ_1 regularization. The candidate parameter is geometric sequence computed as $\lambda = \text{base}^{\text{power}} \times \lambda_{\text{max}}$ (see Section 4.6 for details). In the experiment, we use the base of 0.3 and change the power term from 0 to 10 with unit increase to generate the candidate parameters. Figure 5.6 shows the varying accuracy of the model in terms of total error rate (top) and overall confidence (bottom) as the power term increases. The error rate drops radically as the power term increases and increase gradually after a tipping point. The trend of overall confidence matches well with the error rates and yields the optimal performance with the same setting. And we pick the hyper-parameter using the one standard error rule (i.e. pick the λ that is one standard error higher than the optimal value but with less model complexity) that the model reach the peak performance and remain stable as the power term increases to 5, which is marked with the green vertical line. Note that we use the same hyper-parameter for ℓ_2 regularization.

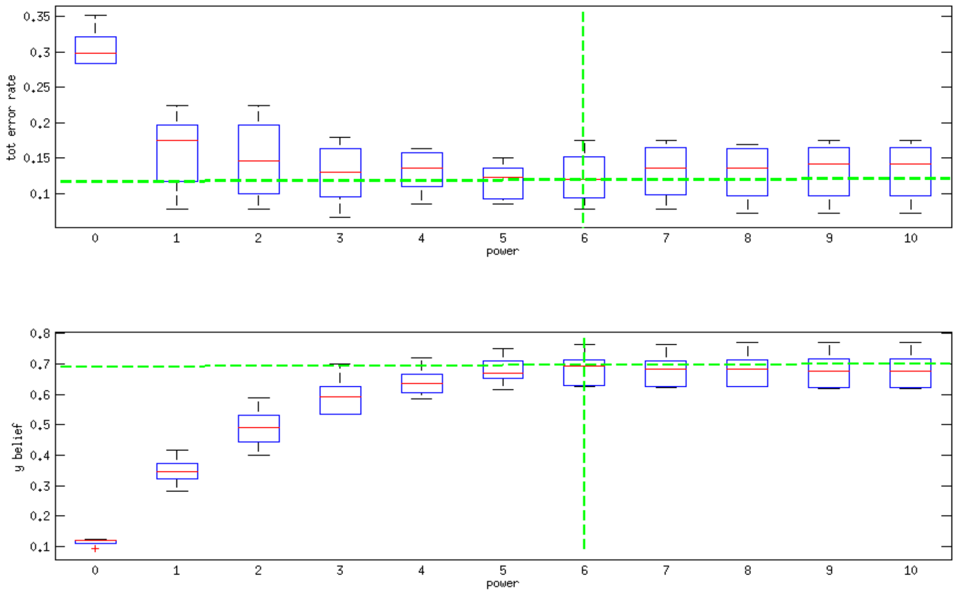


Figure 5.6 5-fold cross-validation estimates for optimal hyper-parameter λ.

Methods	**#Feature**	**ErrRate**			**OverallConf**
		Point Match	Path Discovery	Total	
Hunter_c	6	.139	.204	.170	.521
Hunter_s	2	.176	.289	.230	.403
CRF_L2	21	.131	.186	.157	.598
CRF_L1	19	.129	.177	.152	.640

Table 5.4 Evaluations of map matching results on GPS trajectories of 120s sampling interval.

We evaluate the matching accuracies of our model against baseline methods using the aforementioned two metrics. More specifically, the error rate is compared on finer perspectives, that is, the results are evaluated in three categories of point match, path discovery, and total error rate which treats the point and path equally. And the number of features used in the model is represented to show the complexity of the model, the bigger the count is, the higher the complexity the model contains. The results are summarized in Table 5.4. *Hunter_c* and *Hunter_s* are the baseline methods which also developed based on CRF and follow a learning procedure for parameter (Hunter et al., 2013). *CRF_L2* is our ℓ_2 regularized CRF trained with BFGS (Sha et al., 2003), and *CRF_L1* is ℓ_1 regularized CRF trained with Projected Scaled Sub-Gradient (PSS) methods (Schmidt, 2010). The major differences compared to the baselines is that our models are built on a different formulation of the graphical model and they incorporate a comprehensive feature set and also employ feature selection in the training phase using ℓ_1 regularization.

Several intriguing points can be made from the results. First, path discovery is more challenging than point match to all methods in the test. And using a much richer set of features for path discovery has led to more success (8.5% lower in error rate) for *Hunter_c*, and this confirms the behavior in the original paper. Secondly, both of our models can outperform baselines in each category of evaluation with recognizable margins. The one with the best performance, *CRF_L1*, yields boosts in the metric of overall confidence that it outperforms *Hunter_s* with a margin of 24% and *Hunter_c* with a margin of 11.9%. The results show that incorporating more relevant features (by applying feature selection) can lead to better fit of the data and improved accuracy of the labeling task. Furthermore, it's a surprise that *CRF_L1* only outperforms and *CRF_L2* on overall confidence without noticeable improvements on the total error rate.

The weights of the features learned from the training set in our model is illustrated in Figure 5.7. The weights' magnitude indicate the relevance degree of the feature to the map matching task. Among all the features, distance error (DistErr), the number of left turns (#LeftTurn), the number of links in the path (#Lnk) and the number of different road classes in the path (#RoadClass) are the most relevant ones.

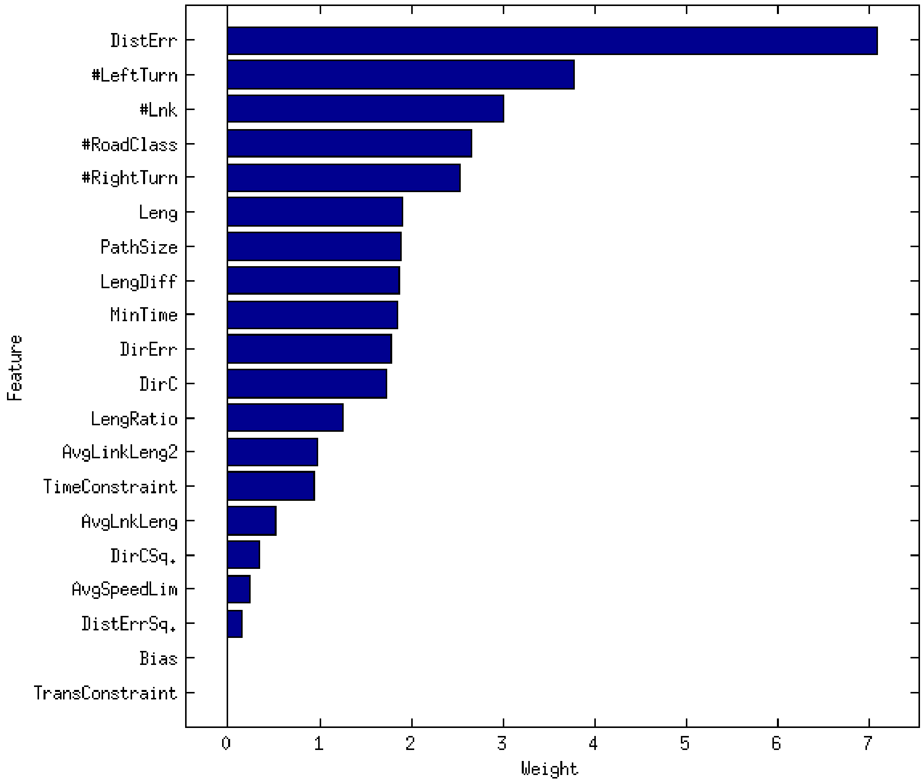


Figure 5.7 Learned weights in ℓ_1 regularized CRF for map matching.

5.3.3. Case Study

In order to investigate the effectiveness of the proposed model (CRF_L1) for map matching beyond the plain numbers, the matching results in both model training and testing are mapped on the road network in comparison with the true labels. As illustrated in Figure 5.8, all recovered routes are overlaid by the ground truth, i.e., manually labeled routes using high sampling rate position data. With visual inspections, the mapping shows that the proposed model managed to recover the routes with meaningful paths (without paths composed by irregular geometric shapes). However, there also exists (red) paths that deviate from the ground truths in several local road network contexts, and these mismatched instances (including both points and paths) are categorized based on their likely causes. The major error cases are missing label (18.3%), parallel roads (13.7%), U-turn (13.0%), starting/ending point (10.0%), and position outlier (9.9%). Missing label occurs when observations locate in the dense road networks and true states are unexpectedly eliminated due to the predefined count of states in implementation. Parallel road (see Figure 5.9) and U-turn (see Figure 5.10) happen when the model fits the observations well but makes no sense compared to real-world driving experience. Starting/ending point (see Figure 5.11) can be eliminated by combining contextual information, e.g., it's more likely to start a trip in the roads close to the building areas rather than in the middle of express roads.

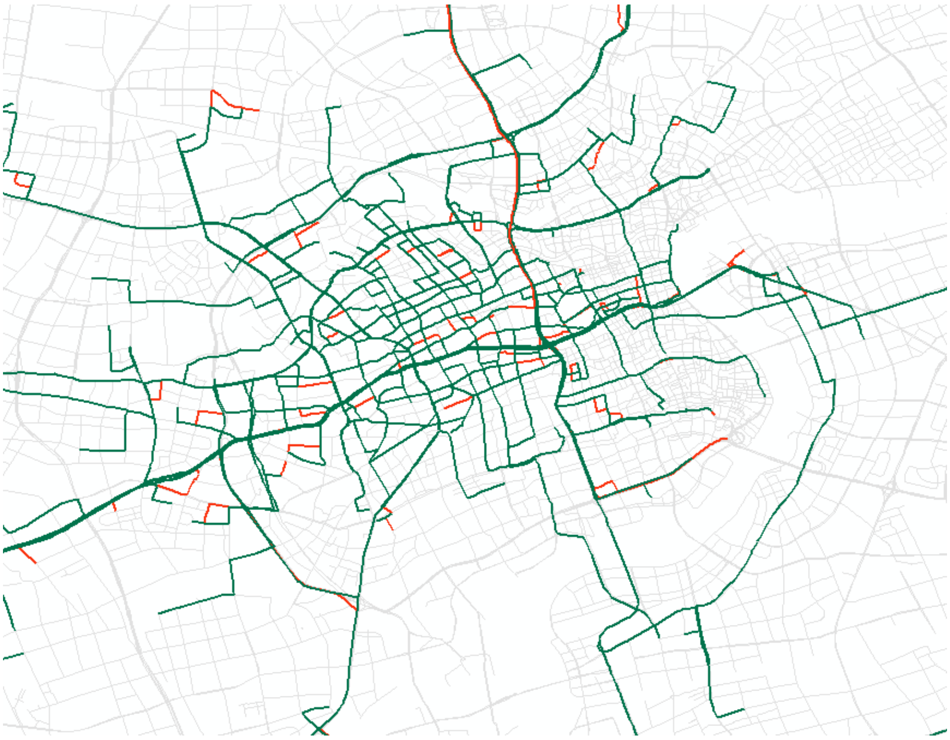


Figure 5.8 Map matching results of CRF_L1 (red) overlaid by ground truth (green) on the road networks (grey).

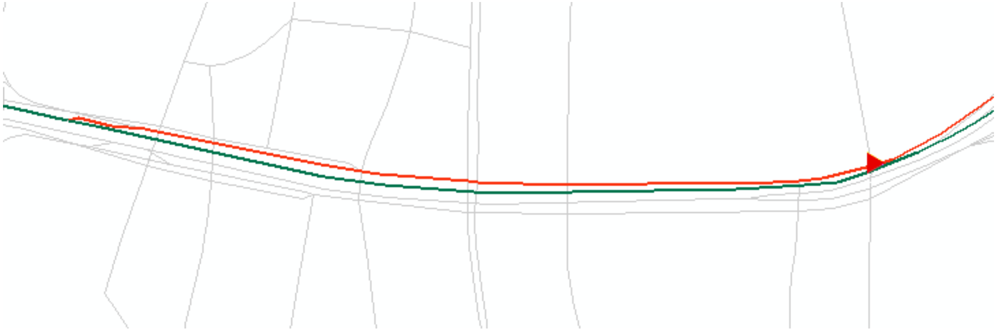


Figure 5.9 Error instance: parallel road. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green.

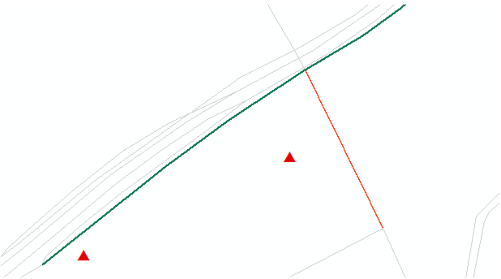


Figure 5.10 Error instance: U-turn. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green.

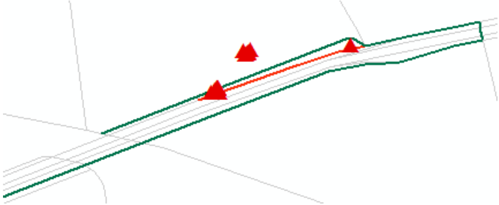


Figure 5.11 Error instance: Starting/ending points. GPS points are marked as red triangles, recovered path is marked in red and the ground truth is marked in green.

5.4. Labeling Task II – Inferring Taxi Status

For the second labeling task, inferring taxi status, the empirical evaluations are conducted using a new test dataset derived from Shanghai FCD. The dataset, different from the one used in the map matching task, obeys no restriction of geographic extent but allows full mobility in the urban area, which helps to collect a variety of taxi trips. For example, long taxi trips from downtown hotel to suburban airport, medium long trip for commuting between business building and passenger's apartment. Following a similar organization of the experiments of map matching, data preparation and preprocessing, experiment settings, model training and testing, evaluation are discussed.

5.4.1. Experiment Setup

The test dataset for taxi status inference targets 50 taxis, which consists of the GPS trajectories from their one-day activities in Shanghai. To make a more faithful evaluation on the error-prone raw floating car data, a data preprocessing procedure is designed to build the test dataset.

The preprocessing procedure includes three steps: 1) Taxi trip extraction from FCD using the identifier of the taxis and given taxi status. Each trip bears one consistent taxi status as indicated in the raw data, namely the trip is either occupied or vacant. The end points of trip are the pick-up/drop-off points. 2) Trip-based validation to eliminate erroneous trips or correct them. For examples, remove starting trips of the day that are recorded as occupied while the taxi is actually idle. And if temporal gaps are found within the trips, the trips are broken into sub-trips. The gaps may indicate the fact that the taxi drivers are taking breaks or making work shifts during the day. 3) Successive trips (trips that have temporal separation of no more than 3mins) are merged into one trajectory. Thus, it may be used to test the status transition within the same trajectory. Note that the resulting trajectories may contain a variety number of trips (with minimum number of one) with varying length.

Then the preprocessing procedure yields a test dataset that consists of 480 sample trajectories. These samples are built of 1999 trips, 23171 GPS observations, and with approximately equal number of occupied trips and the vacant ones. The spatial distribution of the trajectories is shown in Figure 5.12.

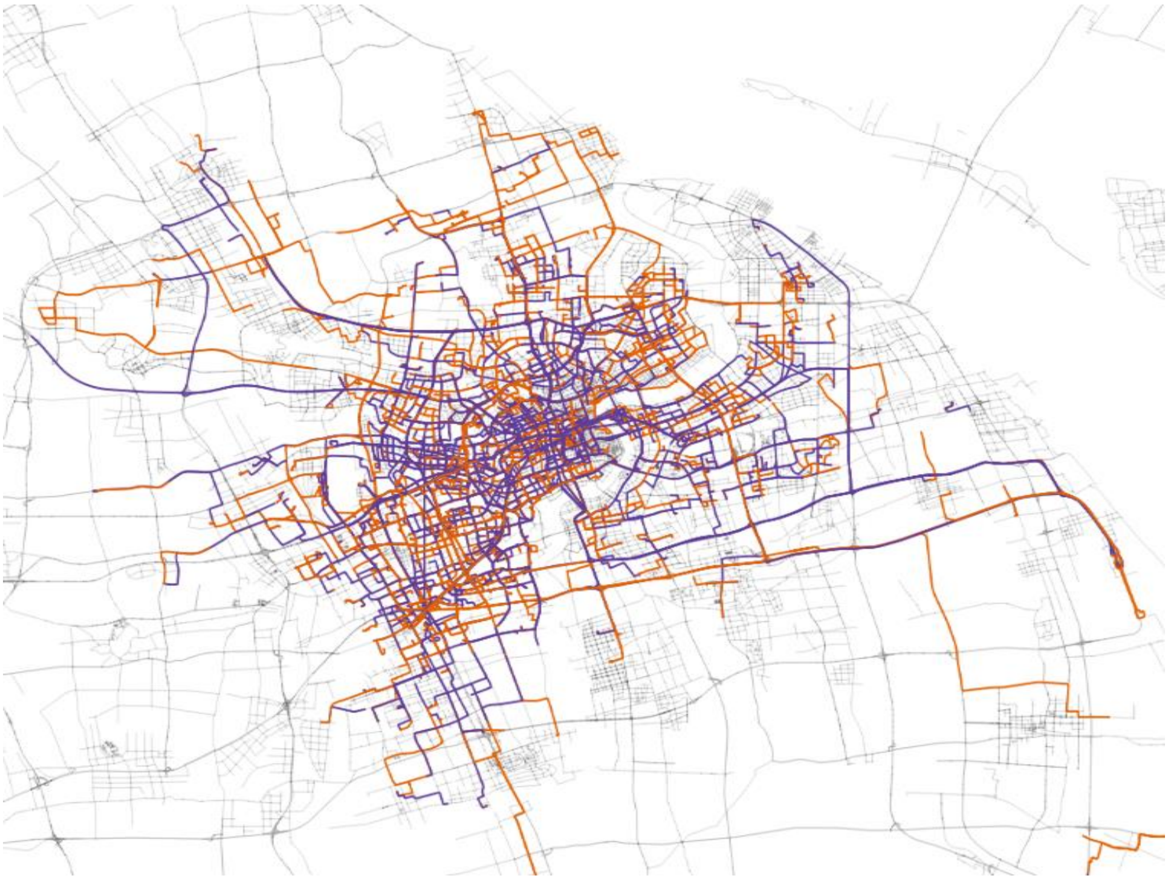


Figure 5.12 50 taxis' one-day trips in Shanghai, China. Blue lines represent occupied taxi trips and red lines are vacant.

Since the raw data have already the taxi status (see Table 5.1), we randomly split the data into a training set and a test set with a ratio of 7:3 (see Table 5.5 for details), similar to the experiment for map matching. The entire training set is used to estimate the hyper-parameters via 5-fold cross validation. These settings are applied to both ℓ_2 and ℓ_1 regularization.

	#Trajectory	#GPS	#Trip
Training set	336	16220	1399
Test set	144	6951	600

Table 5.5 Specification of training/test set for taxi status inference.

5.4.2. Experiment Results

We use *Precision* and *Recall* (Friedman, Hastie, & Tibshirani, 2009) to evaluate the proposed model on this sequential prediction problem using the aforementioned test dataset.

Different to the metric error rate or overall confidence, precision and recall are computed on a class/label basis. Since for the second labeling tasks, the labels bear a more consistent meaning, and thus new metrics are used to distinguish the model's performance on individual label. The two metrics are defined as follows.

- Precision quantifies the labeling accuracy in terms of the ratio of the count of correctly labeled instances among all labeled instances with a range of [0, 1], the larger the better, which can be computed as

$$\text{Precision} = \frac{\#\ \text{correct labels of i-th class}}{\#\ \text{all labels of i-th class}}$$

- Recall quantifies the labeling accuracy in terms of the proportion of the count of correct labeled instances among all labeled instances of the same kind with a range of [0, 1], the larger the better, which can be computed as

$$\text{Recall} = \frac{\#\ \text{correct labels of i-th class}}{\#\ \text{instance of i-th class}}$$

In order to estimate the optimal hyper-parameter λ, a 5-fold cross validation with a parameter search using geometric sequence (base of 0.3, power ranges from 0 to 10) is applied on the training set. Figure 5.13 shows how the precision and recall change for both statuses while manipulating the hype-parameter (top two for *Occupied* and bottom two for *Vacant*). For the status of occupied, increasing the power term (decreasing the hyper-parameter) yields increasing precision and recall simultaneously. Both metrics reach their best results as power term is set to 4 and decrease slightly afterwards. However, the outcomes are different for the status of vacant. Increasing the power term manages to obtain better precision but only in the exchange of the performance in recall. The best result for precision is achieved with almost the smallest hyper-parameter trial, while the largest hyper-parameter leads to the highest recall. Then for the overall hyper-parameter selection, a tradeoff has to be made, that is the better prediction for occupied can be yielded at the cost of worse prediction for the other. In this experiment, we are more interested in identifying the occupied trips and thus the hyper-parameter is set with the preference of higher precision and recall for this status. With the one standard error rule, the power term is set to 3, which applies to both ℓ_2 and ℓ_1 regularization.

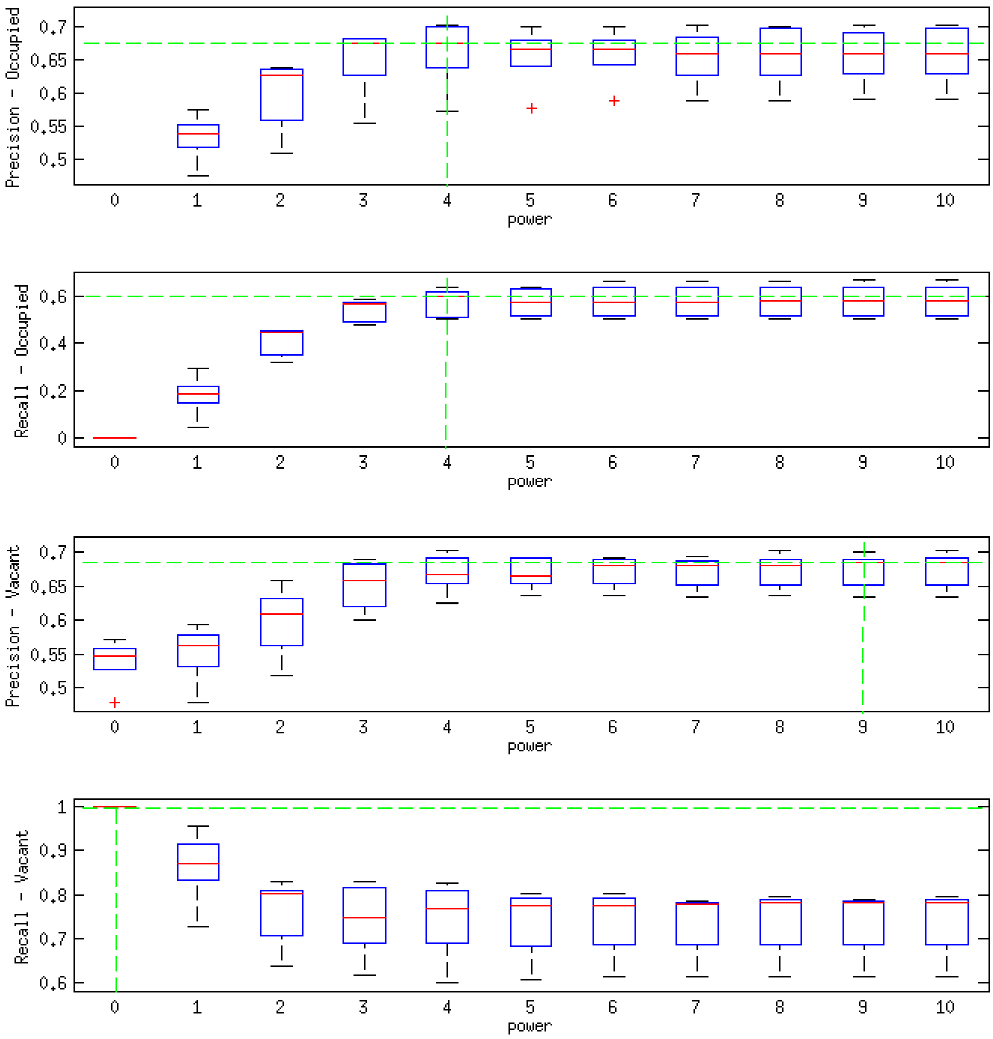


Figure 5.13 5-fold cross-validation estimates for optimal hyper-parameter λ. Horizontal green dash lines mark the highest performance, and the vertical green dash lines indicate the power used to compute the hyper-parameter.

Methods	**#Feature**	**Precision**	**Recall**
CRF_L2	28	.647	.461
CRF_L1	9	.649	.548

Table 5.6 Evaluations of taxi status inference of occupied taxi GPS.

The results of our model on the test set are summarized in the Table 5.6. Two methods, *CRF_L2* and *CRF_L1*, are evaluated, which shows that the both models manage to achieve at moderate precision and recall for identifying the occupied data. In particular, ℓ_1 regularized CRF model yields a slightly better result with much less model complexity (68% features are eliminated in the training). Surprisingly, using more features doesn't appear to improve the prediction compared to the one-feature model proposed in (Ganti et al., 2013), which suggests using HMM with a stretch factor operated on a window basis.

Furthermore, the confusion matrix of *CRF_L1* is also given to demonstrate the label-dependent performance of the model. Even though the hyper-parameter is set with a preference of the status of occupied, the model performs better for labeling vacant data.

	Occupied	**Vacant**		
Occupied	1443	1572	.479	Recall
Vacant	818	2790	.773	
	.638	.640	Accuracy: .639	
	Precision			

Table 5.7 Confusion matrix for CRF_L1.

The weights of the selected features in ℓ_1 regularization are plotted in Figure 5.14. All features that retain a non-zero weight after training are plotted in the order of their magnitudes. The code name of the feature indicates the window size on which it is applied, e.g., SpeedMean-W3-1 indicates that mean speed applied window of three successive observations for status occupied (-2 is for vacant). As it’s shown in the figure, all the selected features are node features (including the two bias terms). And SpeedVarW31, SpeedMeanW72, SpeedMeanW151, SpeedMeanW152 have the largest weights that are learned from the test dataset. Surprisingly none of the transition/edge features are retained in the feature selection.

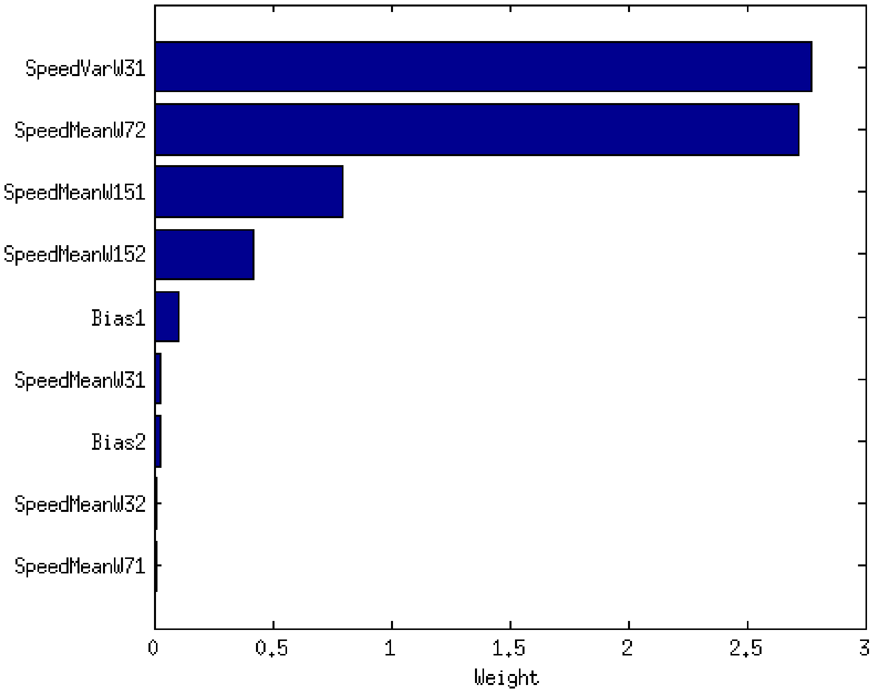


Figure 5.14 Learned weights in ℓ_1 regularized CRF for taxi status inference.

5.5. Discussions

In this section, some discussions are brought forth for the issues arise in the experiments of labeling spatial trajectories in the road network, which attempt to clarify the use of the proposed method for further applications.

The empirical study of the two labeling tasks has shown that the proposed methods are feasible to solve the problem. In map matching, the CRF_L1 outperforms the baselines with a small margin, but it generates a significantly high confident predictions (ca.12%) which demonstrates its capability of tackling uncertainty. As for taxi status inference, the CRF_L1 achieves similar results while consuming more features than baseline. Both labeling tasks have shown the effectiveness of applying feature selection technique, ℓ_1 regularization, in the model training leads to a huge reduction of the model complexity and improvements of the performance.

Both of the labeling tasks can be formulated using the same modeling framework, and benefit from the common efforts in the model development in terms of designing the graphical structure, applying inference and learning algorithm, etc. However, some distinct efforts that help to improve the overall performance can be highlighted as follows.

- In the early experiments of map matching, missing label (i.e., true labels are not covered in the candidates) is the major error source. This is partly because the original models have fixed the number (usually small) of candidate paths. Therefore, a special focus is set on the design of the graphical structure. Using a specific path node in the graph allows to consider an arbitrary number of labels, which helps to avoid early elimination. Since taxi status inference only deals with binary labels, the label size requires no special treatment.

- As shown in the feature selection results for taxi status inference (see Figure 5.14), features that are applied with window function dominates in the feature set. Using window function helps to examine a wider range of mobility characteristics of taxi movement, which compensates the limitation of the chain structure.

- The feature selection has shown different effects on the two labeling tasks. More features are eliminated in the taxi status inference, while ℓ_1 regularization improves the likelihood which leads to higher probability output in the final label assignments.

Besides the model development, there also exist some practical concerns in the implementation. First, both tasks could suffer from poor label quality, e.g., error-prone manually labeled data in map matching, false taxi status recording when the taxi is parking. Though visual inspection can help to justify some cases, more reliable and automatic approaches are needed to account for bigger volume of data. Secondly, preprocessing is needed to clean and transform the data so that the proposed models can be applied.

CHAPTER 6.

Conclusions and Outlook

6.1. Conclusions

This thesis is dedicated to labeling spatial trajectories in road network. It provides a novel perspective to review the currently fast evolving research topics in trajectory data mining. Being motivated by two types of labeling tasks, the author makes his major efforts on the modeling of the trajectory data using probabilistic graphical models. The thesis work along with the gained insights is summarized as follows.

- Labeling spatial trajectory serves the purpose of semantic enrichment and quality enhancement for spatial trajectories. Due to the essential role of movement study in multiple disciplines and unprecedented availability of spatial trajectory data, the problem has been widely addressed by researchers with various backgrounds in recent years. In this work, the problem is further categorized into localization and behavioral classification. Both tasks tackle the uncertainty in the data. The task of localization is to infer the actual position of the moving objects while the behavioral classification infers the latent states which can't be observed in the data. Moreover, a comparative study is carried out to discuss the commons and uniqueness of the two tasks. For instance, localization usually uses a much larger label set than behavioral classification. Both tasks can be addressed with search-based method or statistical models, and auxiliary data are often used to provide contextual information or location reference. In particular, map matching at low-sampling-rate of GPS trajectories and taxi status inference are selected as study tasks. An in-depth literature review shows that the graphical model has been successfully adopted to solve these problems.
- Labeling spatial trajectories share some common challenges despite the variety of the positioning techniques that are used for data collection. Three challenges, namely imprecise positioning, sampling rate and behavioral dynamics are common and they all together contribute to the uncertainty in labeling tasks. To address the uncertainty issues, the discriminative models for sequence labeling are studied. In particular, this thesis investigates the probabilistic modeling of spatial trajectory data in road network using CRF, which is popular for its merits of allowing arbitrary non-independent features and discriminative learning for a better fit of the data. More specifically, chain structured CRF are designed for both labeling tasks. In general, the graphical structure of the model aligns with the spatial trajectory data and uses one node for each location, but an extra node is taken for the localization task in order to make the model flexible when the labeling space grows and the sampling rate decreases. Then, a large set of features are induced with a parameter tying

strategy for practical use. Furthermore, an inference algorithm is selected according to the graphical structure which produce exact and efficient solutions. In order to maintain the tradeoff between the prediction power and the risk of overfit, ℓ_1 regularization is used to perform the feature selection in training the model simultaneously.

- The proposed method is implemented for evaluation on real-world dataset. The implementation serves three tasks, namely label data preparation and preprocessing, feature extraction, and modeling training and testing. A manual labeling procedure is designed for the localization tasks, which relies an interactive route planning tool and volunteers with driving experiences. Due to large workload, repeated visual inspections are needed to ensure the label quality. Though labels are provided for behavioral classification, but validation is also required for the error prone status attributes in the real-world data. A feature extraction module is implemented on spatial database using PL/pgSQL, which eases the early phase of understanding the data and examine the feature designs. The model training and testing are developed using Matlab for prototyping.
- Experiments are conducted on two test datasets that are derived from the real-world dataset. The results have shown that the proposed method is feasible in solving the two tasks. In map matching, the developed model outperforms the baselines marginally while yielding a significant increase in the confidence of the outputs. And in taxi status inference, the results are equally good as the reported state-of-the-art, but it demonstrates the effectiveness of applying the feature selection in reducing the model complexity. The weights of selected features also reveal the relevant features for specific tasks. In the end, a case study is also performed for map matching, which shows the cases where the model fails.

6.2. Outlook

This thesis work can serve well as stepping stones for further investigation into the topic of labeling spatial trajectories, which is drawing growing interests in research communities. Some of the potential developments are envisioned here.

- The sampling issue of the trajectory data has been addressed in this work in terms of path discovery at low-sampling rates. The proposed model treats those unobserved paths between successive location observations equally by computing their potential in the graphical model using the same set of features. However, the actual movement in the urban traffic could cause stops at traffic lights or in traffic jams, that is, the current model makes no difference to these two dramatically different mobility statuses in the sampling intervals. Statistically speaking, the model is legit. But this simple treatment could lead to an undesired smoothing effect on the probability of the data, making the prediction less certain. A refinement can be tried to enhance the fidelity of the model, which is using binary latent variables associated to the path nodes to decide if the taxis have stopped.
- Though the chained structure model manages to generate reasonable results for the labeling tasks, what is a true underlying structure of trajectory data, e.g., long term

dependency, remains an open question. This question is invoked by the manual labeling experience that even for long taxi trips, the driver tend to make only a few decisive turning points rather altering the route all the way through.

APPENDIX A: NOTATION AND MATHEMATICAL CONVENTIONS

The notations used in the thesis follow the conventions employed by (Barber, 2012; Halperin, Hartley, & Hoel, 1965; Murphy, 2012).

A.1 General Math notation

$\mathcal{X}$	A set from which values are drawn from (e.g. $\mathcal{X} = \mathbb{R}^n$).
$\lvert\mathcal{X}\rvert$	Size (cardinality) of a set.
$\mathbb{R}$	The real numbers.
$\mathbb{N}$	The natural numbers. And we make no difference in the notations of $\mathbb{N}$ and $\mathbb{N}_1 = \{1, 2, 3, \dots\}$.
$\mathbb{Z}$	The integer numbers.
$\exp(x)$	Exponential function e^x.
$\ln(x)$	Natural logarithm of x, namely $\log_e(x)$.
$\mathbb{I}(x)$	Indicator function, $\mathbb{I}(x) = 1$ if x is true, else $\mathbb{I}(x) = 0$.
$\arg\max_{\mathbf{x}} f(\mathbf{x})$	Arguments $\mathbf{x}$ of the maximum $f(\mathbf{x})$.
$\arg\min_{\mathbf{x}} f(\mathbf{x})$	Arguments $\mathbf{x}$ of the minimum $f(\mathbf{x})$.

A.2 Vector and Matrix

Vectors are denoted using boldface lowercase letters, and boldface uppercase to denote matrices. Vectors are assume to be column vectors, unless noted otherwise.

$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$, $\mathbf{v} \in \mathbb{R}^n$	A n-dimensional column vector with real-valued components.
$\mathbf{v}^\top = \begin{pmatrix} v_1 & v_2 & \dots & v_n \end{pmatrix}$	Transpose of a column vector, a n-dimensional row vector.
$\lvert\lvert\mathbf{v}\rvert\rvert$	Norm of the vector $\mathbf{v}$.
$\lvert\mathbf{v}\rvert$	Length of a vector $\mathbf{v}$.
$c\mathbf{v}$, $c \in \mathbb{R}$, $\mathbf{v} \in \mathbb{R}^n$	Scale multiplication.
$\mathbf{u} + \mathbf{v} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{pmatrix}$	Vector addition.
$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i = \mathbf{u}^\top \mathbf{v}$	Scalar product of vectors.

$\mathbf{A}_{m\times n} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$	A $m \times n$ matrix.
$\mathbf{A}^\top$	Transpose of a matrix.
$\mathbf{A}^{-1}$	Inverse of a matrix.
$\mathbf{I}$	Identify matrix.

A.3 Multivariate Calculus

Partial derivative. Consider a function of n variables, $f(x_1, x_2, \dots, x_n)$ or $f(\mathbf{x})$. The partial derivative of f w.r.t. x_i is defined as the following limit (when it exists)

$$\frac{\partial f}{\partial x_i} = \lim_{h\to 0} \frac{f(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_n) - f(\mathbf{x})}{h}$$

Gradient vector. For function f the gradient is denoted ∇f or $\mathbf{g}$:

$$\nabla f(\mathbf{x}) \equiv \mathbf{g}(\mathbf{f}) \equiv \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

A.4 Probability

X	A random variable.
x	A sample observation or non-random variable.
$P(x)$	Probability that a random variable $X \leq x$, cumulative probability function.
$p(x)$	Probability that $X = x$, probability (density) function.
$p(y\|x)$	Conditional probability of conditioning y on x.
$\widehat{\theta}$	Estimate of parameter, normally Greek letters are used to denote unknown parameters.
$\mathbb{E}[X]$	Expected value of random variable.
$\widehat{\mathbb{E}}_q[X]$	Estimated expected value of random variable with respect to distribution q.
$\mathbb{E}_q[X]$ or $\mathbb{E}_{X\sim q(x)}[X]$	Expected value of random variable with respect to distribution q.
$\mathbb{KL}(p\|\|q)$	Kullback-Liebler divergence from distribution p to q.
$X \sim q$	X is distributed according to distribution q.
Z	Normalization constant of a probability distribution.

A.5 Graphical model

$\mathcal{C}$	Cliques of a graph.

$\psi_C(\mathbf{x}_C)$	Potential function for clique C.
G	A graph.
$\mathcal{E}$	Edges of a graph.
$\mathcal{V}$	Nodes of a graph.

REFERENCES

Ali, M., Krumm, J., Rautman, T., & Teredesai, A. (2012). ACM SIGSPATIAL GIS Cup 2012. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, 597.

Andrienko, G., Andrienko, N., Bak, P., Keim, D., & Wrobel, S. (2013). *Visual Analytics of Movement*. Springer Publishing Company, Incorporated.

Andrieu, C., Freitas, N. De, Doucet, A., & Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 5–43.

Assam, R., & Seidl, T. (2014). Effective Map Matching Using Curve Tangents and Hidden Markov Model. *2014 10th International Conference on Mobile Ad-Hoc and Sensor Networks*, 213–219.

Barber, D. (2012). Background Mathematics.

Bernal, A., Crammer, K., Hatzigeorgiou, A., & Pereira, F. (2007). Global discriminative learning for higher-accuracy computational gene prediction. *PLoS Computational Biology*, *3*(3), e54.

Bierlaire, M., Chen, J., & Newman, J. (2013). A probabilistic map matching method for smartphone GPS data. *Transportation Research Part C: Emerging Technologies*, *26*, 78–98.

Bitterlich, W., Sack, J., Sester, M., & Weibel, R. (2008). *Representation , Analysis and Visualization of Moving Objects (Dagstuhl Seminar)*.

Bourdeau, A., Sahmoudi, M., & Tourneret, J. (2012). Tight integration of GNSS and a 3D city model for robust positioning in urban canyons. *ION GNSS*, *2012*.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. New York: Cambridge University Press.

Brakatsoulas, S., & Pfoser, D. (2005). On map-matching vehicle tracking data. In *Proceeding of the 31st VLDB Conference* (pp. 853–864). Trondheim, Norway.

Buchin, M., Kruckenberg, H., & Kölzsch, A. (2012). Segmenting Trajectories by Movement States. In S. Timpf & P. Laube (Eds.), *Advances in Spatial Data Handling* (pp. 15–26). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, J., & Bierlaire, M. (2013). Probabilistic Multimodal Map Matching With Rich Smartphone Data. *Journal of Intelligent Transportation Systems*, 1–15.

Collier, W. (1990). In-vehicle route guidance systems using map-matched dead reckoning. In *In Position Location and Navigation Symposium Record. The 1990's-A Decade of Excellence in the Navigation Sciences* (pp. 359–363). IEEE.

Dodge, S. (2011). *Exploring Movement Using Similarity Analysis*. Universität Zürich.

FAA. (2014). *Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report #87*. William J. Hughes Technical Center.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*(3), 37–54.

Frejinger, E. (2008). *Route choice analysis: data, models, algorithms and applications*. Thèse École polytechnique fédérale de Lausanne EPFL.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2009). *The Elements of Statistical Learning:*

Data Mining, Inference, and Prediction, Second Edition. Springer.

Ganti, R., Srivatsa, M., Ranganathan, A., & Han, J. (2013). Inferring human mobility patterns from taxicab location traces. *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '13*, 459.

Ghahramani, Z. (2012). Probabilistic Modelling, Machine Learning, and the Information Revolution. *Citeseer*.

Giannotti, F., & Pedreschi, D. (Eds.). (2008). *Mobility, Data Mining and Privacy*. Springer Berlin Heidelberg.

Giannotti, F., & Pedreschi, D. (2008). Mobility, Data Mining and Privacy: A Vision of Convergence. In F. Giannotti & D. Pedreschi (Eds.), *Mobility, Data Mining and Privacy.* Springer Berlin Heidelberg.

Giovannini, L. (2011). *A novel map-matching procedure for low-sampling GPS data with applications to traffic flow analysis*.

Goh, C., Dauwels, J., & Mitrovic, N. (2012). Online map-matching based on Hidden Markov model for real-time traffic sensing applications. In *ITSC 12'*.

Gong, H. (2011). *Generalization of road network for an embedded car navigation system*. Technische Universität München.

Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *Journal of Navigation*, *64*(03), 417–430.

Gudmundsson, J., Laube, P., & Loon, E. Van (Eds.). (2012). Representation, Analysis and Visualization of Moving Objects. In *Dagstuhl Seminar Proceedings 12512* (Vol. 2).

Guyon, I., & Elisseeff, A. (2006). An Introduction to Feature Extraction. In I. Guyon, M. Nikravesh, S. Gunn, & L. A. Zadeh (Eds.), *Feature extraction, foundations and applications* (pp. 1–25). Springer Berlin Heidelberg.

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design*, *37*(4), 682–703.

Halperin, M., Hartley, H. O., & Hoel, P. G. (1965). Recommended Standards for Statistical Symbols and Notation. COPSS Commitee on Symbols and Notation. *The American Statisticain*, *19*(3), 12–14.

He, X., Zemel, R., & Carreira-Perpinan, M. (2004). Multiscale conditional random fields for image labeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Retrieved from

Hunter, T., Abbeel, P., & Bayen, A. (2013). The path inference filter: model-based low-latency map matching of probe vehicle data. *Algorithmic Foundations of Robotics X*, 591–607.

Hunter, T., & Herring, R. (2009). Path and travel time inference from GPS probe vehicle data. *NIPS Analyzing Networks and Learning with Graphs*, 12(1)

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models. Principles and Techniques*. (Thomas Dietterich, Ed.). Cambridge, Massachusetts, London, England: The MIT Press.

Koller, D., Friedman, N., Getoor, L., & Taskar, B. (2007). Graphical Models in a Nutshell. In L. Getoor & B. Taskar (Eds.), *Introduction to Statistical Relational Learning* (pp. 13–55). The MIT Press.

Krumm, J., & Horvitz, E. (2004). LOCADIO: Inferring motion and location from Wi-Fi signal strengths. In *Proceedings of MOBIQUITOUS 2004 - 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (pp. 4–13).

Krumm, J., Letchner, J., & Horvitz, E. (2007). Map matching with travel time constraints. In *SAE World Congress*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001* (pp. 282–289).

Lange, R., Weinschrott, H., Geiger, L., & Blessing, A. (2009). On a generic uncertainty model for position information. In *Quality of Context* (pp. 76–87).

Laube, P., & Purves, R. S. (2011). How fast is a cow? Cross-Scale Analysis of Movement Data. *Transactions in GIS*, *15*(3), 401–418.

Li, M., Ahmed, A., & Smola, A. J. (2015). Inferring Movement Trajectories from GPS Snippets. In *ACM WSDM 15‘*. Shanghai, China.

Liao, L. (2006). *Location-based activity recognition*.

Liao, L., Fox, D., & Kautz, H. (2005a). Location-based activity recognition. In *NIPS’05*.

Liao, L., Fox, D., & Kautz, H. (2005b). Location-Based Activity Recognition using Relational Markov Networks. In *International Joint Conferene of Artificial Intelligence*.

Liao, L., Fox, D., & Kautz, H. (2007). Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research*, *26*(1), 119–134.

Liao, L., Patterson, D., Fox, D., & Kautz, H. (2004). Learning and inferring transportation routines. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

Liu, K., Li, Y., He, F., Xu, J., & Ding, Z. (2012). Effective map-matching on the most simplified road network. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL ’12*, (c), 609.

Liu, L. (2011). *Data Model and Algorithms for Multimodal Route Planning with Transportation Networks*.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., & Huang, Y. (2009a). Map-matching for low-sampling-rate GPS trajectories. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS ’09*.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., & Huang, Y. (2009b). Map-matching for low-sampling-rate GPS trajectories. In *ACM GIS ’09* (p. 352). ACM Press.

Martins, E. V., & Pascoal, M. B. (2003). A new implementation of Yen’s ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *1*(2), 121–133.

Matsubara, Y., Li, L., & Papalexakis, E. (2013). F-Trail: Finding Patterns in Taxi Trajectories. *Advances in Knowledge Discovery and Data Mining,* 1–12.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* (Vol. 1864, pp. 02–35).

Miller, H. J., & Han, J. (Eds.). (2001). *Geographic Data Mining and Knowledge Discovery* (First Edit). London and New York: Taylor and Francis.

Murphy, K. (2012). *Machine learning: a probabilistic perspective*. The MIT Press.

Newson, P., & Krumm, J. (2009). Hidden Markov map matching through noise and sparseness. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, 336.

Ng, A. Y., & Jordan, michael I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*.

Oran, A., & Jaillet, P. (2013). An HMM-based map matching method with cumulative proximity-weight formulation. In *Connected Vehicles and Expo (ICCVE)*.

OSM Key:Highway. (2015). Retrieved from http://wiki.openstreetmap.org/wiki/Key:highway

OSM wiki. (2015). Retrieved from https://wiki.openstreetmap.org/wiki/About

Osogami, T., & Raymond, R. (2013). Map matching with inverse reinforcement learning. In *Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence* (pp. 2547–2553).

Parent, C., & Spaccapietra, S. (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys*, *45*(4), 1–32.

Pereira, F. C., Costa, H., & Pereira, N. M. (2009). An Off-line Map-Matching Algorithm for Incomplete Map Databases. *European Transport Research Review*, *1*(3), 1–27.

Phithakkitnukoon, S., Veloso, M., Bento, C., Biderman, A., & Ratti, C. (2010). Taxi-aware map: Identifying and predicting vacant taxis in the city. In *Lecture Notes in Computer Science* (Vol. 6439 LNCS, pp. 86–95).

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C: the art of scientific computing* (2nd ed.). Cambridge University Press.

Quattoni, A., & Wang, S. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(10), 1848–1853.

Quddus, M., Ochieng, W., & Noland, R. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, *15*(5), 312–328.

Rabiner, L. R. (1989). A Tutorial on HMM and Selected Application in Speech Recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rahmani, M., & Koutsopoulos, H. N. (2013). Path inference from sparse floating car data for urban networks. *Transportation Research Part C: Emerging Technologies*, *30*(0), 41–54.

Raymond, R., & Morimura, T. (2012). Map matching with hidden Markov model on sampled road network. In *21st International Conference on Pattern Recognition (ICPR 2012)* (pp. 2242–2245). Tsukuba, Japan.

Ren, M. (2012). *Advanced map matching technologies and techniques for pedestrian/wheelchair navigation*.

Sack, J.-R., Speckmann, B., Loon, E. Van, & Weibel, R. (Eds.). (2010). Representation , Analysis and Visualization of Moving Objects. In *Dagstuhl Seminar Proceedings 10491* (pp. 1–14).

Sankararaman, S., Agarwal, P. K., Molhave, T., Pan, J., & Boedihardjo, A. P. (2013). Model-driven matching and segmentation of trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*.

Sarlas, G. (2013). *Processing low-frequency floating cardata for transportation applications.*

Schmidt, M. (2010). *Graphical Model Structure Learning with L1-Regularization*. University of British Columbia.

Schmidt, M., Fung, G., & Rosaless, R. (2009). *Optimization Methods for L1-Regularization*.

Sha, F., Pereira, F., & Science, I. (2003). Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 134–141). Association for Computational Linguistics.

Shamoun-Baranes, J., Bom, R., van Loon, E. E., Ens, B. J., Oosterbeek, K., & Bouten, W. (2012). From sensor data to animal behaviour: an oystercatcher example. *PloS One*, *7*(5), e37997.

Smyth, C. S. (2001). Mining Mobile Trajectories. In H. J. Miller & J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery* (pp. 337–361). Taylor & Francis.

Sohn, T., Varshavsky, A., Lamarca, A., Chen, M. Y., Choudhury, T., Smith, I., … Lara, E. De. (2006). Mobility Detection Using Everyday GSM Traces. *UbiComp*, 212–224.

Soleymani, A., Cachat, J., & Robinson, K. (2014). Integrating cross-scale analysis in the spatial and temporal domains for classification of behavioral movement. *Journal of Spatial Information Science*, *8*(8), 1–25.

Song, R., Lu, W., Sun, W., Huang, Y., & Chen, C. (2012). Quick map matching using multi-core CPUs. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL ’12*, 605.

Srivatsa, M., Ganti, R., Wang, J., & Kolar, V. (2013). Map matching: facts and myths. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2014)* (pp. 474–477).

Sutton, C. (2012). An Introduction to Conditional Random Fields. *Foundations and Trends® in Machine Learning*, *4*(4), 267–373.

Tao, F., & TIMMERMANS, H. J. P. (2013). Map Matching of GPS Data with Bayesian Belief Networks. In *Proceedings of the Eastern Asia Society for Transportation Studies* (Vol. 9).

Torre, F., Pitchford, D., Brown, P., & Terveen, L. (2012). Matching GPS traces to (possibly) incomplete map data: bridging map building and map matching. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 546–549).

Trajcevski, G. (2011). Uncertainty in Spatial Trajectories. In *Computing with Spatial Trajectory* (pp. p63 – p107).

Vishwanathan, S., Schraudolph, N., Schmidt, M., & Murphy, K. (2006). Accelerated training of conditional random fields with stochastic gradient methods. *Proceedings of the 23rd International Conference on Machine Learning*, 969–976.

Volz, S. (2006). An iterative approach for matching multiple representations of street data. In *Proceedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data* (pp. 101–110).

Wallach, H. (2002). *Efficient Training of Conditional Random Fields*.

Wang, G., & Zimmermann, R. (2014). Eddy: an error-bounded delay-bounded real-time map matching algorithm using HMM and online Viterbi decoder. In *Proceedings of the*

22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM.

Wang, Y., Zhu, Y., He, Z., Yue, Y., & Li, Q. (2011). *Challenges and opportunities in exploiting large-scale GPS probe data. HP Laboratories, Technical Report.*

Wei, H., Wang, Y., Forman, G., & Zhu, Y. (2013). Map matching: comparison of approaches using sparse and noisy data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*.

Wei, H., Wang, Y., Forman, G., Zhu, Y., & Guan, H. (2012). Fast Viterbi map matching with tunable weight functions. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12* (p. 613). New York, New York, USA: ACM Press.

Westgate, B. S. (2013). *Vehicle Travel Time Distribution Estimation And Map-Matching Via Markov Chain Monte Carlo Methods*. Cornell University.

Yang, J., & Meng, L. (2014). Feature Engineering for Map Mathicng of Low-Sampling-Rate GPS Trajectories in Road Network. In *SenseML'14*. Nancy.

Yang, J., & Meng, L. (2015). Feature Selection in Conditional Random Fields for Map Matching of GPS Trajectories. In G. Gartner & H. Huang (Eds.), *Progress in Location-Based Services 2014, Lecture Notes in Geoinformation and Cartography* (pp. 121–135). Springer International Publishing.

Yedidia, J., Freeman, W., & Weiss, Y. (2001). *Understanding belief propagation and its generalizations*. *Exploring artificial intelligence in the new millennium.* Cambridge, Massachusetts.

Yuan, J., Zheng, Y., & Xie, X. (2012). Discovering Regions of Different Functions in a City Using Human Mobility and POIs Categories and Subject Descriptors. In *KDD'12* (pp. 186–194).

Yuan, J., Zheng, Y., Zhang, C., & Xie, W. (2010). T-drive: driving directions based on taxi trajectories. *Proceedings of the 8th SIGSPATIAL International conference on advances in geographic information systems. ACM, 2010: 99-108.*

Zhan, F., & Noon, C. (1998). Shortest path algorithms: an evaluation using real road networks. *Transportation Science*, *10*, 65–73.

Zhang, L. (2014). *Mining GPS-Trajectory Data for Map Refinement and Behavior Detection*.

Zhang, L., Thiemann, F., & Sester, M. (2010). Integration of GPS traces with road map. *Proceedings of the Second International Workshop on Computational Transportation Science - IWCTS '10*, 17.

Zhang, M. (2009). *Methods and Implementations of Road-Network Matching*.

Zheng, K., Zheng, Y., Xie, X., & Zhou, X. (2012). Reducing Uncertainty of Low-Sampling-Rate Trajectories. *2012 IEEE 28th International Conference on Data Engineering*, 1144–1155.

Zheng, Y. (2015). Trajectory Data Mining: An Overview. *ACM Transaction on Intelligent Systems and Technology*, *6*(3).

Zheng, Y., Liu, L., Wang, L., & Xie, X. (2008). Learning transportation mode from raw gps data for geographic applications on the web. In *Proceeding of the 17th International Conference on World Wide Web '08* (pp. 247–256).

Zheng, Y., & Zhou, X. (2011). *Computing with spatial trajectories*. (Y. Zheng & X. Zhou, Eds.). New York, NY: Springer New York.

Zhu, Y., Zheng, Y., Zhang, L., Santani, D., Xie, X., & Yang, Q. (2011). *Inferring taxi status using gps trajectories*.

ACKNOWLEDGEMENTS

Having finished the thesis writing, the long-awaited relief was replaced by the vivid memories from my five-year PhD pursuit that started thousand miles away in Munich, Germany. This enduring endeavor is a mixed taste of curiosity, excitement, frustration, love and friendship. And I owe my gratitude to those great people around me and distant ones who are willing to share their enlightening academic experiences in making this thesis possible.

First and foremost, I would like to give my sincere gratitude to my supervisor, Prof. Dr.-Ing. Liqiu Meng. Prof. Meng has been a great mentor for me since my day one in the Chair of Cartography (LFK) at Technical University of Munich. She consistently encouraged my development of research interest, building of confidence and training of soft skills. With her great patience and being open minded, I may enjoy the privilege of equality and freedom to conduct my research in LFK. I believe this traditional while enlightening supervision will benefit my long term career development. And I shall cherish those fascinating stories she shared with us at the coffee breaks.

I would like to offer my gratitude to my co-supervisor, Prof. Dr.-Ing. habil. Monika Sester, for reviewing my thesis and contributing valuable comments.

My gratitude also goes to Prof. Dr. Ning Jing, Prof. Dr. Jun Li and Dr. -Ing. Lu Liu at National University of Defense Technology (NUDT), China. It was them who have encouraged me to take the challenges of doing my PhD in Germany and offered helpful advises at the critical moments of my PhD study. Without their understanding and support, I can never finish my thesis smoothly.

Colleagues at TUM have offered me incredible support as well. For my ProZeit teammates, Alexander Nottbeck, Christian Murphy and Mathias Jahnke, thanks for the constructive ideas and hard work in making ProZeit successful together. For Linfang Ding, thanks for the consistent support to all my academic initiatives in LFK, delicious dishes and encouragement at Duelferstrasse. For Hao Lv, thanks for the company for the deadlines and the competiveness on the baksetball court. For Alan Cheung, thanks for sharing his excellent research experiences and fixing my bike, many times. Thanks also go to our dear secretary Luise Fleißer and other LFK members.

My friends in Munch have given me a convenient and unforgettable life abroad. I owe my thanks to Hongchao Fan, Qing Fu, Wei Yao, Aysha Hua, Weiyong Yi, Guohui Xiao, Hongbo Gong, Yanmin Jin, Lianhuan Wei, Xiao Xie, Jiantong Zhang, Yueqin Zhu, Guiying Du, Xian Wei, Lei Lou, Shen Chi, Weijia Wang, Lin Song, Ming Jin and those lived in Felsennelkenanger 7 and Grasmeierstr. 11.

I owe my deepest gratitude to my parents, Xiang Liu and Zhaodong Yang, for their endless support and understanding. And to my girlfriend, Dr. -Ing. Chen Liu, for her caring and commitment in my toughest days.

And the thesis is financially supported by China Scholarship Council (CSC), DAAD STIBET program and the project ProZeit from Bauindustrie Bayern.

Changsha, China

Nov 2015

CURRICULUM VITAE

Personal Data

Name	Jian Yang
Nationality	Chinese
Date of Birth	7 Sept 1985
Place of Birth	Guangdong, China

Education

2003/08 – 2007/07	B.Eng. Information Engineering National University of Defense Technology, China
2007/07 – 2009/12	M.Eng. Information and Communication Engineering National University of Defense Technology, China

Experience

2010/01 – 2010/12	Research Assistant Database Research Group, College of Electronic Science and Engineering, National University of Defense Technology, China
2011/01 – 2015/05	Research Assistant Lehrstuhl für Kartographie, Technische Universität München
2015/06 – 2015/12	Research Assistant Database Research Group, College of Electronic Science and Engineering, National University of Defense Technology, China