Studies on Geospatial Mobility Analysis of Regions and Trajectories from Location Data

Hiroya Inakoshi

January 2020

Division of Computer Science and Information Technology
Graduate School of Information Science and Technology
Hokkaido University

Contents

1 Introduction **13**

1.1 Background 13

1.1.1 Regional analysis 15

1.1.2 Flow analysis 17

1.2 Contributions 21

1.2.1 Maximizing Regions in Adaptive Quadtree Mesh 21

1.2.2 Regularizing Regions by Description Length 22

1.2.3 Maximizing Posterior Joint Map-matching 22

1.3 Organization 23

2 Preliminaries **25**

2.1 Primary Data Structures 25

2.1.1 GPS observations 25

2.1.2 Regions 26

2.2 Object Functions to Optimize Regions 28

2.2.1 Empirical error and average response 29

2.2.2 Support, confidence, and gain 29

2.2.3 Standard mortality rate 32

2.3 Map-matching Problem 32

3 Maximizing Regions in Adaptive Quadtree Mesh **35**

3.1 Related Work. 35

3.2 Local Maximization of Confidence. 37

3.3 Algorithms. 39

3.4 Experimental Results and Applications. 45

3.4.1 Experiments. 45

3.4.2 Application: Taxi Fleet Control. 47

3.4.3 Application: Discovery of Sights from Social Messages. 48

3.5 Conclusion. 51

4 Regularizing Regions by Description Length **57**

4.1 Related Works 57

4.1.1 Cluster detection test 57

4.1.2 Bump hunting and bichromatic discrepancy 58

4.1.3 Optimized association rule mining 59

4.2 Proposed Method 59

4.2.1 Problem description 59

4.2.2 Regularization 61

4.3 Algorithms 64

4.3.1 Data adaptive mesh forming 64

4.3.2 Peeling 64

4.3.3 Pasting 65

4.4 Experimental Results 65

4.4.1 Generation of synthetic data 66

4.4.2 Experimental configurations 67

4.4.3 Results 67

4.5 Conclusion 71

5 Maximizing Posterior Joint Map-matching **73**

5.1 Related Works 73

5.2 Proposed Method 75

5.2.1 Stochastic generative model 75

5.2.2 Maximizing posterior probability 76

5.2.3 Graph exploration algorithm 78

5.3 Feasible Applications 82

5.3.1 Predicting traffic 82

5.3.2 Pattern discovery 83

5.3.3 Compression and anonymization 83

5.4 Experiments 83

5.4.1 Experimental configurations 84

5.4.2 Experimental results 85

5.4.3 Discussions 87

5.5 Conclusion 88

6 Conclusion **97**

List of Figures

1.1 Example of rectangular regions. 18
1.2 Example of x-monotone regions. 19
1.3 Example of transitive closures. 20

3.1 Building of quad-tree as filling hit and support counts in cells. 43
3.2 Composing transitive closure. 44
3.3 Region discovered by DISCOVERMAXIMALCLOSURE and Rastogi's algorithm for artificial data. 49
3.4 Comparison of processing time and precision between Rastogi's and our algorithms. 50
3.5 Regions discovered by FUKUDA for ununiformly distributed floating car data. 52
3.6 Regions discovered by RASTOGI for ununiformly distributed floating car data. 53
3.7 Regions discovered by proposed method DISCOVERMAXIMALCLOSURE for ununiformly distributed floating car data. 54
3.8 Locations in Okinawa where many messages with photos were posted. . 55
3.9 Locations in Yokosuka where many messages with photos were posted. 56

4.1 Counting the corners of D. Vertical and horizontal arrows represent U, and diagonal arrows represent V. 62

4.2 Precision of MTC and MTCR with K=5 and variable flatness. 71

5.1 Estimated route by single-track, proposed map-matching, and sampled trace. 90

5.2 Residuals of `icdm` versus sampling rate. 91

5.3 Residuals of `bikely` versus sampling rate. 92

5.4 Residuals of `chicago` versus sampling rate. 93

5.5 Regularization term of `icdm` versus sampling rate. 94

5.6 Regularization term of `bikely` versus sampling rate. 95

5.7 Regularization term of of `chicago` versus sampling rate. 96

List of Tables

3.1 Comparison of three algorithms for optimized rule mining of geospatial data. 36

4.1 Parameters of synthetic data. 66

4.2 Precision of four algorithms comparing ununiform to uniform and squashed to round cases. 70

5.1 Popular GPS trace datasets 84

Abstract

As many people have enjoyed their well-beings and comfortable lives owing to the industrial expansion, we are faced with new problems such as population growth, environmental pollution, and aging infrastructure. In order to continue safe and sustainable life in the future, increasing social efficiency and convenience is the major issues. For this purpose, spatial information processing attracts much attention because it helps us to analyze and understand mobility of people and things as well as accompanying consumption of resource and investment by using advanced computation and device technologies.

This includes the following technologies: the first technology is the Global Positioning System (GPS) which is a geographical measurement system using satellites. The United States developed the GPS with which smartphones, car navigation systems, digital cameras, and activity trackers are equipped to measure time and coordinates every moment. Although the GPS was opened for commercial use in 1993, advanced countries developed and have operated their own Global Navigation Satellite System (GNSS). The second technology is Digital Road Maps (DRM). Governments and private companies have developed their own DRMs for political and commercial use, respectively. Furthermore, some of the DRMs are socially maintained and distributed under an open source license.

Real-time and wide-range positioning data made yielded by these technologies is less informative on its own. However, using those technologies could help us to change

demographic and traffic behavior in such a way that can improve the efficiency and convenience of society. In this context, we focused on two problems in spatio-temporal data analysis on people and things. One is discovery of areas with high ratio of events against observed coordinates. For example, by considering the stay, inflow, and outflow of people and things as events, it becomes possible to analyze the flow of people in an open space. The other is map-matching, which is useful for analyzing traffics on road networks.

Cressie et al. have classified research on geospatial information into geostatistics and metric geography. The former treats values associated to coordinates as realizations of random variables over two-dimensional numeric space, and the latter deals with the relationship between values in a discrete space. Although there are few cross references between those two academic results due to their individual development tracks, these are collectively described as spatial statistics, which has been widely applied to public health, city or transportation planning, store opening planning, prediction of crime rate, and image analysis. As mentioned above, however, GPS records on their own are less informative. Therefore, in order to apply the aforementioned academic results, it is necessary to convert events and coordinates into quantitative data by aggregating them. But when it comes to aggregation, it is difficult to determine aggregation units in the first place because people and things as well as their associated events behave dynamically and interactively. On the other hand, analyzing traffics on road networks is a relatively newer problem and thus computing a minimizing path in terms of distance metric or object function has been of main interest. Contrarily, less attention has been paid to estimating unobservable true paths from noisy and sampled GPS traces from statistical perspective.

In this thesis, we study the followings to overcome the aforementioned difficulties. For region discovery problems, we propose algorithms that maximize expected average response of regions through the following two phases. The first phase solves an op-

timized association rule mining problem to discover regions with maximal confidence in a mesh irregularly split in accordance with GPS data. Employing the solution of the first phase as its initial solution, the subsequent phase maximizes expected average response by using an gradient-descent algorithm whose object function has a regularization term defined as description length of regions. For map-matching problems, we formalize a joint map-matching problem as a posterior maximization of observed trajectories. Motivated by this formulation, we propose an algorithm that explores true paths. After introducing preliminaries in Chapter 2, we show the contributions of this thesis in the following chapters.

In Chapter 3, a novel algorithm is presented for discovering areas having locally maximized confidence of an association rule on a collection of location data. Although location data obtained from GPS-equipped devices have promising applications, those GPS points are usually not uniformly distributed in two-dimensional space. As a result, substantial insights might be missed by using data mining algorithms that discover x-monotone or rectangular areas under the assumption that the GPS data points are distributed uniformly. The proposed algorithm composes transitively connected groups of irregular cells that have locally maximized confidence. There is thus no need to assume the uniformity, which enables the discovery of areas not limited to a certain class of shapes. Iterative removal of the cells in accordance with the local maximum property enables the algorithm to perform 50 times faster than state-of-the-art ones.

In Chapter 4, we propose a new approach to discovering regions optimizing the expected responses from data with a strong spatial bias. The methods available thus far do not work well on data of that nature because they assume that coordinates and responses are uniform and isotropic. To relax this assumption, we employ a hypothesis that cells in an irregularly sized mesh are connected transitively. However, it requires considerable computation and possibly overfits data because there are exponentially many transitive closures. Our contributions to overcome these problems are twofold:

we prove the maximal property that shows how irrelevant cells are removed without enumerating candidates in the hypothesis space, and we propose a description length of transitive closure based on which the remaining regions are regularized. We show via experiments that our algorithms do not reduce the precision with unknown data, even when such data is neither uniform nor isotropic. In addition, we show that the regularized region improves the precision by more than 20% compared to the unregularized one.

In Chapter 5, we propose a joint map-matching for estimating unobservable paths from GPS traces. This method is the first to maximize the posterior probability of stochastic generative model, in which traces are emitted as vehicles travel along the roads. We employed an EM algorithm to find the parameters of the generative model as well as to evaluate the expectations of the latent variable, which is indeed the estimated unobservable path. The EM algorithm is reduced to the exploratory search of the *route graph*, which is the geometric graph that is most likely emitting the traces and corresponds to the parameters of the model. Due to this stochastic formulation, our method works well in presence of sampling noise in the traces. Our experimental results show that the residual degradation of the estimated paths was no more than 7.0% even when they are sampled at a rate as low as 40%.

In Chapter 6, we summarize this thesis, and also point out the future direction of this study.

Chapter 1

Introduction

1.1 Background

As many people have enjoyed their well-beings and comfortable lives owing to the industrial expansion, we are faced with new problems such as population growth, environmental pollution and aging infrastructure. In order to continue safe and sustainable life in the future, increasing social efficiency and convenience is the major issues. For this purpose, spatial information processing attracts much attention because it helps us to analyzes and understand mobility of people and things as well as accompanying consumption of resource and investment by using advanced computation and device technologies.

The Global Positioning System (GPS) developed by the United States is a geographical measurement system by using satellites. It measures time and coordinates every moment [76]. GPS was released for private use in 1993 and this made cars, vessels, airplanes, and smartphones equipped with GPS receivers. And some of the developed countries developed and operated their own Global Navigation Satellite System (GNSS). In addition, Wi-Fi and ID tags enabled indoor positioning where satellite communication were difficult. Because of all these technologies, it is now possible to collect

location history both indoors and outdoors, comprehensively and inexpensively. In this thesis, the data obtained by these positioning systems is collectively referred to as GPS data which is comprised of identification number, positioning time and coordinates.

The aforementioned existence of various positioning systems suggests that the application of location information is expected in many fields such as transportation, distribution, manufacturing, and marketing [21, 51, 78, 77, 59, 69, 75]. In particular, the cluster and flow analysis of people and things are highly anticipated in a wide range of industries as long as people and things are handled. For example, it is used for plannings of transportation network, city, delivery, signage placement, etc. Therefore, in this thesis, we focus on GPS data, which is the center of digitization in the real world, and study methods to estimate the accumulation and flow of people and things from the observed incomplete data.

Cressie et al. have classified research on geospatial information into geostatistics and metric geography. The former treats spatial data as a realization of random variables over two-dimensional numeric space, and the latter deals with the relationship between values in a discrete space [17, 18, 32, 6, 9, 52]. Although there are few cross references between those two academic results due to their individual development tracks, these are collectively described as spatial statistics, which has been widely applied to public health, city or transportation planning, store opening planning, prediction of crime rate and image analysis. In those studies, modeling the probabilistic process that caused the spatial observations is of prominent interest, so it is considered useful for the purpose of this thesis. However, even though GPS data represents spatial information indeed, it is meaningless to apply the techniques from spatial statistics due to its asynchronous, imprecise and incomplete nature in the following senses:

Asynchronous GPS devices measure coordinates in different moments.

Inprecise GPS data has error because of neighboring buildings.

Incomplete Data acquisition interval is not dense, or there is a defect.

As data analysis and pre-processing for GPS data with such properties, we are interested in the following two problems: regional analysis and map-matching for flow analysis. In the next two sections, we will introduce each of the previous studies, clarify the issues when applying to GPS data with the above three properties, and motivate our research. The following sections outline the three main results of this thesis.

1.1.1 Regional analysis

This problem has been studied from different literatures including applications of congestion, convergence and divergence discovery [33, 77, 79, 71, 55], bichromatic discrepancy and bump hunting, or box rule induction, from machine learning [22, 2, 26, 37], optimized association rule mining over two-dimensional numeric space [28, 31, 63], and cluster detection test in spatial epidemiology [68, 66, 46, 74, 23, 1]. The idea of those approaches is that solving an optimization problem whose object function needs to be estimated from data is difficult because it is not easy to estimate said function. Instead, those methods try to directly discover the optimized regions from the data.

Giannotti et al. [33] defined regions of interest (RoIs). An RoI is a popular, or dense, area of points. Their *PopularRegion* algorithm extracts dense and rectangular regions. Zheng et al. [77] defined *stay regions*, which are narrow clusters of stay points where a subsequence of a trajectory is within given distance and time thresholds. Their *ExtractStayRegion* algorithm extracts dense and rectangular regions of stay points with sizes within $d \times d$, where d is a given distance threshold. Zheng and Zhou [79] studied convergence and divergence patterns representing aggregate and segregate motions, respectively, as circular or rectangular RoIs.

In machine learning literature, bump-hunting or box rule induction [26, 37] and bichromatic discrepancy [22, 2] have been formalized. Friedman et al. [26] studied the maximization of averaged response in an axis-parallel rectangular hypothesis from

point data. Dobkin et al. [22] studied bichromatic discrepancy of the axis-parallel rectangular hypothesis, which is the number of differences between positive and negative examples. Notably, this is equivalent to the minimization of disagreement between true and predictive class by definition, as well as the optimized gain association rule mining.

In data mining literature, optimized association rule mining [30, 31, 28, 63, 57] has been formalized. The goal is to obtain optimized instantiations in terms of support, confidence or gain of an uninstantiated association rule. Fukuda et al. [31, 28] proposed algorithms that need a grid-like mesh to first aggregates GPS records for every cell, and then they discover the regions with the highest score among a certain hypothesis set such as a rectangle and x-monotone region. Rastogi and Shim [63] proposed enumerating instantiations of numeric attributes before starting processing so that the identified instantiations do not overlap each other.

In geospatial information and epidemiological literature, the cluster detection test (CDT) has been studied [47, 46, 68, 67, 23, 24, 60, 40]. The CDT is comprised of two parts and the first part of maximizing Standard Mortality Rate (SMR) [10, 17, 46, 68, 67] relates to RoI discovery. Kulldorff et al.[46] proposed this framework and employed the circular hypothesis space. Tango et al. [68, 67] proposed a hypothesis set consisting of transitively connected segments in administrative districts.

Although they are from different literatures, they are all related in the sense that they discover optimal regions of their interested hypothesis like circles, rectangles, or connected cells or districts. Note that scores of *PopularRegion* and *ExtractStayRegion*, expected average response of bump hunting, confidence of optimized association rule and likelihood of CDT are equivalent. The differences lie on the hypothesis space that they employ. Figure 1.1 and 1.2 respectively show rectangular and x-monotone regions.

The aforementioned studies are all superior ones in their literatures, though, they are not directly applicable to the RoI discovery problems that handle GPS data because

of the following reasons. First, those employing circles and rectangles as the hypothesis space are implicitly assuming that GPS data and its response distribute isotropic and axis-parallel. This assumption is unrealistic for GPS application where external factors such as geographical or weather conditions dominate, or in scenarios where the mutual effects between people or objects dominate. Second, those employing meshes do not fit because GPS data is less informative on their own. In order to apply the aforementioned academic results, GPS data has to be aggregated to quantitative data. However, it is not always easy to determine the regular and moderate size mesh because GPS data has strong spatial bias as mentioned above. Finally, even if such a mesh were available, the size of hypothesis space becomes huge and there is a high risk of over-fit to the GPS data.

1.1.2 Flow analysis

Among various services enabled by spatial information processing, the analyses of flows of cars and people have enjoyed the most commercial success. For instance, analyzing traffic demands provides feedback to urban traffic design and identification of typical routes improves the efficiency of distribution services [44, 16, 45, 70, 53, 27, 7]. Map-matching served as pre-processing for those traffic applications and is commonly used to attach observed trajectories on to a digital road map (DRM). Governments and private companies have developed their own DRMs for political and commercial use, respectively. Furthermore, some of the DRMs are socially maintained and distributed under an open source license. The authors of [8] surveyed the range of map-matching techniques, and those in [62] discussed recent developments and remaining problems.

Earlier proposals for on-line map-matching algorithms attached each observation to one of the neighboring road segments by considering local connectivity of the segments [73, 34, 15]. Then, off-line map-matching algorithms were proposed, which consider the topological distances between trajectories and paths on a DRM [25, 4, 3, 49, 78,

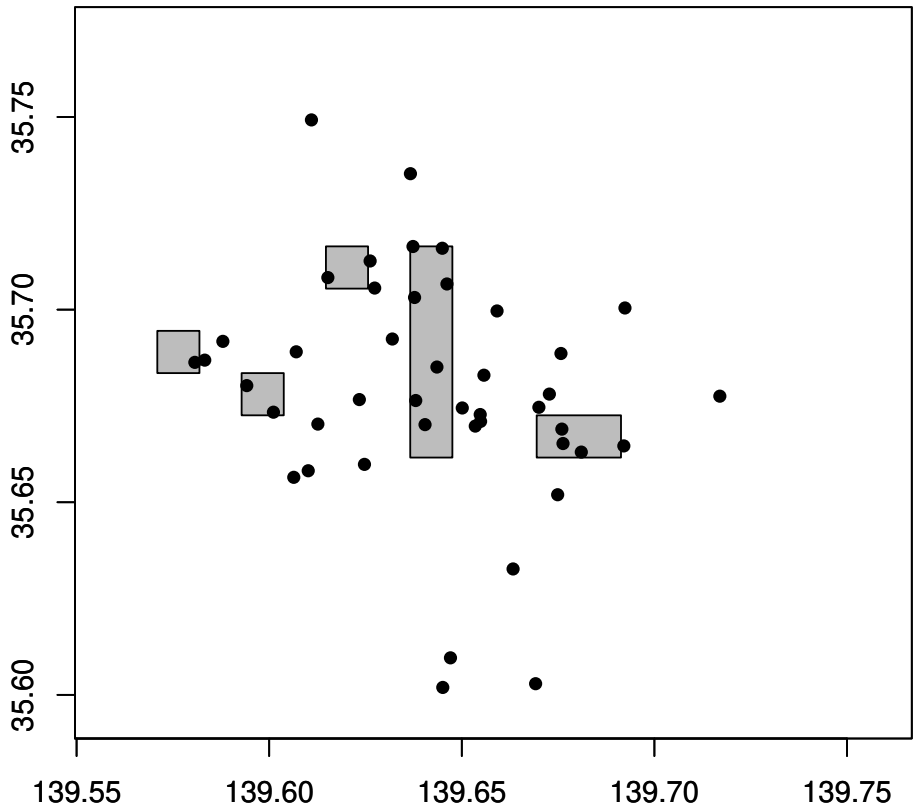


Figure 1.1: Example of rectangular regions.

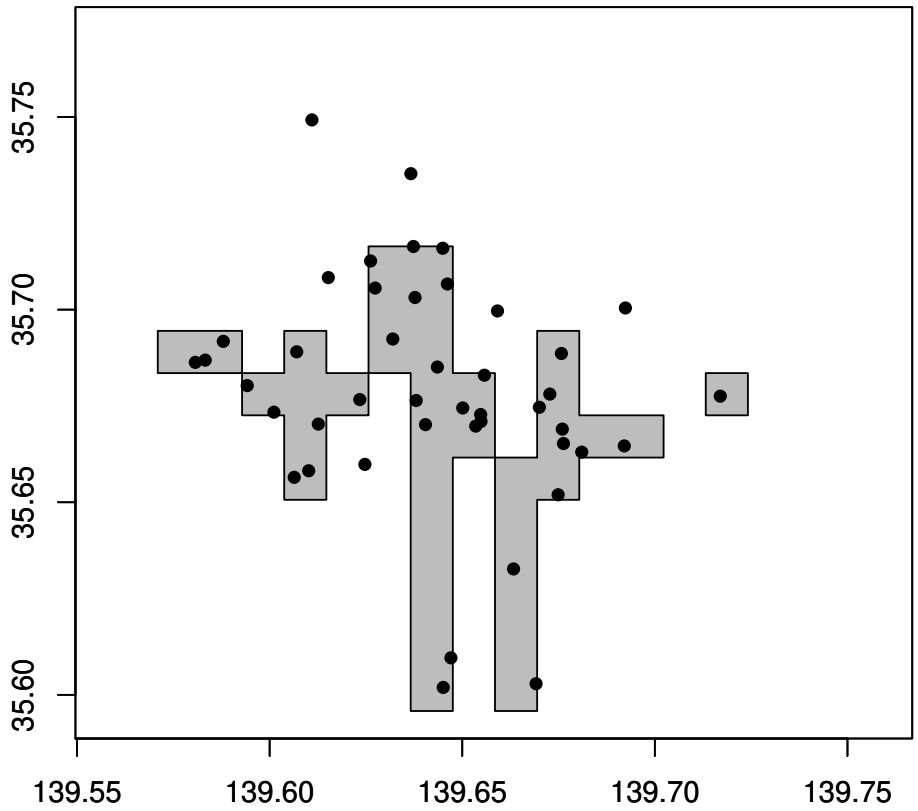


Figure 1.2: Example of x-monotone regions.

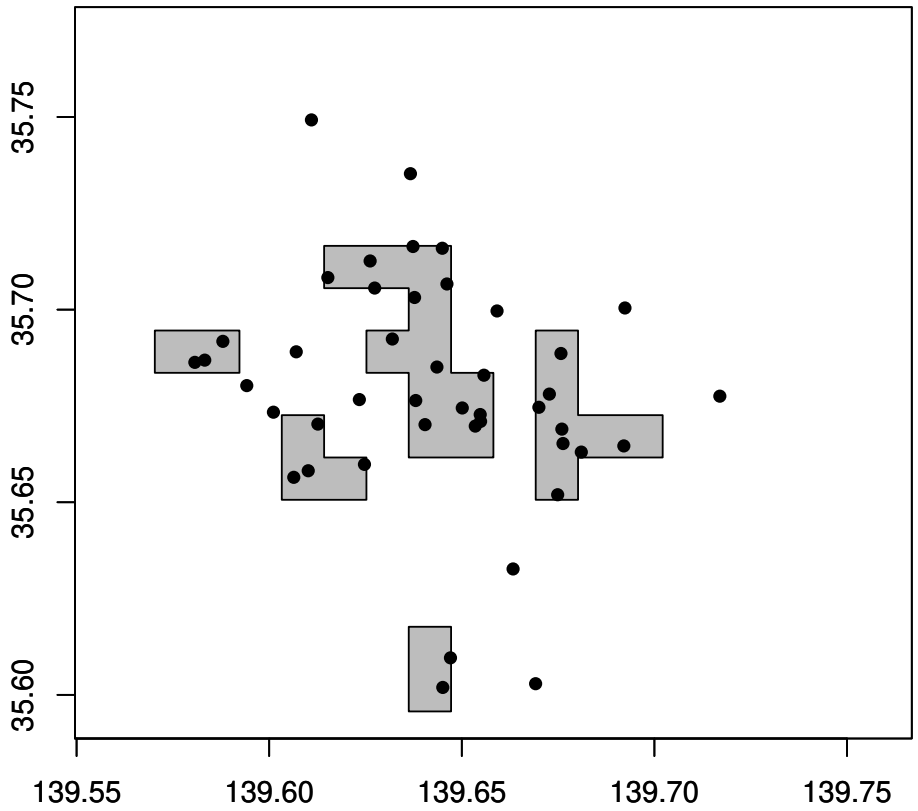


Figure 1.3: Example of transitive closures.

38, 12, 64, 50]. Probabilistic map-matching algorithms have also been proposed for estimating the road links from which observations are made [61, 5, 58]. Due to the limited network bandwidth or the constraint on power consumption, map-matching low-sample trajectories has attracted recent interest. One advanced algorithm utilizes observations from other trajectories to map a trajectory onto a DRM [41]. Another maps trajectories to the segments embedded in a DRM all at once by formalizing map-matching as an optimization problem [48].

Most of these preceding approaches, however, mainly focused on assigning trajectories to the routes that seem natural on a DRM, or computing a minimizing path in terms of distance metric or object function has been of main interest. By contrast, less attention has been paid to estimating unobservable paths, which are unaccessible in practical situations. In addition, we must pay more attention to identifying major streams in the trajectories to provide useful insight for realizing applications such as demand analysis and urban design, as mentioned above.

1.2 Contributions

1.2.1 Maximizing Regions in Adaptive Quadtree Mesh

A novel algorithm is presented for discovering areas having locally maximized confidence of an association rule on a collection of location data. Although location data obtained from GPS-equipped devices have promising applications, those GPS points are usually not uniformly distributed in two-dimensional space. As a result, substantial insights might be missed by using data mining algorithms that discover x-monotone or rectangular areas under the assumption that the GPS data points are distributed uniformly. The proposed algorithm composes transitively connected groups of irregular cells that have locally maximized confidence. There is thus no need to assume the uniformity, which enables the discovery of areas not limited to a certain class of shapes.

Iterative removal of the cells in accordance with the local maximum property enables the algorithm to perform 50 times faster than state-of-the-art ones.

1.2.2 Regularizing Regions by Description Length

In Chapter 4, we propose a method to discover regions optimizing the expected response from data with strong spatial biases. Our method employs a hypothesis set consisting of transitively connected cells in an irregularly shaped mesh. Figure 1.3 shows examples of the regions discovered by the proposed method. By building the mesh adaptively for a given set of data such that its cells have approximately the same numbers of coordinates, we can prevent areas containing a small number of coordinates from being assigned high priority, even if the coordinates are ununiformly distributed. Moreover, because this hypothesis considers a variety of shapes, the proposed method allows us to discover hotspots with non-isotropic responses.

By contrast, relaxing the hypothesis likely requires much computation, and the discovered regions tend to overfit the data because the number of transitively connected cells can be large. To overcome those problems, our contributions are as follows:

- We prove the maximal property that holds for cells outside of the optimized region and propose a peeling procedure that iteratively removes irrelevant cells without enumerating candidate regions.
- We introduce a description length of connected cells and propose a gradient descent based pasting procedure that prevents the remaining regions from overfitting by employing this length as a regularization term.

1.2.3 Maximizing Posterior Joint Map-matching

In Chapter 5, we propose a joint map-matching method, which is formulated to maximize the posterior probability of a stochastic generative model. This model represents

a process in which GPS devices on vehicles generate observations as they travel along the paths, which are actually unobservable. Using this stochastic model whose latent random variable represents an occurrence of a drive on a path, our method is able to directly estimate the unobservable paths from the observed trajectories. Our contributions are as follows: first, we present the process that generates GPS observations and formulate it as a stochastic generative model whose latent random variable represents the occurrence of a drive on a path, and whose observed random variable represents the distance between the path and the trajectory. Second, we formulate an EM algorithm that maximizes the posterior probability of the generative model. Then, we show that the log-likelihood of the posterior probability should be reduced to an object function consisting of the residual of the trajectories from their maximizing paths and the description length of the DRM. Finally, we present our algorithm, which iteratively explores the subgraphs likely to emit the observations. The experimental results show that the residual degradation was within 7.0% even if we map-match trajectories sparsified at a rate of 40%.

1.3 Organization

In Chapter 2 we give the basic notations and important concepts used in this thesis. We also refer the studies that introduced those concepts.

In Chapter 3, we propose the discovery of maximal connected regions in an adaptively sized mesh. After providing related works in Sec.3.1. we introduce the local maximum property over location data on two-dimensional plane in Sec.3.2. We then describe algorithms for extracting regions of interest in Sec.3.3. Next, we present experimental results in Sec.3.4. We then present two applications and end with a summary of the key points in Sec.3.5.

In Chapter 4, we propose the regularization of the connected regions by description

lengths to maximize average response of the regions. After providing a brief review of related works in Sec.4.1, we propose the framework of our algorithm and describe its key features along with a few preliminaries in Sec.4.2. Next, we explain the algorithms in Sec.4.3 and present our experimental results in Sec.4.4, in addition to describing application of the algorithm to taxi fleet control. Finally, we present our concluding remarks in Sec.4.5.

In Chapter 5, we propose the joint map-matching algorithm that maximizes posterior probability after observing trajectories. After providing a brief review of related works in Sec.5.1, we propose the new map-matching problem and describe its key features along with a few preliminaries in Sec.5.2, followed by applications potentially improved by combining the joint map-matching algorithm with conventional analysis techniques in Sec.5.3. Next, we present our experimental results in Sec.5.4. Finally, we conclude the paper and suggest future work in Sec.5.5.

In Chapter 6, we summarize this thesis and also point out the future direction of this study.

Chapter 2

Preliminaries

2.1 Primary Data Structures

2.1.1 GPS observations

GPS data emitted by a device is comprised of time, latitude, longitude, and identification number of object. It could be associated with other attributes. For example, data from Controller Area Network (CAN) collects a lot of measurements from sensors in a vehicle. They are recorded together with their GPS observation in an application. Consequently, GPS records are in the form of $(x, y, a_1, \cdots, a_m)$ where x and y are longitude and latitude, respectively, and $\{a_j\}_{j=1}^{m}$ are other attributes associated with position $\mathbf{x} = (x, y)$. A device identifier or time stamp could be one in a_j.

Let n, m be natural numbers and i, j, k, p, q be non-negative integers. Consider a point dataset $S = \{ (\mathbf{x}_i, l_i) \mid i = 1, 2, \cdots, n \}$ where $\mathbf{x}_i \in \mathbb{R}^2$, $l_i \in \{0, 1\}$, each of which is drawn *iid* from a probability distribution $p(\mathbf{X}, L)$ of random variables $\mathbf{X} \in \mathbb{R}^2$ and $L \in \{0, 1\}$. Let $U \subset \mathbb{R}^2$ be a two-dimensional region containing all records in S. Let $D \subseteq U$ be a region and $\overline{D} = U \setminus D$ be the complement region of D. We also denote selection of coordinates from S as S_{xy} or $S_{\mathbf{x}}$.

2.1.2 Regions

In this section, we define the regions that should be discovered. The existing methods introduced in Chapter 1 employed their own hypothesis sets of regions, such as circular or rectangular ones. Before describing those hypothesis sets, we define a mesh and its cells, as well as relation between cells to see if they share a border.

Let us consider a mesh and its cells as $M = \{c_i\}_{i=1}^{m}$ where $U = \bigcup_{i=1,\cdots,n} c_i$ and $c_i \cap c_j = \phi$ if $i \neq j$. Note that this definition only requires cells to be exhaustive and exclusive. It never requires cells to have a particular shape.

A grid-like mesh obviously follows the definition of mesh above. A mesh built using a *quad-tree*, as shown in Fig.3.1, is another example of mesh. This is useful when the distribution of points is biased because the number of points in each cell should be nearly equal. Another example is a set of polygonal meshes like administrative districts. This is useful when a mesh should be manually defined for some reasons.

Obviously, two cells may share their boundary and we call the shared part in the boundary a border. In other words, by stitching borders so that they enclose a single cell, we have a boundary of that cell. We call the border and the point at which two consecutive borders meet an edge and a corer of that cell, respectively. The relation $\mathcal{R} : M^2 \mapsto \{True, False\}$ representing whether two cells share a border satisfies $\mathcal{R}(c_j, c_k) =$ *True iff* the two cells share a border when $j \neq k$.

In the following paragraphs, we explain the concrete hypothesis sets that have been employed by previous studies.

Circular hypothesis is a class of regions whose shapes are circular with radius $r \in \mathbb{R}$ and centered at $(x_i, y_i) \in S_{xy}$ in the records.

$$H = \{(x, y) \in U \mid (x - x_0)^2 + (y - y_0)^2 \leq r \text{ where } (x_0, y_0) \in U\} \tag{2.1}$$

It assumes isotoropy of observations and responses, however, it falls short for GPS applications where geographical or weather conditions dominate. A circle could be

constrained so its center is one of the observations and its radius is the distance to its k-neighbors, which is denoted as neighbors_k.

$$H = \{(x,y) \in U \mid (x-x_i)^2 + (y-y_i)^2 \leq r \quad \text{where } (x_i,y_i) \in S_{xy}, r \in \{\sqrt{(x_j-x_i)^2+(y_j-y_i)^2} \mid j \in \text{neighbors}_k(i)\}\}. \tag{2.2}$$

This hypothesis was used in [47, 46].

Rectangular hypothesis is a class of regions whose shapes are axis-parallel rectangular.

$$H = \{(x,y) \mid x \in [x_l, x_u], y \in [y_l, y_u] \text{ where } (x_l,y_l), (x_u,y_u) \in U\} \tag{2.3}$$

It assumes axis-wise isotoropy of observations and responses, but is easier to design efficient algorithms. Considering all rectangles with different observations in them, it is enough to consider the following hypothesis.

$$H = \{(x,y) \mid x \in [x_i, x_{i'}], y \in [y_j, y_{j'}] \text{ where } x_i, x_{i'} \in S_x,\ y_j, y_{j'} \in S_y\} \tag{2.4}$$

This hypothesis was employed by [22, 37] and its size is $O(n^4)$. Some methods employ a grid-like mesh, which split the domain U into m either vertically or horizontally. Let its cells indexed by (p,q) where $p,q = 0, \cdots, m-1$. In other words, we have intersections at $\{(x^{(p)}, y^{(q)}) \mid p,q \in \{0,1,\cdots,m\}\}$ in the mesh. By mapping the index (p,q) to the cell c_{mp+q+1}, the rectangular hypothesis over the grid-like mesh is given as:

$$H = \{c_{mi+j+1} \mid x^{(i)} \in [x^{(p)}, x^{(p')}),\ y^{(j)} \in [y^{(q)}, y^{(q')}) \text{ where } p,p',q,q' \in \{0,1,\cdots,m\}\}. \tag{2.5}$$

This hypothesis was employed by [29, 28, 63].

***x*-monotone hypothesis** is a class of closed regions who are undivided with any vertical line. The formal definition of x-monotone hypothesis is given below:

$$H = \{(x',y') \mid (x',y') \in D \text{ if } y'_{\min} \leq y' \leq y'_{\max} \text{ for } x_{\min} \leq \forall x' \leq x_{\max}\} \tag{2.6}$$
$$\text{where } \begin{cases} x_{\min} = \min_{(x,y)\in D} x, & x_{\max} = \max_{(x,y)\in D} x, \\ y'_{\min} = \min_{(x',y)\in D} y, & y'_{\max} = \max_{(x',y)\in D} y \end{cases}$$

The x-monotone hypothesis over the grid-like mesh is give below:

$$H = \{c_{mi+j+1} \mid (x^{(i)}, y^{(j)}) \in D \text{ if } q_i \leq j < q'_i \text{ for } p \leq \forall i < p'\} \tag{2.7}$$
$$\text{where} \begin{cases} p = & \min_{(x^{(k)}, y^{(j)}) \in D} k, & p' = & \max_{(x^{(k)}, y^{(j)}) \in D} k, \\ q_i = & \min_{(x^{(i)}, y^{(k)}) \in D} k, & q'_i = & \max_{(x^{(i)}, y^{(k)}) \in D} k \end{cases}$$

This hypothesis was employed by [29, 28].

Cluster hypothesis is a class of regions defined by using mesh M. We can enclose an arbitrary region D on the mesh M by stitching borders. Formally, the cluster hypothesis is defined as

$$H = \{D \mid \mathcal{R}^+(c_j, c_k) = \mathit{True} \text{ for all } c_j, c_k \in D\} \tag{2.8}$$

where $\mathcal{R}^+$ denotes the transitive closure of $\mathcal{R}^+$. We denote this hypothesis H as $\mathcal{R}^+$ because the hypothesis is defined by using $\mathcal{R}^+$. This definition does not assume isotropy. Tango et al. employed the cluster hypothesis, given administrative districts as the mesh [67], although they imposed constraints on diameter of regions to ignore unnaturally distorted ones.

Finally, we define vector representation of region if it is on a mesh. D is denoted as a column vector as

$$\mathbf{x}_D = (x_{Dj})_{j=1}^m, \text{ where } x_{Dj} = \begin{cases} 1 & \text{if } c_j \in D, \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

We also define $\mathbf{z}_D \in \mathbb{R}^m$ to introduce the continuous version of $\mathbf{x}_D = \sigma(\mathbf{z}_D) \in (0,1)^m$ where $\sigma(\cdot)$ is a sigmoid function. The continuous version is used when we execute the pasting in Sec.4.2.2 to regularize the discovered regions.

2.2 Object Functions to Optimize Regions

In this section, we introduce object functions employed in existing methods to obtain optimized regions. Some of them are mutually equivalent as shown in later chapters.

2.2.1 Empirical error and average response

Bichromatic discrepancy is to minimize empirical error given below:

$$\mathrm{Err}_S(D \in H) = \sum_i (l_i = 0) \oplus (\mathbf{x}_i \in D). \tag{2.10}$$

Dobkin et al. studied the theory on bichromatic discrepancy and presented an algorithm that discover rectangular region minimizing the empirical error.

Let $\mathbf{X} \in \mathbb{R}^2$ and $L \in \{0, 1\}$ random variables which follow the joint probability $p(\mathbf{X}, L)$. Average response is given below:

$$E[L \mid \mathbf{X} \in D] = \int lp(l \mid \mathbf{x} \in D)\mathrm{d}l. \tag{2.11}$$

The goal of box rule induction or bump-hunting [26, 37] is, given a dataset S, to obtain D maximizing $E[L|\mathbf{X} \in D]$. Note that $p(\mathbf{X}, L)$ is unknown. Thus, box rule induction has two steps. It first iteratively peels either lower or higher end of the variables' domain one at a time until the averaged response of the remaining rectangular region stops decreasing, and it then performs pasting to ensure that the rectangular region does not overfit the concrete training data.

2.2.2 Support, confidence, and gain

An association rule has the form $C_1 \to C_2$ where $C_i (i = 1, 2)$ are logical conditions on attributes of records. Their Boolean values are fixed by instantiating attributes with values, although we omit them from the expression for simplicity. Each association rule has associated support and confidence. Let the support for condition $C_i (i = 1, 2)$ be the number of records satisfying C_i. It is represented as $sup(C_i)$. The support for a rule in the form $C_1 \to C_2$ is then the same as the support for C_1 while its confidence

is the ratio of the supports of conditions $C_1 \wedge C_2$ and C_1:

$$sup(C_1 \to C_2) = sup(C_1), \tag{2.12}$$

$$conf(C_1 \to C_2) = \frac{sup(C_1 \wedge C_2)}{sup(C_1)}. \tag{2.13}$$

A rule is *confident* if its confidence is not less than a given confidence threshold θ. A rule is *ample* if its support is not less than a given support threshold Z.

The hit and gain of a rule are defined as:

$$hit(C_1 \to C_2) = sup(C_1 \to C_2) conf(C_1 \to C_2), \tag{2.14}$$

$$gain(C_1 \to C_2) = hit(C_1 \to C_2) - \theta\ sup(C_1 \to C_2). \tag{2.15}$$

Although many studies have considered various forms of the presumptive and objective conditions, here we consider the case in which both are conjunctions of atomic conditions, which are $a_j = v$ for nominal attributes, $a_j \in [l, u]$ for numeric attributes, and $(x, y) \in D \subset \mathbb{R}^2$ for location attributes.

In the optimized association rule mining, the presumptive condition can have uninstantiated atomic conditions. Atomic conditions are uninstantiated (*resp.* instantiated) when one of v, $[l, u]$, and D is a variable (*resp.* a value). If the uninstantiated and instantiated conditions are separately written, the association rule $U \wedge C_1 \to C_2$ appears. If U_i denotes an instantiation of U, U_i can be obtained by replacing variables in U with values.

Optimized association rule mining is categorized into three types:

1. An *optimized support problem* is to discover confident instantiations that maximize the support.

2. An *optimized confidence problem* is to discover ample instantiations that maximize the confidence.

3. An *optimized gain problem* is to discover ample instantiations that maximize the gain.

The related works introduced in Sec. 3.1 consider some or all of those types of problems. We concentrated on the optimized confidence problem because it was the most important problem for the geospatial applications in which we were interested.

Example 1 *Consider the database of a social messaging service. Let a record in the database be denoted by* $(x, y, id, year, mon, mday, hour, photo)$*, where photo is a Boolean attribute indicating whether a photo is attached to the message. Let the condition for position* $(x, y) \in D$ *be uninstantiated by letting* D *be a variable and the conditions* $hour \in [l, u]$ *and* $photo = v$ *be instantiated by letting* $[l, u]$ *and* v *be values. An uninstantiated association rule is thereby obtained.*

$$((x, y) \in D) \wedge (hour \in [l, u]) \rightarrow (photo = v),$$
$$where \begin{cases} l = 5:00, u = 8:00, \\ v = True \end{cases}$$

To find an area where messages are likely attached to photos and posted in the early morning, we solve the optimized confidence problem by discovering ample instantiations of variable D *such that the confidence of the uninstantiated rule above is maximized.*

Note that either cell c_j or region D includes some points in S. The number of points in c_j is called *support* of the cell and denoted as $s_j = sup(c_j)$. The number of points in c_j such that $l_i = 1$ is called *hit* of the cell and is denoted as $h_j = hit(c_j)$. Similarly, $sup(D)$ and $hit(D)$ denote the support and the hit of region D. $sup(c_j)$ and $hit(c_j)$ are represented by the column vectors, $\mathbf{s} = (s_j)_{j=1}^m$ and $\mathbf{h} = (h_j)_{j=1}^m$, respectively.

Given an association rule, optimized association rule mining is to discover instantiations of the rule so that it maximize either support, confidence, or gain. An attributes could be a location as well as numeric or nominal. Fukuda et al. [30] solved optimized support and confidence problems on two-dimensional numeric attributes in rectangular hypothesis, and presented $O(n)$ time algorithms. They also presented $O(\sqrt{n})$ time algorithm for optimized gain problem in x-monotone hypothesis.

2.2.3 Standard mortality rate

In spatial epidemiology literature, spotting regions with high SMR of a particular disease has been of interest. The final goal is to spot the region with statistical significance, however, discovery of region with maximal SMR is important as its first step. The SMR of region is formalized below:

$$\mathrm{SMR} = \frac{\sum_{c_i \in D} d_i}{\sum_{c_i \in D} e_i}. \tag{2.16}$$

where d_i and e_i are number of death and expected number of population, respectively. The expected number of population e_i is corrected by age group. If the numbers in age groups are not available, e_i is identical to the number of population.

2.3 Map-matching Problem

In this section, we define two map-matching frameworks. One is the single-track map-matching that is employed by the joint map-matching proposed in Chapter 5. The other is the multi-track map-matching as the joint map-matching is categorized to it. We leave concrete conventional methods to the survey papers [8, 62] as there are a huge number of studies. Many of them are categorized either of the frameworks defined later in this section.

Let i, j be non-negative integers and k, n, p, q, N, K be natural numbers. We call $G = (V, E)$ a geometric graph, or simply a graph, where $V = \{(x, y) \mid x, y \in \mathbb{R}\}$ and $E = \{(u, v) \mid u, v \in V \text{ and } u \neq v\}$. Denoting a list as $[\,]$ whose elements are ordered by i or j, an element α in $\mathbb{P}$ is a trajectory of length p, where $\alpha = [\,\alpha^{(i)} \in \mathbb{R}^2 \mid i \leq p\,]$, and an element β in $\mathbb{P}_G$ is a path in G of length q, where $\beta = [\,\beta^{(j)} \in V \mid j \leq q \text{ where } (\beta^{(j-1)}, \beta^{(j)}) \in E \text{ if } j \geq 1\,]$. Both trajectory and path in G are polylines. Note that they may contain an element multiple times and that we have $\mathbb{P}_G \subset \mathbb{P}$. A path can be regarded as a sub-graph of G and we denoted it as $G(\beta) = (V_\beta, E_\beta)$ where

$V_\beta = \{\beta^{(j)} \mid j \leq q\}$ and $E_\beta = \{(\beta^{(j-1)}, \beta^{(j)}) \mid 1 \leq j \leq q\}$. To introduce binary set operators on graphs, we define an edge-induced graph of given E as $G(E) = (V, E)$ where $V = \bigcup_{(u,v)\in E}\{u, v\}$. With this definition, given two graphs $G_i = (V_i, E_i)$ for $i \in \{1, 2\}$, we have $G(E_1) \circ G(E_2) = G(E_1 \circ E_2)$ where $\circ$ is a binary set operator.

Before defining single- and multi-track map-matching frameworks, we define the distance function between polylines. For example, Fréchet distance follow this definition. The two frameworks are defined by using the distance function.

Definition 1 *(Distance function): Let $d : \mathbb{P} \times \mathbb{P} \to \mathbb{R}$ be a distance function between polylines, where the following inequality and equalities hold for all $\alpha, \beta \in \mathbb{P}$:*

$$d(\alpha, \beta) \geq 0, \quad d(\alpha, \alpha) = 0, \quad d(\alpha, \beta) = d(\beta, \alpha).$$

A single-track map-matching maps a trajectory to a path on a geometric graph by minimizing the distance. Many existing off-line map-matching follow this definition [4, 3, 78]. A multi-track map-matching maps a collection of trajectories to paths on a geometric graph by minimizing an object function. The object function usually has its residual term which comes with other terms. The residual term penalize the solution in terms of distance between trajectories and their paths as it is done by a single-track map-matching. On the other hand, the other terms penalize in terms of complexities of the solution, which differentiate multi-track map-matching from single-track map-matching.

Definition 2 *(Single-track map-matching) Let $G = (V, E)$ be a geometric graph and $d(\alpha, \beta)$ be a distance function between polylines. Given a trajectory $\alpha \in \mathbb{P}$, a single-track map-matching algorithm, or simply a map-matching algorithm $\mathcal{M}_G : \mathbb{P} \to \mathbb{P}_G \times \mathbb{R}$ finds its minimizing path and its minimum distance, which are $\hat{\beta}_G(\alpha) = \operatorname{argmin}_{\beta\in\mathbb{P}_G} d(\alpha, \beta)$ and $\hat{d}_G(\alpha) = \min_{\beta\in\mathbb{P}_G} d(\alpha, \beta)$, respectively.*

Definition 3 *(Multi-track map-matching) Let $G = (V, E)$ be a geometric graph and $d(\alpha, \beta)$ be a distance function between polylines. Given a collection of trajectory $T = \{\alpha_i\}_{i=1}^n$, a multi-track map-matching algorithm $\mathcal{M}_G : \mathbb{P}^{|T|} \to \mathbb{P}_G^{|T|} \times \mathbb{R}$ maps α_i to β_i so that they minimize a object function $\sum_{\alpha \in T} d(\alpha_i, \beta_i) + \lambda \|\{\beta_i\}_{i=1}^n\|$, where λ is a positive constant and $\|\{\beta_i\}_{i=1}^n\|$ is a certain function that penalize strange paths.*

The penalty depends on methods to methods. For example, a method introduced stitching and regularity terms [48].

Chapter 3

Maximizing Regions in Adaptive Quadtree Mesh

In this chapter, we study a method for maximizing regions in adaptive quadtree mesh.

3.1 Related Work.

Optimized association rule mining was first introduced by Fukuda et al. They formulated discovery of association rules for a single numeric attribute [31] and two-dimensional numeric attributes [28]. Their algorithms include an $O(m)$ time algorithm, where m is the size of mesh for a x-monotone region, that maximizes the gain achieved by using dynamic programming (DP) with fast matrix search, $O(m^{1.5})$ time algorithms for a single rectangular region that maximizes the gain, and approximation algorithms for a single rectangular region that maximizes the support or confidence.

These algorithms identify a single rectangle or x-monotone region because their goal is to keep the optimization rules simple enough for people to easily understand them. Note that they are based on the assumption that the uninstantiated numeric attributes are uniformly distributed. Therefore, they start by splitting relations into

	Fukuda	Rastogi	Dobkin
assumes uniformity?	yes	yes	no
no. of regions	1	k	1
x-monotone regions	yes	no	no
rectangular regions	yes	yes	yes

Table 3.1: Comparison of three algorithms for optimized rule mining of geospatial data.

equal-sized buckets or grid-like meshes for one- or two-dimensional data, respectively.

Rastogi and Shim generalized the formulation of optimized association rule mining [63]. Their optimized association rules can have disjunctions over an arbitrary number of uninstantiated attributes, which can be either categorical or numeric. This means that the discovered instantiations can be multiple hyper rectangular regions.

They also showed that their problem is NP-hard and proposed algorithms that search through the space of instantiations in decreasing order of the weighted sum of the confidence and support, by using a branch and bound technique to prune the search space efficiently.

Their algorithms require the enumeration of instantiations of numeric attributes, each of which has the form $a_i \in [l_i, u_i]$, before they start processing so that the identified instantiations do not overlap each other. The number of instantiations could be huge for numeric attributes since they take conbinations of values. The *pruneInstArray* reduces those instantiations that are never included in the final disjunctions of instantiations. Note that the numeric attributes are assumed to be uniformly distributed in Rastogi's algorithms as in Fukuda's algorithms.

Dobkin et al. [22] presented an algorithm that solves optimized gain problems for a single rectangular region. It takes $O(n^2 \log n)$ time, where n is the number of points, and does not necessarily suppose that numeric attributes are uniformly distributed since it takes all distinct values of those numeric attributes in the records.

Table 3.1 compares the algorithms of Fukuda, Rastogi, and Dobkin.

3.2 Local Maximization of Confidence.

Our goal was to develop an algorithm that does not require the numeric attributes to be uniformly distributed and that identifies multiple regions. Both features are important for us to discover RoIs in geospatial data. We achieved both by introducing locally optimized association rule mining, which provides another mathematically sound formalization for discovering RoIs.

Definition 4 *For given a hypothesis H, a threshold Z, and $D \in H$, we define $\widetilde{D}$ as the maximized region with respect to confidence satisfying*

$$sup(\widetilde{D}) \geq Z \text{ and } conf(D) < conf(\widetilde{D}) \text{ for any } D \supset \widetilde{D}. \tag{3.1}$$

Informally, this means that $\widetilde{D}$ is maximal if there is no area D containing $\widetilde{D}$ with confidence greater than $\widetilde{D}$. The following lemma trivially holds:

Lemma 1 *Given a mesh $M = \{c_i\}_{i=1}^{m}$ and an association rule $C_1 \rightarrow C_2$, let s_i and h_i be the support and hit of cell c_i, respectively. For any $E \in H$,*

$$conf(E) = \frac{\sum_{c_i \in E} h_i}{\sum_{c_i \in E} s_i} \leq \frac{1}{|E|} \sum_{c_i \in E} \frac{h_i}{\min_{c_i \in E} s_i}, \tag{3.2}$$

where $|E|$ is the number of cells in E.

For a naive algorithm that enumerates all possible transitive closures of $\mathcal{R}$, it could take exponential time to compute the locally maximized confidence regions, $\widetilde{D} \in 2^M$, since there are $\sum_{k=1,\cdots,n} {}_nC_k$ possible transitive closures.

The following theorem presents the condition that each cell should satisfy when it is not included in the maximum area.

Theorem 1 *For* $\forall \widetilde{D}, D \in H$ *such that* $\widetilde{D} \subset D$, *let* $E = D \setminus \widetilde{D}$. *Then,* $conf(D) < conf(\widetilde{D})$ *holds if*

$$\frac{h_i}{\min_{c_i \in E} s_i} < conf(D) \text{ for } \forall c_i \in E. \tag{3.3}$$

Proof 1 *To keep the formulas simple, we introduce some invariants:* $S_E = \sum_{m_i \in E} s_i$, $H_E = \sum_{m_i \in E} h_i$, $S_{\widetilde{D}} = \sum_{m_i \in \widetilde{D}} s_i$ *and* $H_{\widetilde{D}} = \sum_{m_i \in \widetilde{D}} h_i$. *Since* $D = E \cup \widetilde{D}$ *and* $E \cap \widetilde{D} = \phi$,

$$S_{\widetilde{D}} = S_D - S_E,$$
$$H_{\widetilde{D}} = H_D - H_E.$$

Now $conf(\widetilde{D}) - conf(D)$ *is evaluated:*

$$\begin{aligned} conf(\widetilde{D}) - conf(D) &= \frac{H_{\widetilde{D}}}{S_{\widetilde{D}}} - \frac{H_D}{S_D} \\ &= \frac{S_D(H_D - H_E) - (S_D - S_E)H_D}{S_{\widetilde{D}} S_D} \\ &= \frac{S_E}{S_{\widetilde{D}}} \frac{S_E H_D - S_D H_E}{S_E S_D} \\ &= \frac{S_E}{S_{\widetilde{D}}} \left(\frac{H_D}{S_D} - \frac{H_E}{S_E} \right) \\ &= \frac{S_E}{S_{\widetilde{D}}} \left(conf(D) - conf(E) \right). \end{aligned} \tag{3.4}$$

By summing (3.3) *over* $c_i \in E$ *and combining this with* (3.2), *we get*

$$conf(E) \leq \frac{1}{|E|} \sum_{c_i \in E} \frac{h_i}{\min_{c_i \in E} s_i} < conf(D). \tag{3.5}$$

From (3.4) and (3.5), $conf(\widetilde{D}) > conf(D)$ *is proved.*

Theorem 1 implies that it is not necessary to enumerate all combinations of cells and evaluate their confidence as a naive algorithm would do. Instead, it is sufficient to see if Eq.(3.3) holds for each individual mesh. However, Eq.(3.3) is unlikely to hold for most of the cells if $\min_{m_i \in E} s_i$ is too small. This is the case for car probe data and location data from other movable objects, where the position data is not distributed

Algorithm 1 DISCOVERMAXIMALCLOSURE(S, M, C_1, C_2, Z)

1: $M \leftarrow$ BUILDQUADTREE(S, M, C_1, Z)
2: obtain $sup(m)$ and $hit(m)$ of $C_1 \rightarrow C_2$ for $\forall m \in M$
3: $\Gamma \leftarrow \{M\}$
4: **repeat**
5: **for** $D \in \Gamma$ **do**
6: $E \leftarrow \{\}$
7: $\delta \leftarrow \min_{m \in D} sup(m)$
8: **for all** $m \in D$ such that $hit(m)/\delta < conf(E)$ **do**
9: $E \leftarrow E \cup \{m\}$
10: **if** $E = \{\}$ **or** $sup(D \backslash E) < Z$ **then**
11: report D
12: $E = D$
13: $M \leftarrow M \backslash E$
14: $\Gamma \leftarrow$ COMPOSETRANSITIVECLOSURE(M)
15: **until** no report occurred at line 11

uniformly. In those cases, it is better to use a mesh of cells whose support counts are almost equal to each other.

With this in mind, we present an algorithm for discovering regions with locally maximal confidence in the following section. We also mention an algorithm for reducing the number of meshes to be checked to see if they share borders.

3.3 Algorithms.

In the remaining sections of this chapter, some of the notations are replaced to different ones. Let Mesh M be comprised of $\{m_i\}_{i=1}^{n}$, instead of $\{c_i\}_{i=1}^{m}$, and S be of size N, instead of n.

Algorithm 1 shows our algorithm for discovering locally maximal regions. The algorithm comprises two parts. The first part initializes a mesh and calculate the support and hit for each cell (lines 1–3). It does this by using a position database S, an initial fixed-sized mesh set M, the association rule to be optimized $C_1 \rightarrow C_2$, and the minimum support threshold Z. One such mesh is called *quad-tree*. As illustrated in Fig. 3.1, it is obtained by recursively splitting a cell evenly into four smaller cells with horizontal and vertical borders. The algorithm continues to split the given mesh M into finer one so long as its cells have more points than the minimum support threshold Z. The white and black points in the figure represent position data satisfying C_1 and $C_1 \wedge C_2$, respectively. The result is different size cells with support counts less than $Z = 2$.

The second part of the algorithm identifies transitively connected groups of cells with locally maximal confidence (lines 4–15). The algorithm iteratively narrows a transitive closure by removing cells that do not satisfy the local maximum property given by Theorem. 1 with δ determined for that transitive closure (lines 8–9). Using $\min_{m \in D} \mathit{sup}(m)$ for δ is not any problem since $\delta \leq \min_{m \in D \setminus \widetilde{D}} \mathit{sup}(m)$ always holds. By sorting M in ascending order of $\mathit{conf}(m)$, each mesh is visited only once by line 9. Therefore, the nested loop (lines 5–13) iterates at most n times. If the transitive closure D is maximal and ample, then it is reported at line 11. The remaining cells are processed by COMPOSETRANSITIVECLOSUREfunction, which composes transitive closures for the next iteration (line 14).

The computation time of DISCOVERMAXIMALCLOSURE is $O(n \log n)$ since COMPOSETRANSITIVECLOSURE takes $O(n \log n)$ time as shown below and the outer most loop iterates constant times proportional to Z.

It takes $O(n^2)$ time to naively check if the remaining cells share borders with each other. Figure 3.2 shows an outline of the sweep-line algorithm used to reduce the number of cells to be checked to see if they share borders. A sweep-line is a vertical

Algorithm 2 composeTransitiveClosure(M)

1: let Q be a list comprising left-bottom and right-top vertices of $\forall m \in M$
2: sort Q in the order representing the time when cells become active or inactive
3: $A \leftarrow \{\}$
4: $E \leftarrow \{\}$
5: **while** Q is not empty **do**
6: poll one from Q and call it v_1
7: let m_1 be the mesh of v_1
8: **if** v_1 is a left-bottom vertex, **then**
9: $A \leftarrow A \cup \{m_1\}$
10: $E \leftarrow E \cup \{(m_1, m_2) |\ m_2 \in A$ s.t. $m_1.btm \leq m_2.top$ **and** $m_1.top \geq m_2.btm\}$
11: **else**
12: $A \leftarrow A \backslash \{m_1\}$
13: $V \leftarrow \cup_{(m_1,m_2)\in E}\{m_1, m_2\}$
14: report all connected graphs in $G(V, E)$ by traversing it in breadth first manner

line that makes cells *active* while it crosses over them. All cells are initially inactive. As the sweep-line moves from left to right, only the active cells are checked to see if they share borders with each other. If one shares a border with another, the pair is output. Such pairs are never missed because no two cells share a border unless both of them become active at the same time. A cell becomes inactive again as the sweep-line moves away from the top of that cell and thus is never examined with other cells. Once the sweep-line has traversed completely to the right, all pairs of adjacent cells have been identified and output. By traversing a graph comprised of those pairs as its edges in a breadth first manner, the sweep-line algorithm composes all transitive closures.

Algorithm 2 implements this process. The sweep-line is emulated by using queue Q containing left-bottom and right-top vertices sorted in the order that represents

the time when the cells become active and inactive, respectively (line 1–2). A brief description of how the lines 5–12 work is given in the caption of Fig 3.2. It computes in $O(n \log n)$ time even if the cells have arbitrary sizes and shapes as long as they can be represented as polygons. Interested readers can consult a text book on computational geometry, such as [20].

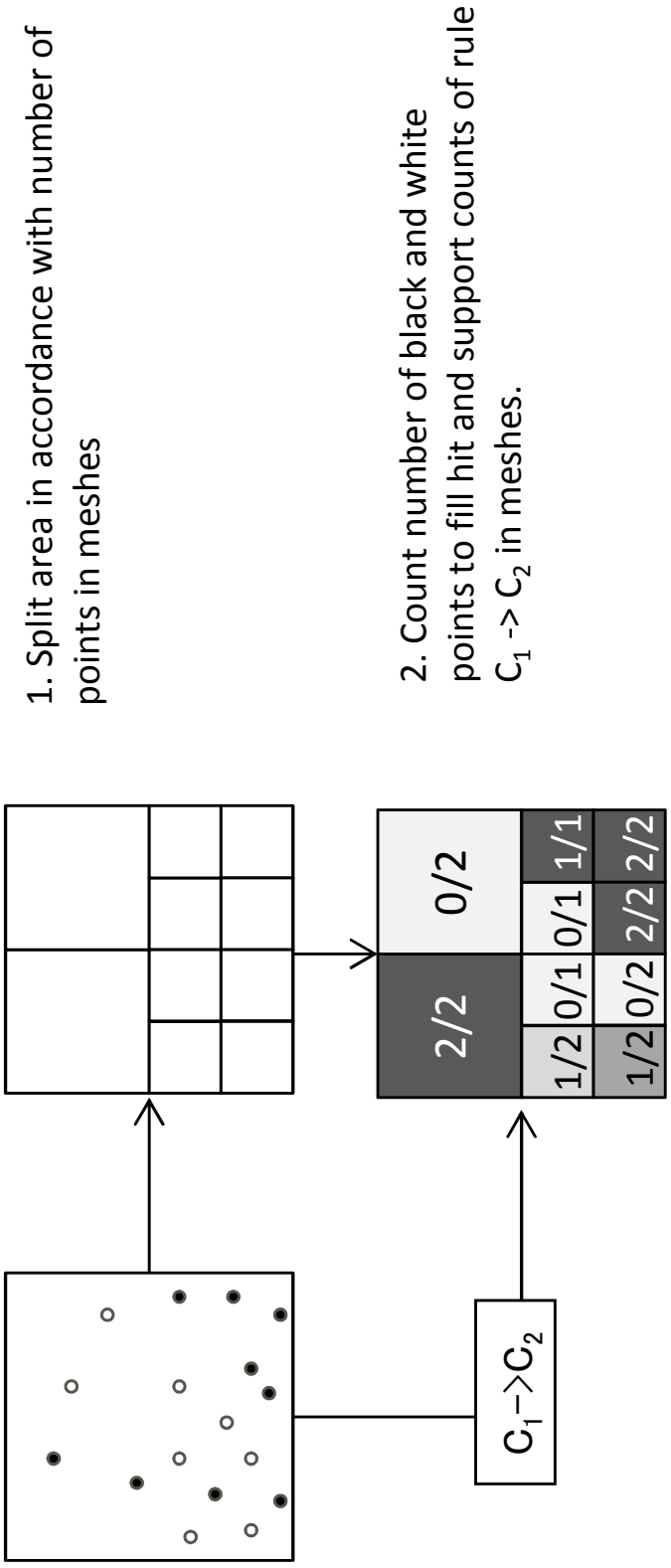


Figure 3.1: Building of quad-tree as filling hit and support counts in cells.

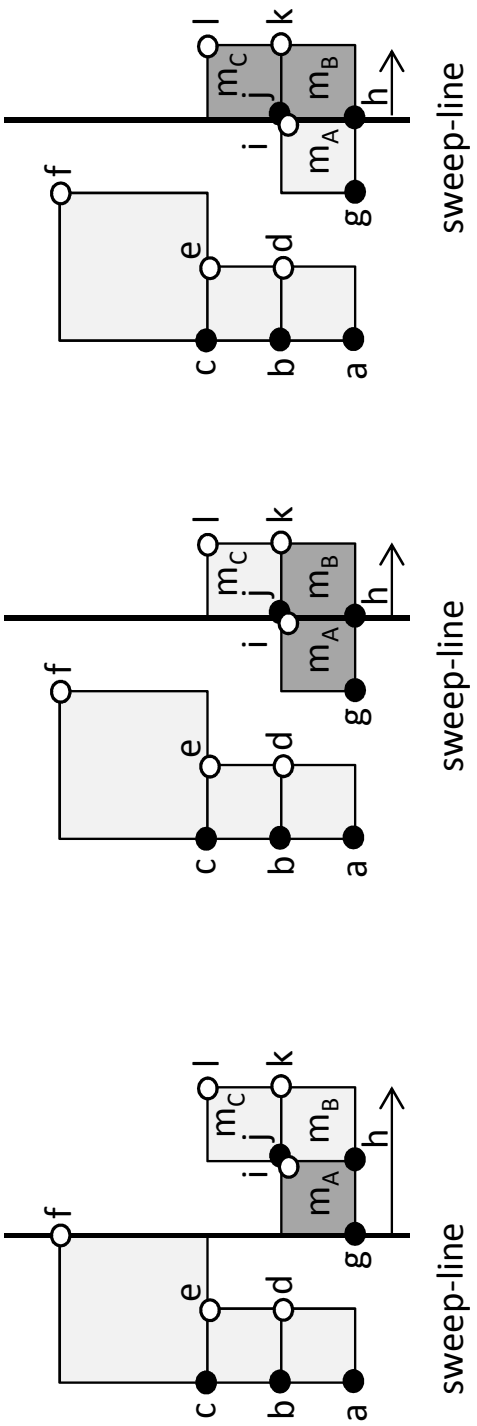


Figure 3.2: Composing transitive closures. Polling a vertex in $Q = \{a, b, \cdots, l\}$ one another, the sweep-line moves from left to right. m_A is activated at the beginning (left). Polling h, activates m_B, which is found to share a border with m_A (middle). Polling i, then deactivates m_A. Next, polling j, activates m_C, which is found to share a border with m_B (right).

3.4 Experimental Results and Applications.

This section presents experimental results for both artificial and realistic data sets to demonstrate the performance of our algorithm and the goodness of the areas discovered. Then, two example applications are presented. All the experiments were run on a Core i5-680 3.60GHz machine with 4GB of RAM running Windows.

3.4.1 Experiments.

An experiment was done using artificial data to compare the performance of our DISCOVERMAXIMALCLOSURE algorithm to that of Rastogi's algorithm. Rastogi's algorithm was selected for comparison because it is the only one among the related ones that can discover multiple areas with mathematically sound definitions for the areas it discovers, as mentioned in Sec. 3.1.

The artificial data comprised N uniformly distributed points, either enabled or disabled in accordance with a two-dimensional Gaussian mixture distribution of the K components:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k), \text{ where } \mathbf{x}, \mu_k \in R^2 \text{ for } k = 1, \cdots, K. \tag{3.6}$$

Let $\pi_k = 1/K$ and $\Sigma_k = \mathrm{diag}(\sigma, \sigma)$ for simplicity, and let μ_k be determined randomly for $k = 1, \cdots, K$. To obtain an artificial data point, each point $\mathbf{x}$ was generated uniformly and then enabled if $p(\mathbf{x}) > 1/K$ or disabled otherwise by random processes.

Rastogi's and our algorithms are used to discover areas containing as much enabled points as possible while eliminating disabled points. This is achieved by discovering the areas that maximize the confidence of the rule:

$$C_1 \rightarrow C_2, \text{ where } \begin{cases} C_1 & : \textit{True}, \\ C_2 & : \textit{True} \text{ if the point is enabled, } \textit{False} \text{ otherwise} \end{cases}$$

Rastogi's algorithm requires three parameters, $n, M,$ and *minSup*, and that the entire rectangle be evenly separated into $\sqrt{n} \times \sqrt{n}$ grids. It can then discover a set of

maximally confident M regions that have no less support than *minSup*. It is reasonable to set the parameters M to K.

Our DISCOVERMAXIMALCLOSURE algorithm, on the other hand, requires one parameter, Z. Although it adaptively separates the entire rectangle in accordance with the given data by using the BUILDQUADTREE function, we used $\sqrt{n} \times \sqrt{n}$ grids to enable us to compare performance. This function discovers locally maximal transitive closures each of which has a support no less than Z.

Figure 3.3 shows the regions discovered by the two algorithms with $\sqrt{n} = 16$, as well as the black (enabled) or white (disabled) points generated with $K = 3$ and $N = 1000$. The entire rectangle was about $15.9km \times 19.6km$, and variance σ was set to $2.0km$.

We compared the processing times for $N = 10,000$, $K \in \{1, 2, \cdots, 10\}$, and $\sqrt{n} \in \{16, 32, 64\}$. The processing times were invariant with K for every $\sqrt{n}$. As shown in the left graph of Fig. 3.4, our DISCOVERMAXIMALCLOSURE algorithm was about 50 times faster than Rastogi's for $\sqrt{n} = 32$. The right graph of Fig. 3.4 compares the probabilities of capturing enabled points in the discovered regions to evaluate the effectiveness of the two algorithms. The expected probability is defined as $\frac{\sum_{\mathbf{x} \in D} p(\mathbf{x})}{sup(D)}$, where D is a discovered region. The DISCOVERMAXIMALCLOSURE was comparable or better in capturing the Gaussian mixture distribution than Rastogi's algorithm, although we do not understand why Rastogi's algorithm do not perform well in some cases.

Figure 3.6 and 3.7 show the regions discovered by the two algorithms for realistic data that were not distributed uniformly. Fig 3.5 shows those by Fukuda's algorithm for reference. We could not disclose the detail of this data because of some contractual reasons, but it was from taxis as introduced in Sec. 3.4.2. As described above, the DISCOVERMAXIMALCLOSURE algorithm adaptively separates the entire rectangle in accordance with the given data by using the BUILDQUADTREE function while Rastogi's uses fixed grid-like meshes. Our algorithm is better for non-uniformly distributed

data points because it discovers well-limited and high confidence regions. Rastogi's algorithm, on the other hand, would need to use finer grid-like meshes and take more time to discover such well-limited regions.

3.4.2 Application: Taxi Fleet Control.

Based on the results shown in Fig. 3.5, 3.6 and 3.7, one feasible application is taxi fleet control based on expected demand. The points indicate the locations of the taxis. Herein, the taxis regularly report their location every minute or so, and whether or not they were available. Let the data points be:

$$(id, time, x, y, pflag, cflag),$$

where *cflag* or *pflag* is *True* if the taxi identified by *id* is unavailable in the current time step or was unavailable in the previous one, respectively, and *False* otherwise.

Optimizing the confidence of the association rule given below would increase the probability of taxis picking up passengers in the discovered regions because there should be more demand for taxis and fewer available ones around:

$$C_1 \;:\; pflag = False,$$
$$C_2 \;:\; pflag = False \text{ and } cflag = True.$$

In other words, a point is enabled when a taxi picks up a passenger, otherwise, it is disabled. Note that taxis unavailable in the previous time step are not taken into consideration. We believe our method is more effective than the alternatives because the points are distributed ununiformly, as can be seen in the figure. Knowing the maximal regions would increase the probability of taxis picking up passengers in the discovered regions because there should be greater demand for taxis and fewer taxis available. The quick processing speed of our algorithm enables taxis to obtain in realtime the locations of areas where they are more likely to pick up passengers without competing with other ones.

3.4.3 Application: Discovery of Sights from Social Messages.

Given the popularity of social network services (SNSs), another application is discovering sights to see on the basis of location data and/or photos attached to text messages. People could identify attractive places to visit by discovering areas that maximize the confidence given by the following rule:

C_1 : *True* if message has location data, *False* otherwise,

C_2 : *True* if message has both location data and a photo, *False* otherwise.

The gray areas in Fig. 3.8 and 3.9 represent attractive areas as determined from the confidence. Those in Okinawa include capes and beaches and several popular sights, like Shuri Castle, an aquarium, and memorial parks around Naha City. Those in Yokosuka include the aquarium on Hakkei Island, parks, and museums.

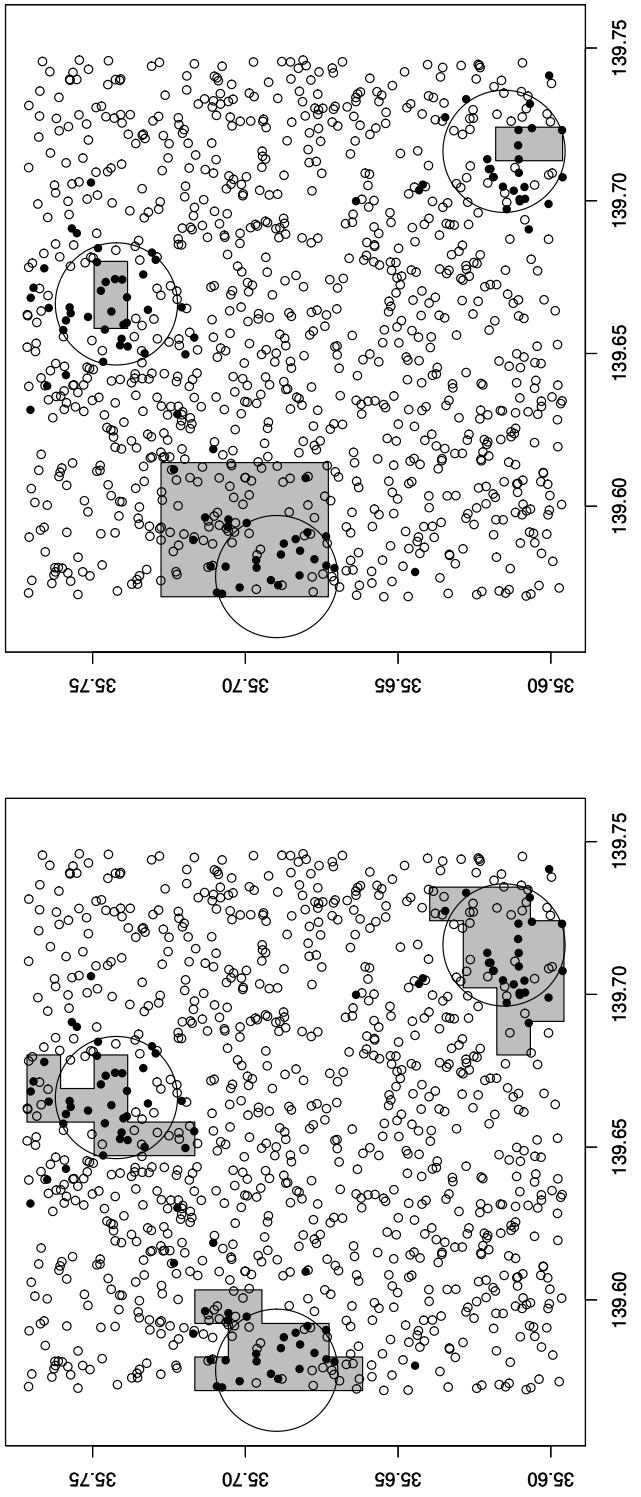


Figure 3.3: Region discovered by DISCOVERMAXIMALCLOSURE algorithm (left) and by Rastogi's algorithm (right) for artificial data with $K = 3, N = 1000$ and $\sqrt{n} = 16$. Center and radius of each circle respectively represents mean and standard deviation of two-dimensional Gaussian mixture component.

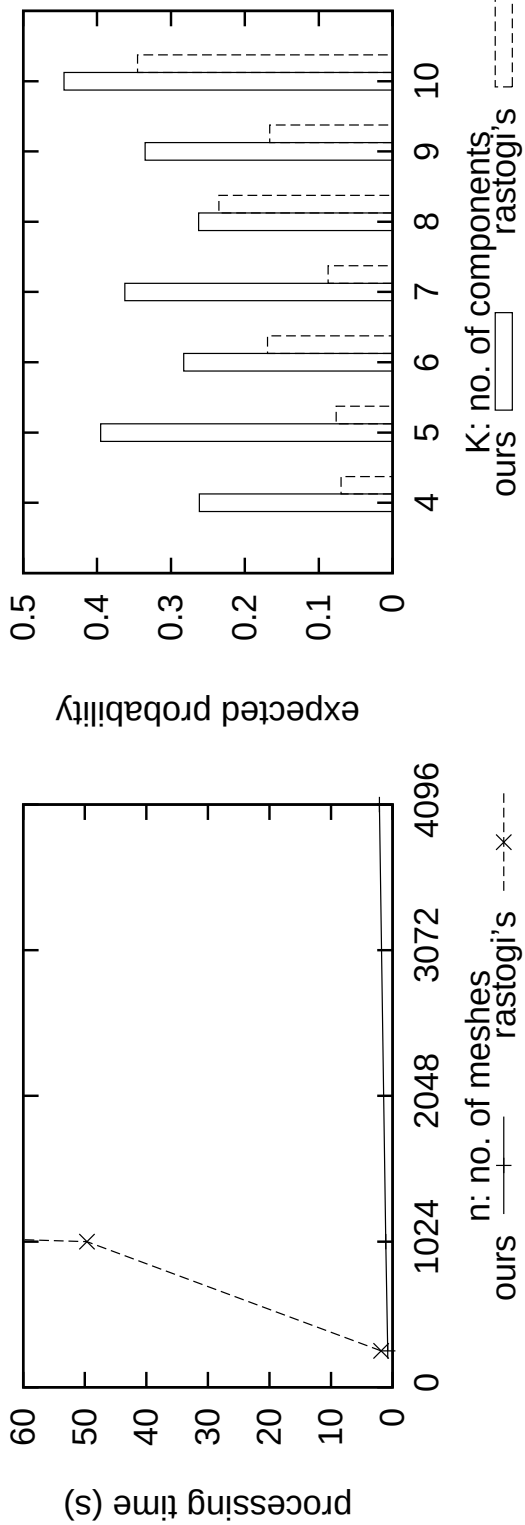


Figure 3.4: Comparison between Rastogi's algorithm and our algorithm. Left graph plots processing times for variable grid size and fixed number of components ($K = 8$), right graph plots probabilities of capturing enabled points within regions.

3.5 Conclusion.

The locally optimized association rule mining has now been formalized. A theorem that holds for locally maximized two-dimensional areas was formulated, and from it an algorithm was derived that efficiently discovers confident areas with shapes that are not limited to being rectangular or x-monotone. It does not require the assumption that position data are uniformly distributed as is often the case in analyzing GPS data from vehicle and smartphones. Experimental results showed that our DISCOVERMAXIMALCLOSURE algorithm was 50 times faster than Rasitogi's, which discovers globally optimized, rectangular regions of interest. Applications include ones where it is essential to discover regions of interest. Its efficiency makes our algorithm well suited for helping taxis to determine where to go and pick up passengers. It can also be used to determine which sights to visit by using the location data and/or photos attached to text messages.

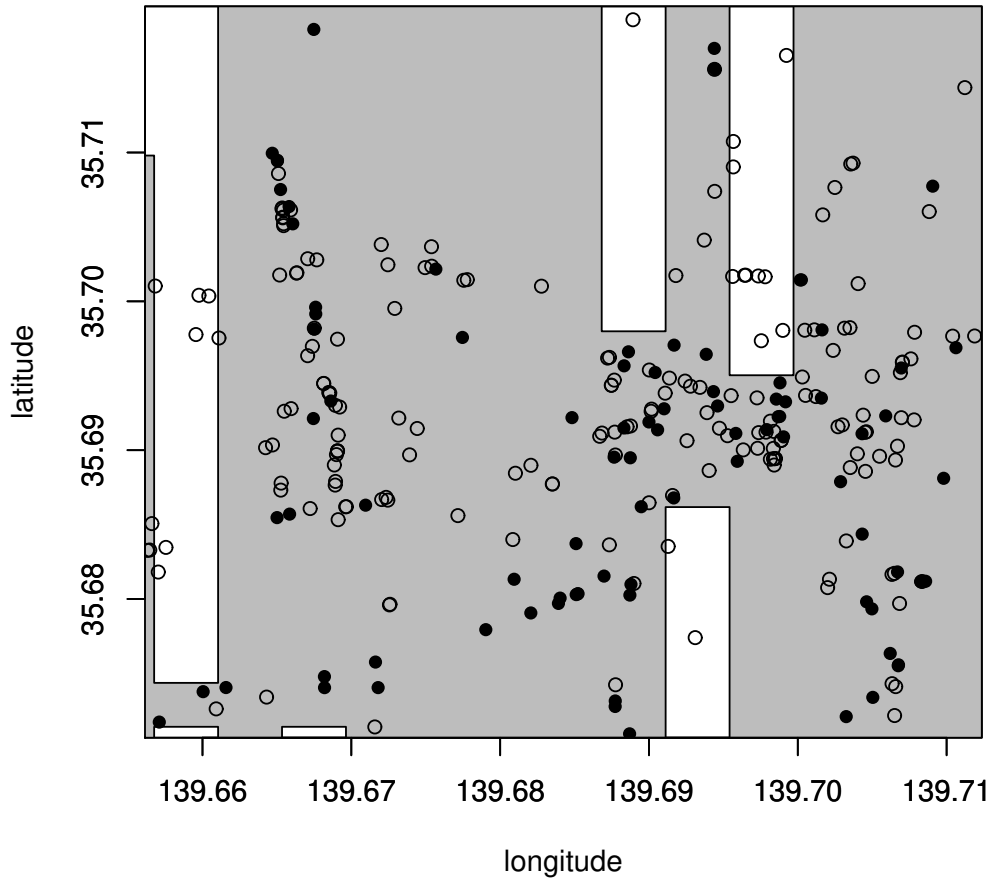


Figure 3.5: Regions discovered by FUKUDA for ununiformly distributed floating car data.

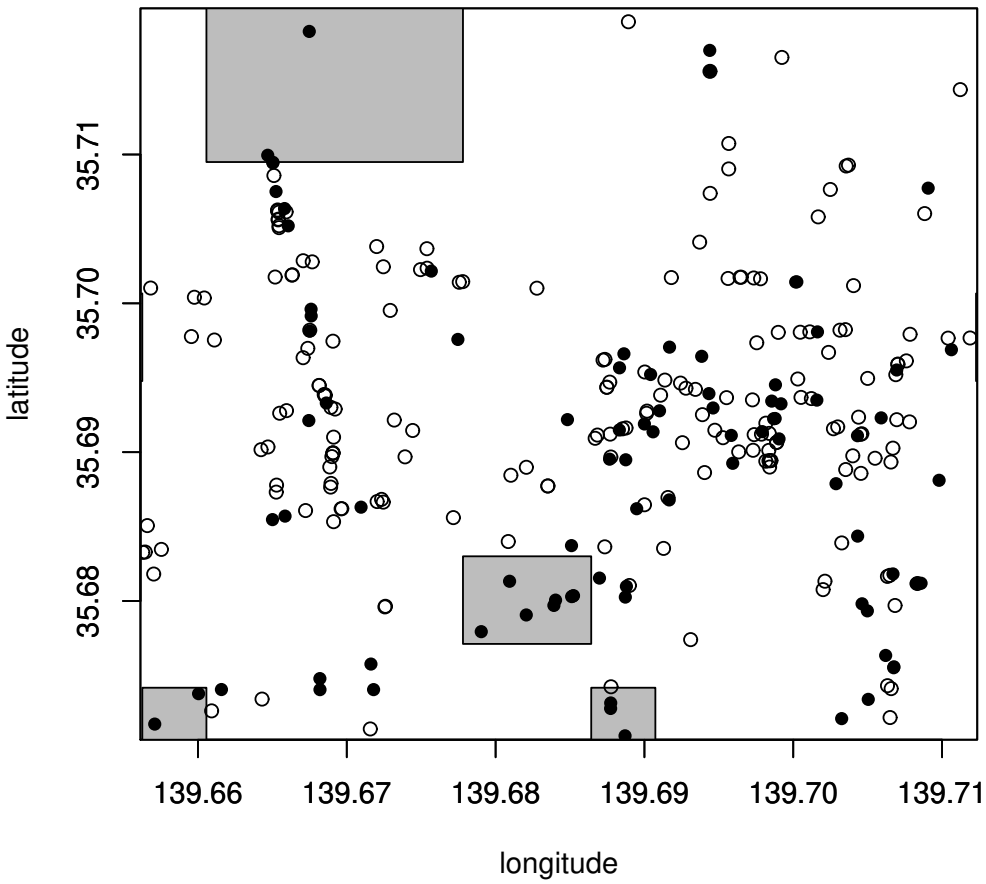


Figure 3.6: Regions discovered by RASTOGI for ununiformly distributed floating car data.

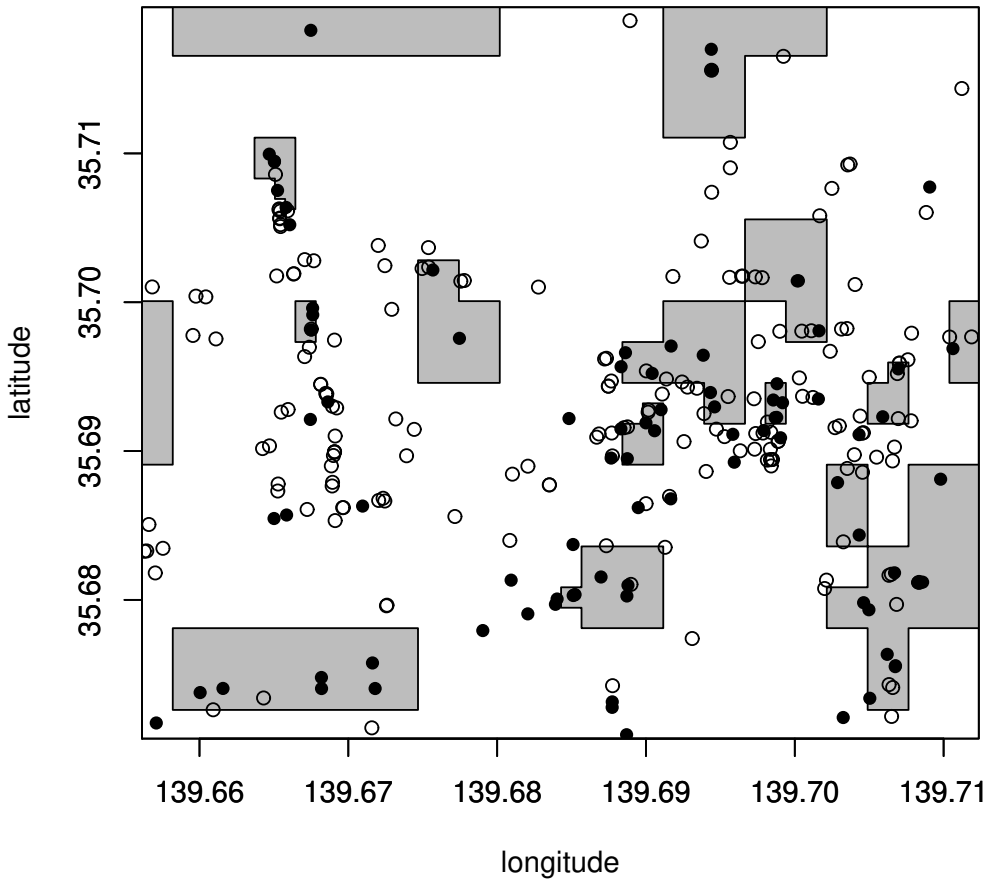


Figure 3.7: Regions discovered by proposed method DISCOVERMAXIMALCLOSURE for ununiformly distributed floating car data.

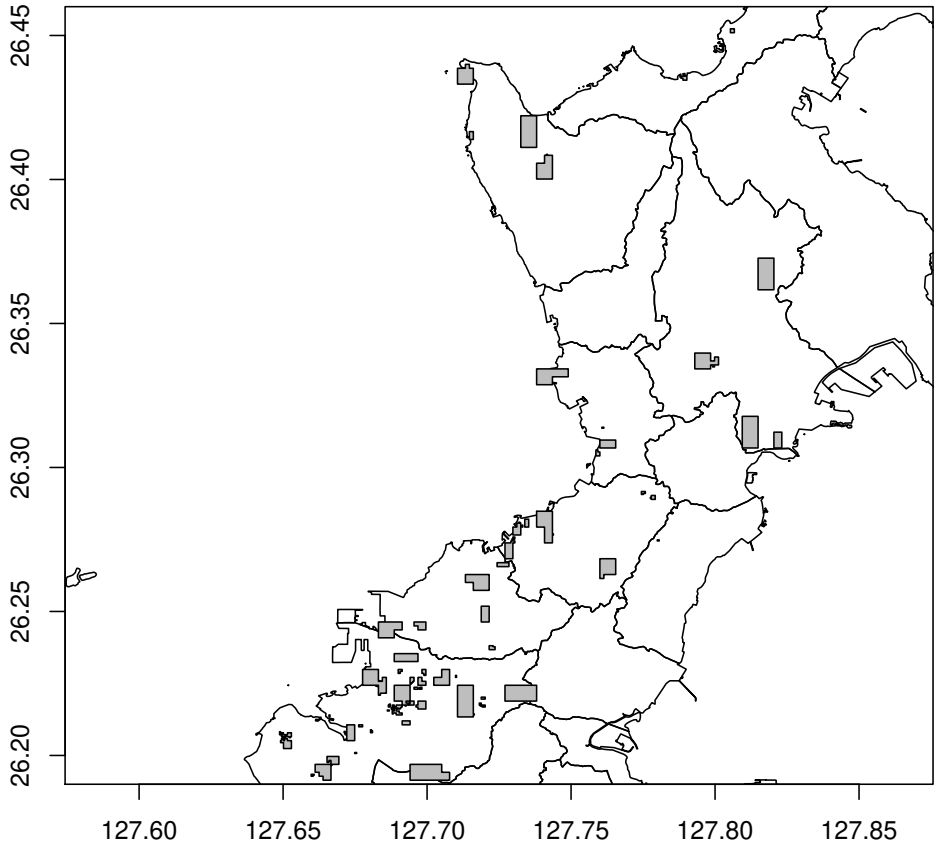


Figure 3.8: Locations in Okinawa where many messages with photos were posted.

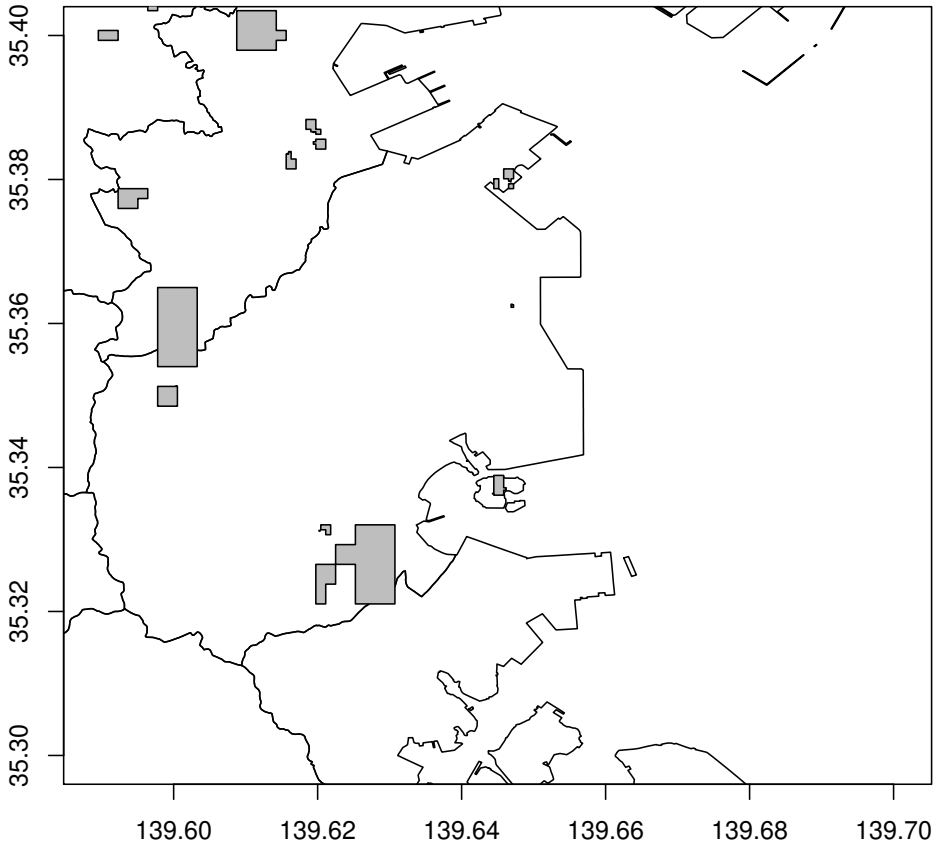


Figure 3.9: Locations in Yokosuka where many messages with photos were posted.

Chapter 4

Regularizing Regions by Description Length

In this chapter, we study a method for regularizing regions by description length.

4.1 Related Works

4.1.1 Cluster detection test

Cluster detection test (CDT) was developed in the fields of epidemiology. A CDT method usually consists of two phases; first, the region with highest likelihood ratio between the inside and the outside of that region is discovered by means of standard mortality rate represented using binomial or Poisson distributions. Then, a statistical test is performed to confirm whether the data inside and outside of the discovered region follow different probability distributions. Kulldorff et al.[46] proposed this framework and employed the circular hypothesis space. Their method has been utilized frequently in epidemiological studies up to the present day. In some cases, however, diseases spread along winds or rivers, and for this reason, a more flexible hypothesis

space than the circular one is required. Tango et al. [65, 67] proposed a hypothesis set consisting of transitively connected segments in administrative districts. Both Kulldorff and Tango's methods naively enumerated connected regions, while other studies employed simulated annealing [23, 24], upper-level set [60], or echelon structure [40] for enhancing computational efficiency. These methods, however, consider the empirical average response and consequently execute a statistical test to validate the significance of the discovered region.

4.1.2 Bump hunting and bichromatic discrepancy

When we are interested in the regions with high response rates, it is easier to obtain them directly rather than estimating the joint probability of variables followed by discovering the regions using the estimated probability. Friedman et al. [26] studied the maximization of averaged response in an axis-parallel rectangular hypothesis from point data. Patient rule induction method [26, 37] iteratively peels either lower or higher end of the variables' domain one at a time until the averaged response of the remaining rectangular region stops decreasing and then performs pasting to ensure that the rectangular region does not overfit the concrete training data. This procedure is a type of heuristic and does not find the exactly optimized solution.

Dobkin et al. [22] studied bichromatic discrepancy of the axis-parallel rectangular hypothesis in the context of machine learning. Bichromatic discrepancy is the number of differences between positive and negative examples. They proposed an algorithm to discover the region with the maximum bichromatic discrepancy by employing its monotonous property. Notably, this is equivalent to the minimization of disagreement between true and predictive class by definition, as well as the optimized gain association rule mining. Agrawal et al. [2] proposed an algorithm to maximize the likelihood ratio between the inside and the outside of a domain by using a technique similar to that proposed by Dobkin.

4.1.3 Optimized association rule mining

Optimized association rule mining was first introduced by Fukuda et al. They formulated the discovery of association rules for a single numeric attribute [30] and two-dimensional numeric attributes [28]. The goal is to obtain optimized instantiations in terms of support, confidence, or gain of the rule, given an uninstantiated association rule. A concrete method first aggregates the measurements and baselines for every cell of an equal-sized grid and then discovers the regions with the highest score among a certain hypothesis set such as a rectangle. Their algorithms include one for a x-monotone region that maximizes the gain achieved by using dynamic programming with fast matrix search, one for a single rectangular region that maximizes gain, and approximation algorithms for a rectangular region that maximizes support or confidence.

Rastogi and Shim generalized the optimized association rule mining [63] such that it has the disjunctive normal form over either categorical or numeric attributes, and they showed that the problem is NP-hard. They proposed a branch-and-bound algorithm to prune the search space as well as enumerating instantiations of numeric attributes before starting processing so that the identified instantiations do not overlap each other.

Neill et al. [57] proposed the branch-and-bound algorithm over an overwrap k-dimentional tree to discover exact solutions of extended densities of regions in a two-dimensional plane.

4.2 Proposed Method

4.2.1 Problem description

Using the preliminaries introduced in Chapter 2, the optimization problem is formalized as follows:

Problem 1 *(Maximization problem of spatial cluster) Given a point dataset S, a mesh*

set M, an object function f, and a hypothesis set H, which is a transitive closure of cells, find $D \in H$ to maximize $f(D)$.

We employ the expected average response as the object function f, as given below.

$$E[L \mid \mathbf{X} \in D] = \int lp(l \mid \mathbf{x} \in D)\mathrm{d}l \tag{4.1}$$

$$= \frac{\int lp(\mathbf{x} \in D, l)\mathrm{d}l}{p(\mathbf{x} \in D)} \tag{4.2}$$

$$= \frac{\iint_D lp(\mathbf{x}, l)\mathrm{d}l\mathrm{d}\mathbf{x}}{\int_D p(\mathbf{x})\mathrm{d}\mathbf{x}} \tag{4.3}$$

$$\sim \frac{\{\sum_{\mathbf{x}_i \in D} l_i\}/n}{\{\sum_{\mathbf{x}_i \in D} 1\}/n}. \tag{4.4}$$

At the fourth equality, we estimate either the joint or marginal probabilities, respectively, $p(\mathbf{x}, y)$ and $p(\mathbf{x})$, from the examples. We call this final expression *average response* hereinafter.

The average response is equivalent to the confidence of an association rule. We will show this as follows. An association rule is of the form $C_1 \rightarrow C_2$ where $C_i (i = 1, 2)$ are logical conditions on $\mathbf{X}$ and L. Their Boolean values are fixed by instantiating $\mathbf{X}$ and L with some coordinates and a response, although we omit either $\mathbf{X}$ or L for simplicity. By letting the support of condition, denoted as $sup(C_i)$, be the number of records in D satisfying C_i, the support and confidence of the rule are defined as follows:

$$sup(C_1 \rightarrow C_2) = sup(C_1), \tag{4.5}$$

$$conf(C_1 \rightarrow C_2) = \frac{sup(C_1 \wedge C_2)}{sup(C_1)}. \tag{4.6}$$

A rule is *confident* if its confidence is not less than a given confidence threshold θ. A rule is *ample* if its support is not less than a given support threshold Z.

Let C_1 be *True* for all $\mathbf{x}_i \in D$ and C_2 be *True iff* $l_i = 1$. Then, we obtain $sup(C_1) = \sum_{\mathbf{x}_i \in D} 1$ and $sup(C_1 \wedge C_2) = \sum_{\mathbf{x}_i \in D} l_i$. Obviously, they are equivalent to the denominator and numerator of (4.4), respectively. Hence, we interchangeably use the expected average response and confidence.

Let us go revert to the object function. As mentioned earlier, we employ transitive closures, $\mathcal{R}^+$, as the hypothesis to discover hotspots from neither uniform nor isotropic data. This requires us to avoid overfitting because the hypothesis is highly complex. Therefore, the object functions should include a regularization term as follows:

$$f(D) = \frac{\sum_{\mathbf{x}_i \in D} l_i}{\sum_{\mathbf{x}_i \in D}} - \lambda \|D\| \tag{4.7}$$

where $\|D\|$ is the description length which will be defined in Sec.4.2.2, and λ is a hyper parameter.

Obviously, it is difficult to obtain the optimized solution explicitly. Instead, our algorithms first obtain an initial solution and then perform a gradient descent. We already studied the technique to obtain the initial solution by using the proved maximal property of regions in Sec.3.2. In Sec.4.2.2, we propose the concrete description length used as the regularization term. Thereafter, in Sec.4.3 we explain the gradient descent based algorithms.

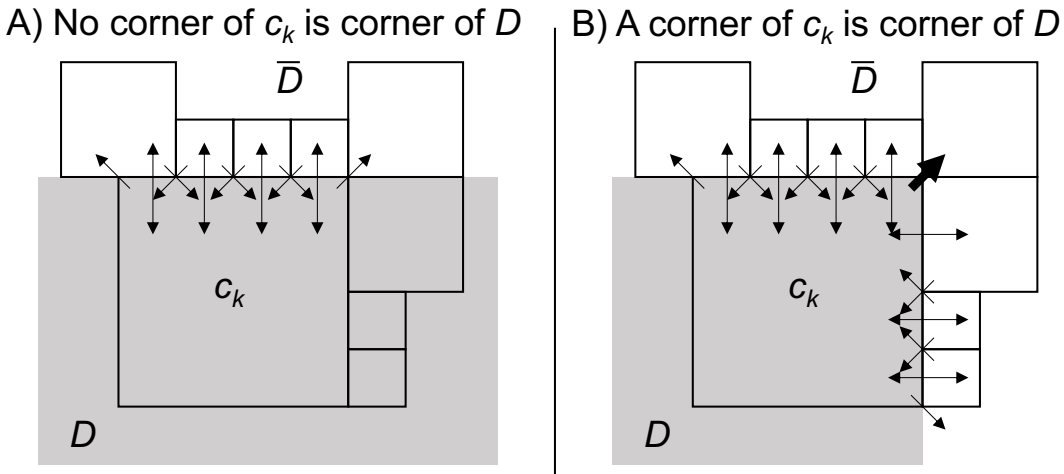
4.2.2 Regularization

This section discusses the regularization term of object function (4.7) that penalizes the region by its complexity on shape. It is well understood that the VC-dimension of convex polygons is related to the number of their edges, or equivalently, number of their corners [54]. We employ this measure as the regularization term denoted by $\|D\|$ and explain how to evaluate it.

Lemma 2 *The number of corners of transitive closure $D \in \mathcal{R}^+$ is given as follows:*

$$(\mathbf{1} - \mathbf{x}_D)^{\mathrm{T}} W \mathbf{x}_D \text{ where } W = 2U - V \tag{4.8}$$

where $\mathbf{1}$ is an m-dimensional column vector with all elements equaling 1, and $U =$



Equation: description		Symbol	A	B
$(1\text{-}x_D)^T U x_D$: number of shared edges on border between c_k and cells in D			4	7
$(1\text{-}x_D)^T V x_D$: number of shared corners on border between c_k and cells in D	outgoing		2	3
	incoming		6	10
	Σ		8	13
$(1\text{-}x_D)^T W x_D = (1\text{-}x_D)^T (2U\text{-}V) x_D$			0	1

Figure 4.1: Counting the corners of D. Vertical and horizontal arrows represent U, and diagonal arrows represent V.

$(u_{jk})_{j=1,k=1}^{m,m}$, $V = (v_{jk})_{j=1,k=1}^{m,m}$ *represent* $m \times m$ *matrices, such that*

$$u_{jk} = \begin{cases} 1 & \textit{if } c_j \textit{ and } c_k \textit{ share their borders,} \\ 0 & \textit{otherwise.} \end{cases} \tag{4.9}$$

$$v_{jk} = \begin{cases} 2 & \textit{two corners of } c_j \textit{ are on a edge of } c_k, \\ 1 & \textit{one corner of } c_j \textit{ is on a edge of } c_k, \\ 0 & \textit{otherwise.} \end{cases} \tag{4.10}$$

Proof 2 *Equation* (4.8) *is decomposed as*

$$(\mathbf{1} - \mathbf{x}_D)^{\mathrm{T}} W \mathbf{x}_D = \sum_{c_j, c_k \in D} (1 - x_{Dj}) w_{jk} x_{Dk} \tag{4.11}$$

$$+ \sum_{c_j, c_k \in \overline{D}} (1 - x_{Dj}) w_{jk} x_{Dk} \tag{4.12}$$

$$+ \sum_{otherwise} (1 - x_{Dj}) w_{jk} x_{Dk}. \tag{4.13}$$

Equation (4.11) *represents the case in which both* c_k *and* c_j *are in* D*. In this case, this term is always 0 because* $1 - x_{Dj} = 0$*. Similarly,* (4.12) *is always 0 because* $x_{Dk} = 0$*. Therefore, we only need to investigate* (4.13)*, where either* c_k *or* c_j *is in* D*, whereas the other is not. Without loss of generality, we have* $c_k \in D$ *and* $c_j \in \overline{D}$*.*

We provide a sketch of the proof. Twice the number of shared edges between c_k *and* $\overline{D}$ *equals the number of corners on the border between* c_k *and the cells in* $\overline{D}$ *if no corner of* c_k *is a corner of* D *(left column in 4.1). The former number differs from the later by the number of* c_k*'s corners that are also those of* D *as well (right column in 4.1). With this in mind, note that the* j*-th element of* $U\mathbf{x}_D$ *is the number of the edges that* c_j *shares with the region* D *and that accordingly,* $(\mathbf{1} - \mathbf{x}_D)^{\mathrm{T}} U \mathbf{x}_D$ *gives the number of edges shared between* D *and* $\overline{D}$*. Similarly, note that the* k*-th element of* $(\mathbf{1} - \mathbf{x}_D)^{\mathrm{T}} V$ *is the number of corners on the border between* c_k *and the region* $\overline{D}$ *and that accordingly,* $(\mathbf{1} - \mathbf{x}_D)^{\mathrm{T}} V \mathbf{x}_D$ *gives the number of corners on the border between* D *and* $\overline{D}$*. Consequently,* (4.8) *was proved to give the number of corners in* D*.*

The matrices U and V were obtained by employing sweep-line algorithms in a manner similar to that used for evaluating transitive closure from a collection of cells. For detail, refer to [39].

4.3 Algorithms

This section explains the procedures used to discover the optimized hypothesis (Algorithm 3). The proposed method involves building a quad-tree comprising irregularly sized cells (line 2), peeling the cells irrelevant to the maximal regions (line 3), and pasting cells by running the gradient descent method to regularize the regions(line 4). The notations are the same as those used in Sec.4.2. These three processes are detailed in the following sections.

4.3.1 Data adaptive mesh forming

Lines 5–14 describe the procedure to build a quad-tree. Because the cells are split iteratively into four along the horizontal and vertical borders so long as they have more points than the threshold α (line 6), the cells in the final quad-tree are expected to have roughly the same number of points as that in the fixed mesh of size m. In the case where the points are distributed uniformly in the two-dimensional space, the quad-tree converges to a grid mesh as more points become available. It is easy to obtain $\mathbf{h}$ and $\mathbf{s}$ by counting the points in cells during quad-tree formation (line 14).

4.3.2 Peeling

Lines 15–24 describe the peeling procedure. All maximal regions G are reported by iteratively removing the irrelevant cells in accordance with Thm.1 until no peeling occurs at line 20. The process for obtaining new transitive closures (line 22) is implemented

efficiently by using the sweep-line algorithm [20, 39]. **h** and **s** are needed at lines 20 and 23.

4.3.3 Pasting

Lines 25–31 describe the pasting procedure. First, we obtain the matrices $W = 2U - V$ needed to evaluate the regularization term $\|D\|$ based on their definitions in Sec.4.2.2. The process of efficient implementation is omitted because it is obvious as we presented the sweep-line algorithm over quad-tree in Sec.3.3. Second, we initialize x_{Dj} a positive number if c_j is in one of the regions in G, or a negative number otherwise (line 27). Finally, we run the gradient descent method (line 28–30). To make the object function differentiable, we replace $\mathbf{x}_D$ with its continuous version as explained in Sec.2.1.2. With this trick, the gradient is given as follows:

$$\begin{aligned} \nabla_{\mathbf{z}_D} f(\mathbf{x}_D) &= \left(\frac{\partial f}{\partial x_{Dj}} \frac{\partial x_{Dj}}{\partial z_{Dj}} \right)_{j=1}^{m} \\ &= \left(x_{Dj}(1 - x_{Dj}) \frac{\partial f}{\partial x_{Dj}} \right)_{j=1}^{m}. \end{aligned} \tag{4.14}$$

4.4 Experimental Results

This section presents the experimental results showing that the proposed method captures the probability $p(\mathbf{X}, L)$ well and achieves a higher average response by using synthetic data. We compared the method with and without regularization, denoted as MTCR and MTC, respectively. MTC stands for "Maximized Transitive Closure". In addition, we present the results obtained by RASTOGI and FUKUDA. Note that RASTOGI, FUKUDA, and MTC can represent CDT introduced in Sec.4.1.1 because they are supposed to output the similar regions as CDT in the following experimental settings.

Table 4.1: Parameters of synthetic data.

Name	Values	Description
n	4096	#data
K	{ 1,2, $\cdots$, 9 }	#Gaussian components
m	64×64	#segments
F	{ 0.2, 0.4, 0.6, 0.8, 1.0}	flatness of components
q	0.65	probability to split cells

4.4.1 Generation of synthetic data

The synthetic data is used to emulate spatially isotropic or non-isotropic cases by making the coordinate distribution $p(\mathbf{X})$ uniform or ununiform, as well as by making the response probability $p(L \mid \mathbf{X})$ round or squashed, respectively. Table 4.1 shows the parameters needed to achieve this.

In the uniform case, the synthetic $p(\mathbf{X})$ follows an uniform distribution. In the ununiform case, on the contrary, a quad-tree is employed to determine the distribution of coordinates $p(\mathbf{X}) = p(\mathbf{X} \mid c_j)p(c_j)$. It is built by letting cells split with the probability q^d, where d is the depth of quad-tree. $p(\mathbf{X} \mid c_j)$ and $p(c_j)$ follow individual uniform distributions. This implies that the cells have the same number of coordinates, either large or small, which makes the coordinates sparse or dense from place to place. Note that the parameter q was determined such that the average number of cells becomes n for ensuring fair comparison between the ununiform and uniform cases.

The response probability $p(L \mid \mathbf{X})$ follows mixed Gaussian probability with K components whose mixture coefficients are equal. Their covariance matrices are elliptic, and the flatness of these components is F, but the angles of these components are determined randomly and differ from each other.

Once the joint probability $p(\mathbf{X}, L) = p(L \mid \mathbf{X})p(\mathbf{X})$ is fixed, we generate n coordinates with their responses such that they follow the joint probability using a random

process.

4.4.2 Experimental configurations

Simply evaluating the average responses for the data from which the regions are discovered is not sufficient for demonstrating the aim of the present study. Instead, we must evaluate the expected average responses. To this end, we designed the following experimental configurations: first, we generated 11 independent datasets by using the procedure explained in Sec.4.4.1. Then, by using one of these datasets and the algorithms, we discover the maximal regions with the parameter m, which determines the fineness of the grid or quad-tree mesh these algorithms employ. Finally, we evaluate the precision of discovered regions for the remaining 10 datasets. This test process is repeated on 20 differently parameterized probabilities and the average of these trials is calculated.

4.4.3 Results

1. Uniform or ununiform distributions: we first compared each algorithm between the uniform and the ununiform cases. The response was fixed to be round (F=1.0) to eliminate factors other than uniformity. We investigated both unimodal (K=1) and multimodal (K=5) cases (upper in Tab.4.2). With this experiment, we expect that quad-tree meshes benefit in the ununiform cases by taking dense and sparse areas fairly. The precision of MTC and MTCR improved significantly by around 10% in the unimodal cases, and by 1.5%–3.0% in the multimodal cases. Naturally, the improvement was lower in the multimodal cases because the number of coordinates per component was lower than that in the unimodal cases. Although RASTOGI enjoyed its highest precision in the unimodal and uniform cases, MTCR was superior in the ununiform cases and comparable to FUKUDA in the uniform cases.

2. Round or squashed responses: next, we compare each algorithm with the squashed and round responses. To eliminate factors other than the flatness, the responses were fixed as unimodal (K=1). We investigated both uniform and ununiform cases (lower in Tab.4.2). With this experiment, we expect that the hypothesis of transitive closure captures regions with high expected average response even when the response probability is squashed. The precision did not change significantly between the squashed and the round cases with both MTC and MTCR, while it decreased by 7.21%–9.78% when using RASTOGI. Notably, FUKUDA did not degrade because the x-monotone hypothesis can also represent squashed regions.

3. Advantage of regularization: thus far, our methods, MTC and MTCR, advantageous in the case with non-isotropic distribution.

   Finally, we show the improvement resulting from regularization by comparing the algorithms in the case of a non-isotropic distribution with variable flatness in Fig.4.2. The improvement rate made by MTCR from MTC were 11.6%–23.0%.

Algorithm 3 Proposed method

1: **procedure** MAIN(S, U, m, Z)
2: $M, \mathbf{h}, \mathbf{s} \leftarrow$ BUILDQUADTREE(S, U, m)
3: $G \leftarrow$ PEELING($M, \mathbf{h}, \mathbf{s}, Z$)
4: **return** PASTING(G, $\mathbf{h}$, $\mathbf{s}$) as $\widetilde{D}$
5: **procedure** BUILDQUADTREE(S, U, m)
6: $\alpha \leftarrow |S|/m$
7: initialize queue Q by $\{U\}$, M by empty set
8: **while** Q is not empty **do**
9: pop cell from Q as U
10: **if** $sup(U) > \alpha$ **then**
11: split U into four cells and push them to Q
12: **else**
13: append U to M
14: **return** M as well as its $\mathbf{h}$ and $\mathbf{s}$
15: **procedure** PEELING(M, $\mathbf{h}$, $\mathbf{s}$, Z)
16: initialize maximal regions, G, by $\{M\}$
17: **repeat**
18: initialize M by empty set
19: **for all** $D \in G$ **do**
20: peel irrelevant cells from D by using (3.3)
21: add cells in D to M
22: initialize G by closures discovered in M
23: **until** $sup(M) < Z$ or no cell peeled at line 20
24: **return** G
25: **procedure** PASTING(M, G, $\mathbf{h}$, $\mathbf{s}$)
26: initialize W from M
27: initialize $\mathbf{z}_D$ in accordance with G
28: **repeat**
29: $\mathbf{z}_D \leftarrow \mathbf{z}_D - \eta \nabla_{\mathbf{z}_D} f(\mathbf{x}_D))$
30: **until** $\mathbf{z}_D$ converges
31: **return** $\{c_j \mid z_{Dj} > 0\}$

Table 4.2: Precision of four algorithms which compare the ununiform to uniform cases (upper) and squashed to round cases (lower). MTC and MTCR are the proposed methods, which are without and with regularization, respectively.

	unimodal(K=1)				multimodal(K=5)			
	MTC	MTCR	FUKUDA	RASTOGI	MTC	MTCR	FUKUDA	RASTOGI
uniform	25.80	27.43	28.76	34.18	7.37	8.14	8.47	6.68
ununiform	36.56	38.22	30.53	36.22	8.89	11.16	8.97	7.14
lift	10.76	10.79	1.77	2.04	1.52	3.02	0.50	0.46

	uniform				ununiform			
	MTC	MTCR	FUKUDA	RASTOGI	MTC	MTCR	FUKUDA	RASTOGI
round (F=1.0)	25.80	27.43	28.76	34.18	36.58	38.22	30.53	36.22
squashed (F=0.2)	29.16	29.01	30.41	26.97	37.70	37.64	28.64	26.44
lift	3.36	1.58	1.65	-7.21	1.12	-0.58	-1.89	-9.78

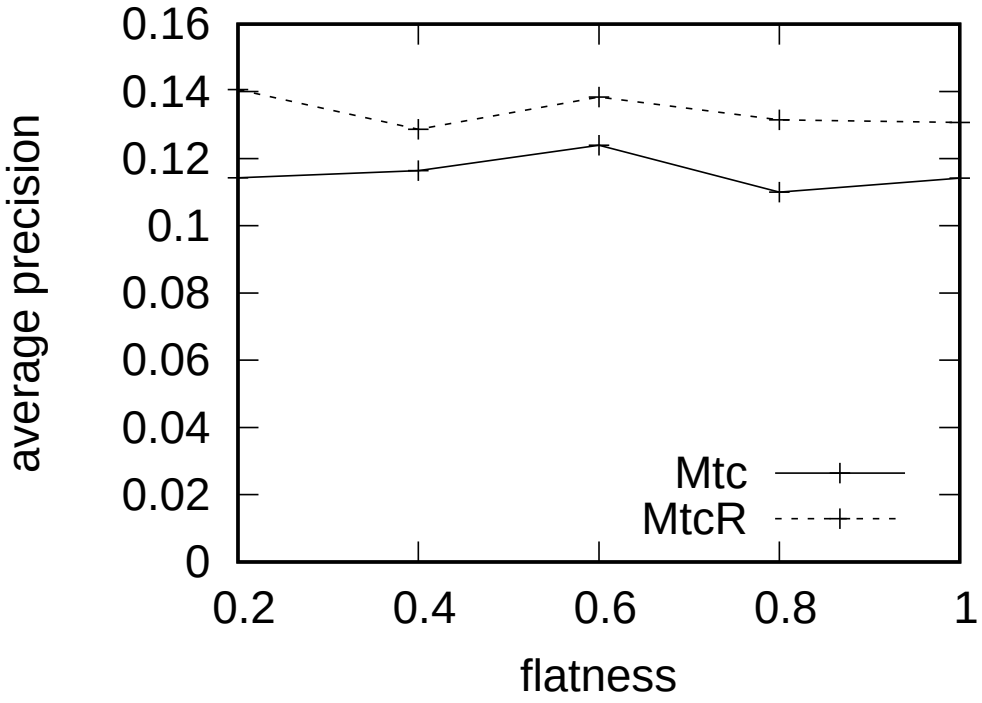


Figure 4.2: Precision of MTC and MTCR with K=5 and variable flatness.

4.5 Conclusion

We proposed a new approach to maximize the expected average response. In this method, we introduced a regularization term that penalizes the complex regions and relaxes the assumptions related to coordinate distribution. These enhancements made our method adaptive to observed data while improving its predictive performance. The experiments showed that our algorithms did not decrease the precision in the case of unseen data, even when it was neither uniform nor isotropic. In addition, they showed that the regularization improved precision by more than 20%.

Chapter 5

Maximizing Posterior Joint Map-matching

In this chapter, we study a method for maximizing posterior joint map-matching.

5.1 Related Works

On-line or local map-matching methods attach a newly observed GPS point to one of the neighboring links in a DRM. These methods use coordinates, direction, and speed localized to the current point to take into account the connectivities of these links [34, 73, 15]. By contrast, off-line or global map-matching methods consider distance between a trajectory and a path in a DRM in a topological sense, from its origin to destination [12]. Alt et al. proposed a map-matching algorithm that utilizes Fréchet distance [4, 3]. Algorithms with the relaxed Fréchet distance have also been proposed [25, 12]. All these approaches map individual trajectories to the nearest paths in accordance with their own policy or distance function.

Due to the constraints on power consumption and transmission cost, trajectories are very sparse, and the above approaches do not always work well with low-sampled

trajectories. To tackle these problems, probabilistic methods estimate the link from which the observation is made [50, 78, 58]. A multi-track map-matching method exploits the ensemble nature buried in the trajectories. It iteratively estimates the order of observations from different trajectories and the most likely segments from which they come [41]. A joint map-matching enumerates fixed-sized segments from a DRM and, using them as variables of the optimization problem, it discovers the paths to which the trajectories are assigned such that they seem to be as natural as routes [48]. This method, however, requires hyper parameters to balance the features such as the distance to the segments, stitching of segments, and regularity of the solution. Our method falls to this category that tackles the problem with low-sampled trajectories, which is not accessed well by single-track, especially on-line, map-matching methods. Therefore, we would concentrate on multi-track and off-line map-matching methods hereinafter.

Map-generation algorithms have been proposed for overall traffic analysis. Although governments, information companies, and social communities have manually developed DRMs, sometimes roads open or close either permanently or temporarily. To be adaptable, map-generation techniques maintain DRMs with less cost by automatically building them from a huge collection of observed GPS points. Some methods reconstruct DRMs through a series of dedicated procedures [19, 42, 43], and others do so based on Morse theory [72]. Their interests are mainly in building accurate DRMs, not in understanding the traffic flow. A method that consolidates trajectories to form a map considers traffic flows to some extent [13], but it may not work on cases where the trajectories are sparsely sampled.

Many applications have been proposed for the analysis and prediction of traces. Some learn the repeated patterns of a car owner's history, e.g., commuting routes, the dropping off and picking up of family members, and visiting relatives or friends [27]. Turn prediction is another typical application for predicting which direction a car will

take at an intersection, based on the route taken up to this point by learning others' traces [44]. These applications, however, are developed to predict particular purposes.

5.2 Proposed Method

5.2.1 Stochastic generative model

The notations employed hereinafter follow those defined in Sec.2.3. Let $b(s) \in \mathbb{R}^2, 0 \leq s \leq q, s \in \mathbb{R}$ be a route, which is also described as a path $\beta \in \mathbb{P}_G$, such that:

$$b(s) = \begin{cases} v^{(j)} & \text{if } s = j, \\ (\lceil s \rceil - s)v^{(\lfloor s \rfloor)} + (s - \lfloor s \rfloor)v^{(\lceil s \rceil)} & \text{otherwise.} \end{cases} \tag{5.1}$$

Note that, without loss of generality, we attached the origin and destination of the route to the first and last vertices of the path, respectively.

The observations in a trajectory $\alpha \in \mathbb{P}$ are emitted on the route $b(s)$ at $s \in \{s^{(i)} \in \mathbb{R} \mid i = 0, \cdots, p\}$ such that $s^{(i)} < s^{(j)}$ for all $0 \leq i < j \leq p$. Additionally, assuming the first and last observations are made from the origin and destination of the route, respectively, we have $s^{(0)} = 0$ and $s^{(p)} = q$. Each observation has its observation noise $\epsilon_i \in \mathbb{R}^2$ and thus we have:

$$\alpha^{(i)} = b(s^{(i)}) + \epsilon_i. \tag{5.2}$$

Also note that a trajectory has sampling noises that are induced by interpolating the finite number of observations comprising the trajectory.

In summary, although the routes are unobservable, the trajectories are observed and are emitted from one of the routes. We introduce the stochastic generative model with the observed and latent random variable $X, Z \in \mathbb{R}^K$, as follows:

Latent variable Z is a 1-of-K random variable whose realization is $z = (z_k)_{k=1}^K$, where $z_k \in \{0, 1\}$. There is a k^* such that $z_k = 1$ if $k = k^*$ and $z_k = 0$ otherwise,

which represents the occurrence of k^*-th route out of K possible routes. The occurrence follows the prior probability distribution of $P(z) = \prod_{k=1}^{K} \pi_k{}^{z_k}$ such that $\sum_{k=1}^{K} \pi_k = 1$.

Observed variable X is a random variable whose realization is $x = (x_k)_{k=1}^{K}$, which represents the distance between trajectory and path. This distance follows the probability distribution of $P(x|z) = \prod_{k=1}^{K} f(x_k|\sigma)^{z_k}$ where $f(x_k|\sigma) = \sigma \exp(-\sigma x_k)$ is the probability density function of exponential distribution. The parameter σ is determined in accordance with the volume of the sampling noise.

5.2.2 Maximizing posterior probability

A generic EM algorithm maximizes the posterior probability of the stochastic model parameterized by θ. With an initial θ^{old}, it iterates the following E-step and M-step by replacing θ^{old} with θ^{new} until either Q' or θ converges:

E-step updates the conditional probability $P(Z|X, \theta^{\text{old}})$, and

M-step finds the parameter θ^{new} that maximizes the log-likelihood of posterior probability:

$$Q'(\theta, \theta^{\text{old}}) = \sum_{Z} P(Z|X, \theta^{\text{old}}) \ln P(X, Z|\theta) + \ln P(\theta). \tag{5.3}$$

Given a collection of traces $T = \{\alpha_n \mid n = 1, \cdots, N\}$ and if let $x_n = (x_{nk})_{k=1}^{K}$ and $z_n = (z_{nk})_{k=1}^{K}$ be the independent realizations of the random variables X and Z, respectively, we have the concrete E-step and M-step for the joint map-matching by applying the above probability distributions to the generic EM algorithm in a similar manner as that for a Gaussian mixture model [11]:

E-step evaluates the responsibility $\gamma(z_{nk})$ with the parameter π^{old}, and

M-step finds π^{new} that maximizes the log-likelihood of posterior probability $Q'(\pi, \pi^{\text{old}})$

.

The responsibility and the log-likelihood are respectively defined as follows:

$$\gamma(z_{nk}) = \frac{\pi_k f(x_{nk}|\sigma)}{\sum_{k'=1}^{K} \pi_{k'} f(x_{nk}|\sigma)}, \tag{5.4}$$

$$Q'(\pi, \pi^{\text{old}}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \sigma\gamma(z_{nk})x_{nk} + \sum_{k=1}^{K} z_k \ln \pi_k. \tag{5.5}$$

Although there are too many paths on the graph, it is practically sufficient to consider the paths that have shorter distances from each trajectory. This is feasible if we employ an algorithm [64] that can enumerate all the paths whose distances from the trajectory are within a certain threshold, such as σ. In extreme, considering just the minimizing path $\beta_{k^*} = \hat{\beta}_G(\alpha)$, we have $\gamma(z_{nk}) = 1$ if $k = k^*$ and $\gamma(z_{nk}) = 0$ otherwise. Assuming that the prior distribution is uniform, namely $\pi_k = 1/K$ for all k, the second term of Eq.(5.5) is straightforward and equal to $-K$. If we accept that K is proportional to the description length of the geometric graph G, the joint map-matching is equivalent to the minimization problem below.

Definition 5 *(Route graph discovery) Let a hypothesis space of a graph be $\mathbb{G}$, a single-track map-matching be $\mathcal{M}_G$, and a collection of trajectories be T. A graph $G \in \mathbb{G}$ most likely emits the trajectories T if it minimizes the following loss function:*

$$L(G;T) = \sum_{\alpha \in T} \hat{d}_G(\alpha) + \lambda \|G\|, \tag{5.6}$$

where $\|G\|$ is the description length of the graph G, such as the total length of its edges, and $\lambda > 0$ is a hyper parameter.

The first term is for the residual and the second term is for the regularization. This problem is equivalent to single-track map-matchings if λ is zero. Otherwise, some edges are left unused so that $\|G\|$ decreases even though the distance $\hat{d}_G(\alpha)$ becomes longer for some trajectories.

5.2.3 Graph exploration algorithm

To minimize $L(G;T)$, we employ an exploratory search in the graph space, and obtain a decreasing series of graphs $G^{(t-1)} \supset G^{(t)}$ for $t = 1, 2, \cdots$ such that their losses also decreases. Let us denote the output of map-matching $\mathcal{M}_{G^{(t)}}$ as $\hat{\beta}_\alpha^{(t)} = \hat{\beta}_{G^{(t)}}(\alpha)$ and $\hat{d}_\alpha^{(t)} = \hat{d}_{G^{(t)}}(\alpha)$ for short.

Before presenting the important property that drives the exploration, we note that $\hat{d}_\alpha^{(t-1)} \leq \hat{d}_\alpha^{(t)}$ always holds. This is trivial because if there were a path closer to α in $G^{(t)}$, it must be closer to α than the minimizing path in $G^{(t-1)}$ and this is contradictory. We do not care how the map-matching is implemented as long as it satisfies the inequality above.

The following theorem gives the condition to ensure that decreasing series of graphs decrease their losses.

Theorem 2 *Given a collection of trajectories T, and two graphs $G^{(t-1)}$ and $G^{(t)}$, $L(G^{(t)};T) < L(G^{(t-1)};T)$ holds iff the following inequality holds:*

$$\lambda \|\Delta^{(t)}\| > \sum_{\alpha \in T_{|\Delta^{(t)}}} \left\{ \hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)} \right\}, \tag{5.7}$$

where $\Delta^{(t)} = G^{(t-1)} \setminus G^{(t)}$ and $T_{|\Delta^{(t)}}$ denotes the collection of trajectories whose minimizing paths run through $\Delta^{(t)}$.

Proof 3 *By evaluating the difference between losses of the two consecutive graphs in*

the series, we have the following:

$$
\begin{aligned}
& L(G^{(t)};T) - L(G^{(t-1)};T) \\
&= \left(\sum_{\alpha \in T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t)} + \sum_{\alpha \notin T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t)} + \lambda \|G^{(t)}\| \right) \\
&\quad - \left(\sum_{\alpha \in T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t-1)} + \sum_{\alpha \notin T_{|\Delta^{(t)}}} \hat{d}_\alpha^{(t-1)} + \lambda \|G^{(t-1)}\| \right) \\
&= \sum_{\alpha \in T_{|\Delta^{(t)}}} \left(\hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)} \right) - \lambda \|\Delta^{(t)}\|.
\end{aligned}
$$

Note that $\hat{d}_\alpha^{(t)} = \hat{d}_\alpha^{(t-1)}$ *holds for* $\alpha \notin T_{|\Delta^{(t)}}$ *because both* $\hat{\beta}_\alpha^{(t)}$ *and* $\hat{\beta}_\alpha^{(t-1)}$ *are irrelevant with* $\Delta^{(t)}$*. It then follows that, Eq.*(5.7) *holds iff* $L(G^{(t)};T) < L(G^{(t-1)};T)$ *holds.*

Algorithm 4 is the pseudo-code for the route graph discovery. Given a collection of trajectories T and an initial graph $G^{(0)}$, e.g., a DRM, it finds the final graph that minimizes the loss (line 6). Note that B maintains the minimizing path and the minimum distance for each trajectory α throughout every t-th stage (line 17). The main loop (line 2–5) explores a series of subgraphs with decreasing losses as explained in Thm.2. First, an edge e is selected, for instance, in increasing order of the cardinality, which is the number of the minimizing paths running through it (line 3), and a new graph $G^{(t)}$ is obtained by *re-routing* with the edge e disabled (line 4). We explain later what re-routing is, as well as why and how we select a single edge. Then, the graph $G^{(t)}$ is probabilistically accepted or rejected (line 4). Finally, the main loop either continues or breaks in accordance with the history of the obtained graphs (line 5).

Next, we explain how to obtain the new graph $G^{(t)}$ by finding $\Delta^{(t)}$. The re-routing technique serves this by map-matching with some edges of graph $G^{(t)}$ disabled (line 12). Note that trajectories not in $\Delta^{(t)}$ are irrelevant to $\Delta^{(t)}$ and that any edge e in $\Delta^{(t)}$

satisfies the following inequality:

$$g \subseteq \Delta^{(t)} \subseteq \overline{\Delta}_g^{(t)}$$

$$\text{where} \quad \begin{cases} g = G(\{e\}), \\ \overline{\Delta}_g^{(t)} = G^{(t-1)} \setminus \bigcup_{\alpha \notin T_{|g}} G(\hat{\beta}_\alpha^{(t-1)}), \end{cases}$$

and $T_{|g}$is the collection of trajectories whose minimizing paths run through g. Thus, the following strategy works: first conservatively select an edge e from $G^{(t)}$ and optimistically initialize $\Delta^{(t)}$ with $\overline{\Delta}_g^{(t)}$ (line 8), as well as the cumulative differential residual ϵ with 0 (line 9). Then, as we re-route a trajectory in $T_{|g}$, $\Delta^{(t)}$ is subtracted by $G(\hat{\beta}_\alpha^{(t)})$, and ϵ is added by the differential residual before and after the re-routing (line 12–14). In this implementation, we employed Zeheng et al.'s algorithm [78] for re-routing.

The procedure terminates as soon as it becomes obvious that Eq.(5.7) will never be satisfied (line 15,16). This is safe because of Thm.2 and, notably, this saves much computation by skipping unnecessary map-matchings. If no early termination has occurred, the procedure returns with the new reduced graph as $G^{(t)}$ (line 18). We describe four implementation issues in the following sections.

1. Meta-heuristic optimization in main loop: the reason we probabilistically accept the graph at line 4 is to escape local minima. For simplicity, however, the current implementation always accepts the returned graph $G^{(t)}$. Other meta-heuristic algorithms are also applicable, in addition to this implementation.

2. Initial graph: a DRM is one candidate of the initial graph $G^{(0)}$. A Delauney graph whose vertices are GPS observations is another. In the former case, Alg.4 performs a joint map-matching. If it is certain that the GPS traces are from objects moving on a DRM, this is the reasonable option. We chose this option for the sake of experimentation in Sec.5.4. In the latter case, Alg.4 performs a map-generation. This is useful when no DRM is available, although GPS observations should be carefully sampled if the density differs from place to place

over the two-dimensional space. This is because too many observations increase the computation whereas too few observations decrease the accuracy of the route graph.

3. Map-matching algorithm: as mentioned at the beginning of Sec.5.2.3, the route graph discovery employs a map-matching algorithm that follows Def.2. The current implementation employed Zeheng et al. [78] because it is easy to implement. Simply introducing their algorithm, it performs the A* algorithm to find the shortest path between two vertices in the graph. Considering the combinations of vertices, each of which is one of the neighbors of consecutive observations within a window of size w, their algorithm finds the route that minimizes the cost function of the A* algorithm so long as w is large enough. Thus, this algorithm satisfies the requirement of Def.2. Notably, any map-matching algorithm could be used as long as it follows Def.2.

4. Selection of edges: there can be several priorities when selecting a disabled edge:

   (a) by the length of edge,

   (b) by the cardinality of edge,

   (c) by both the length and cardinality,

   (d) by the size of $\Delta^{(t)}$ for the edge, and

   (e) at random.

   Except for the fifth option, these share the idea of first selecting an edge that is most unlikely to remain in the final graph. Although the fourth option is an exact greedy method, too much computation is needed because it requires roughly as many re-routings as the average cardinality times the number of edges in just one iteration. The first to the third options approximate the preference of edges without the eager computation of $\Delta^{(t)}$. Let us consider the case where a

DRM is the initial graph. Intuitively, the length of the edge seems irrelevant to the likelihood that it will remain in the final graph. For this reason, the second one is the option we take because the first and the third employ the length of the edge. Note that, whether an edge is selected with or without replacement is another option. This implementation never replaces edges, but only selects an edge once because the reduced graph is always accepted at line 4.

5.3 Feasible Applications

Supplied the trajectories, the proposed framework generates a route graph that simultaneously minimizes the distance to them and the size of itself. As every trajectory is represented as a sequences of vertices or edges, it is easy to count the traffic of each intersection or segment. The following applications are realized just by combining the existing technologies, instead of developing specific algorithms.

Careful readers might notice that ordinary map-matching can do the same thing and allows us to count the traffic. The road graph and DRM, however, are different in the sense that the former consolidates trajectories and the later does not. Thus, even though the map-matching provides the similar function, their accuracies are different. Section 5.4 gives the empirical proof of this.

5.3.1 Predicting traffic

Estimating probability $p(v)$ for every vertex v and conditional probability $p(f \mid e)$ for every consecutive edges, e and f, is straight forward as we can easily count the traffic for any vertex or segment. Evaluating the conditional probability $p(f \mid e^{(1)}, \cdots, e^{(k)})$ for a directed segment $e^{(1)}, \cdots, e^{(k)}$ and its consecutive edge f realizes the turn prediction at the the terminating end of that segment. These conditional or joint probabilities are evaluated with $p(v)$ or $p(f \mid e)$ and some other useful predictive applications can

be realized similarly. Moreover, recommending routes between origin and destination is realized by using Viterbi algorithm on this Markov model.

5.3.2 Pattern discovery

Once trajectories are transformed to sequences of vertices or edges, the well studied flexible pattern matching [56] and frequent pattern mining technique [36, 35] are applicable to them. The former technique realizes search for similar trajectories to given trajectory, while the later enumerates all popular routes, uncovers the demand for traffic. The route graph can substitute for flock pattern mining [7, 14] because it consolidates trajectories if they flock together.

5.3.3 Compression and anonymization

While traffic analysis contributes to social goods, much care should be paid for privacy concerns because GPS traces record exact locations of citizens. The important thing is to exclude or obfuscate the private information from the discovered results as k-anonymity mining does. Because the route graph aims to reduce its size, traces go along nearby with each other are likely consolidated to common segments. Because of this, the route graph maintains both privacy and utility after removing segments with less than k-populations from it.

5.4 Experiments

In this section, we examine whether our algorithm is able to estimate unobservable paths from sampled trajectories using the benchmark datasets. First, we explain the experimental configurations and then present the results.

We implemented the algorithm with Python and run it on an Ubuntu 16.04 box equipped with Intel Xeon E5-2623 v3 3.00GHz and 256GB memory. The process

	GPS		OSM	
name	#pts	#trace	#nodes	#edges
icdm	2859950	4257	18716	35170
bikely	549920	3150	262699	540017
chicago	118360	889	46533	88942

Table 5.1: Popular GPS trace datasets

employed 16 cores managed by multiprocessing module that comes with Python such that each re-routing runs concurrently for computational efficiency.

5.4.1 Experimental configurations

We use the benchmark datasets of GPS traces, which are collected and shared by various research groups or volunteers. The DRM should be contemporary with the GPS traces, although we utilized the Open Street Map (OSM) of 2017. Table 5.1 shows descriptions of the datasets and their corresponding DRMs. Note that they might differ from those in other reports because those datasets were preprocessed differently.

The goal of this experiments was to examine the accuracy of the algorithms in estimating unobservable paths from sparsified trajectories. These paths, however, are never available because the traces have been originally sampled. As such, we must make some assumptions to evaluate the accuracy even in the following settings:

- an unobservable path is a series of connected links in the DRM, indicating that a car drives on the roads,
- both the observation and sampling noises of the traces are sufficiently small,
- a certain algorithm, such as those using the Fréchet distance [3, 64], can map a trace to its unobservable path if it contains sufficiently little noise.

The first setting is acceptable because the trajectories in the datasets we are using are all from those of cars or bikes. The third setting is also acceptable because map-matching is trivial in that unrealistic case. Even though the second setting depends on datasets, for the sake of experimentation, we decided to accept it.

In the experiments, we sampled observations in a trajectory-wise manner with variable rates, and compared the residuals of the following:

1. unsampled trajectory from DRM (lower bound),
2. unsampled trajectory from route graph (proposed),
3. sampled trajectory from DRM (upper bound).

The first situation gives the lower bounding residual, in that no algorithm can do better than this method, as we had accepted the three above assumptions. The second method is our proposed method. To evaluate how well the route graph represents the major streams in the GPS traces compared to the DRM, we evaluated the residual of the unsampled trajectory from the route graph. The third method evaluates the residual arising from the injection of sampling noises. This gives the upper bounding residual in the sense that no off-the-shelf single-track map-matching does worth than this.

5.4.2 Experimental results

Figure 5.1 shows example results of a single-track and the proposed map-matching algorithms with variable sampling rates. Note that the unobservable path is identical to the result of single-track map-matching in the bottom picture. As the sampling rate increases, the results get similar with each other and they finally become almost identical. The estimated routes by the single-track map-matching cling to the traces because they minimize the Fréchet distances. Especially with the lowest sampling

rate, the route unnaturally comes and goes across the river. On the other hand, the estimated routes by the proposed method are less dependent on the sampling rates than those by the single-track map-matching. Furthermore, with the unsampled trace, the proposed method estimates the even natural route because of the regularization term as we will mention later.

Figure 5.2, 5.3, and 5.4 describe the residuals of the three methods with variable sampling rates. Note that the lower bound is constant because the first method is irrelevant with the sampling rates. We can see that the upper bound curve steeply increases as the sampling rate decreases. This is what we expected, as the lower is the sampling rate, the more the residual experiences injected sampling noises.

In contrast, the curve for the proposed method increases moderately. For instance, in the `icdm` and `chicago` datasets, the proposed method reduced the residual by more than 70% and 40%, respectively, for the upper bound at sampling rate of 40%. The reduction rate tends to increase when the dataset has a larger number of trajectories, which means that the proposed method leverages the residual using the other trajectories. Indeed, we can see that the `icdm` was able to decrease the sampling rate to 40% while the degradation of residual remained nearly constant at 7.0%.

The residual is slightly larger than the upper bound at a sampling rate of 100%, this is because our algorithm accepts a slight increase in the residuals to reduce the route graph. The behavior of the residual as well as the empirical loss is well understood in the regularization technique. The results may accordingly indicate that our algorithm may further exploit the observation and sampling noises.

Figure 5.5, 5.6, and 5.7 describe the description length of graph with variable sampling rates, which corresponds to the regularization term that we introduced to the loss function. In all three datasets, the description length monotonically and asymptotically increases as the sampling rate increases. By contrast, the residual monotonically and asymptotically decreases in Fig.5.2, 5.3, and 5.4, which shows that our algorithm

reasonably favors these contradictory terms depending on the sampling rate.

5.4.3 Discussions

In this section, we discuss the differences between the existing methods and our method. Recall that we have been trying to estimate unobservable paths from traces with sample noise by proposing a new joint map-matching method. Therefore, we would discuss the relation between a joint map-matching [48] and our proposal, as well as the expected advantage of our method to the proceeding stochastic method [58].

Li et al. [48] formalized a joint map-matching as an optimization problem whose objective function contains residual, stitching, and regularization terms. Their method is essentially similar to our method as these two methods share two terms in their objective functions, although the proceeding method requires the stitching term to penalize the fragmentations introduced by its formalization. They differ, however, in terms of algorithms. The proceeding method requires three hyper-parameters while ours requires a single hyper-parameter which can be fixed as $\lambda = 1$ with a good reason. And we remarkably disclosed that the residual and regularization terms are obtained as we formalize the joint map-matching as a generative stochastic process. Concretely, the residual term is caused by the assumption that the distance between trace and unobservable path follows the exponential probability distribution, and that the regularization term is from the other assumption that the paths are equally likely.

Then, we compare our method to another stochastic method that employs HMM [58]. They has two major differences. One is that the latent variables of the HMM-based method correspond to the vertices from which the observations come, while those of our method correspond to the paths from which the traces come. The other is that the HMM-based method maximizes likelihood estimator (MLE), while ours maximizes a posterior (MAP). Thus, we suppose that the HMM-based method likely overfits without restricting the number of hidden states by manually tuning the complexity of

the graph. On the other hand, our method automatically determines the complexity of the graph depending on the number of GPS observations. Because of these differences, we believe that the proposed method has advantages especially with a low sampling rate or less traces.

5.5 Conclusion

In Chapter 5, we proposed a joint map-matching method based on the generative model for estimating unobservable paths by maximizing the posterior probability. Maximization is achieved by the EM algorithm whose object function consists of residual and regularization terms. We presented an iterative algorithm for exploring the route graph, which avoids as many map-matchings as possible by taking advantage of the proven property holding of the residual and the regularization terms. The experimental results showed that the residual degradations from the lower bound were no more than 7.0% when the sampling rate was reduced to 40%. This means that this algorithm reduces the volume of sampling noises and identifies the major streams in the trajectories.

Algorithm 4 Building route graph

Require: trajectories T, minimizing paths B

procedure MAIN($G^{(0)}$)
  for all $t = 1, 2, 3, \cdots$ **do**
    select e from edges in $G^{(t-1)}$
    $G^{(t)} \leftarrow$ APPLY($G(\{e\})$, $G^{(t-1)}$) with some probability
    break by history $\cdots, G^{(t-1)}, G^{(t)}$
  report $G^{(t)}$
function APPLY(g, $G^{(t-1)}$)
  let $\Delta^{(t)}$ be subgraph only $T_{|g}$run
  $\epsilon \leftarrow 0$
  for all $\alpha \in T_{|g}$ **do**
    $\hat{\beta}_\alpha^{(t-1)}, \hat{d}_\alpha^{(t-1)} \leftarrow B[\alpha]$
    $\hat{\beta}_\alpha^{(t)}, \hat{d}_\alpha^{(t)} \leftarrow \mathcal{M}_{G \setminus g}(\alpha)$
    $\Delta^{(t)} \leftarrow \Delta^{(t)} \setminus G(\hat{\beta}_\alpha^{(t)})$
    $\epsilon \leftarrow \epsilon + (\hat{d}_\alpha^{(t)} - \hat{d}_\alpha^{(t-1)})$
    if $\|\Delta^{(t)}\| < \epsilon$ **then**
      return $G^{(t-1)}$
  update $B[\alpha]$ with $\hat{\beta}_\alpha^{(t)}, \hat{d}_\alpha^{(t)}$ for $\alpha \in T_{|g}$
  return $G^{(t-1)} \setminus \Delta^{(t)}$

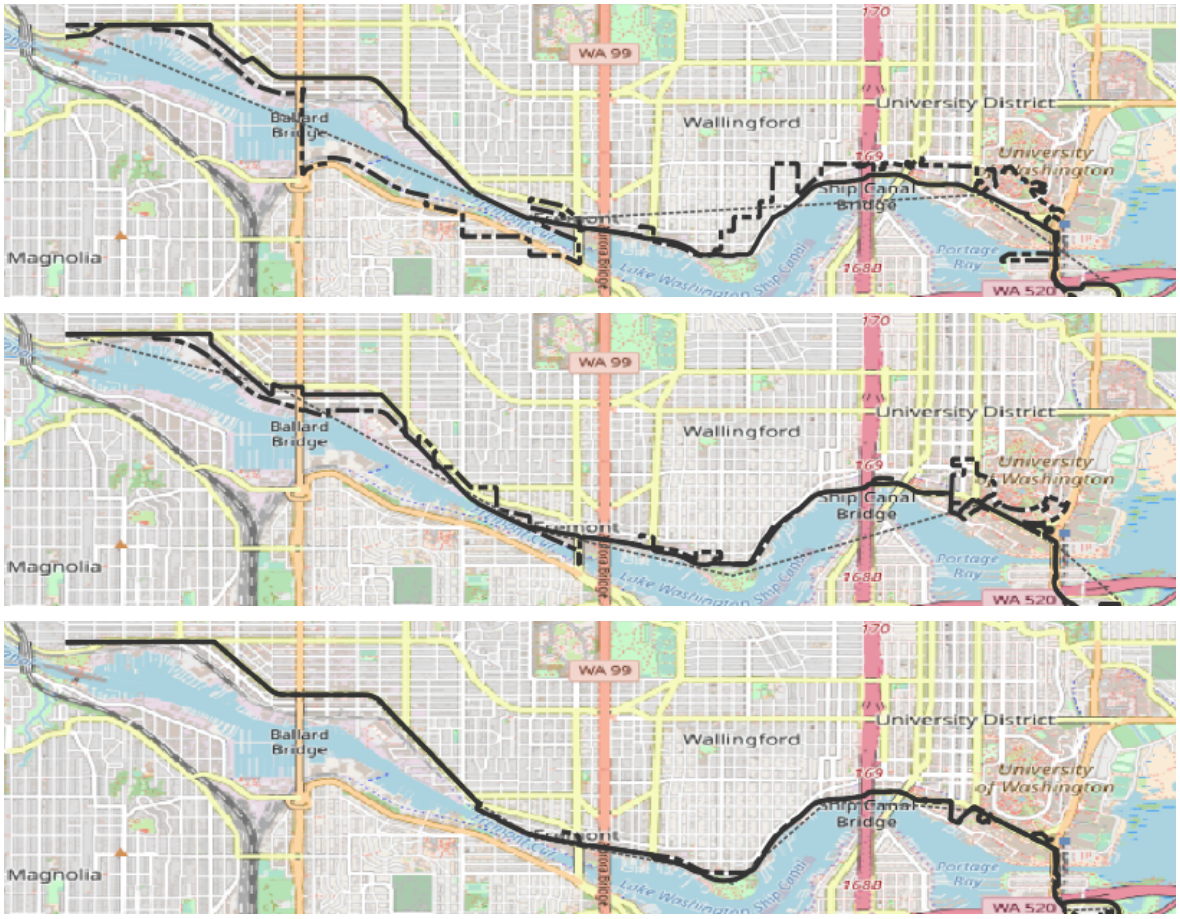


Figure 5.1: Estimated routes by single-track (chain), proposed (solid) map-matching, and sampled trace (dashed) from bikely with rate 20% (top), 40% (middle), and 100% (bottom).

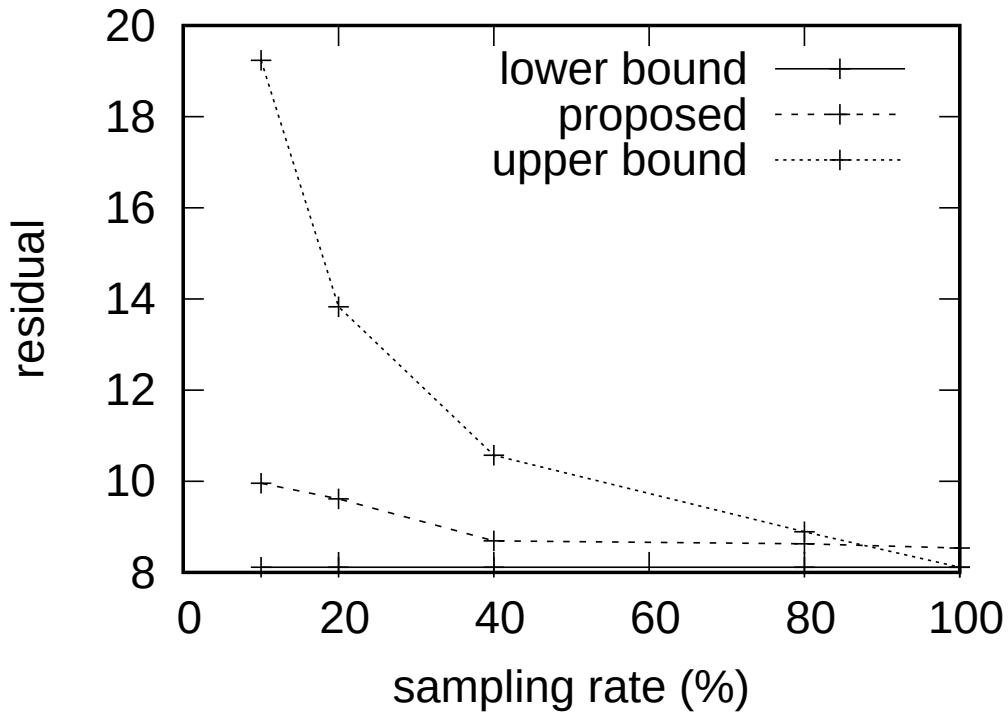


Figure 5.2: Residuals of `icdm` versus sampling rate.

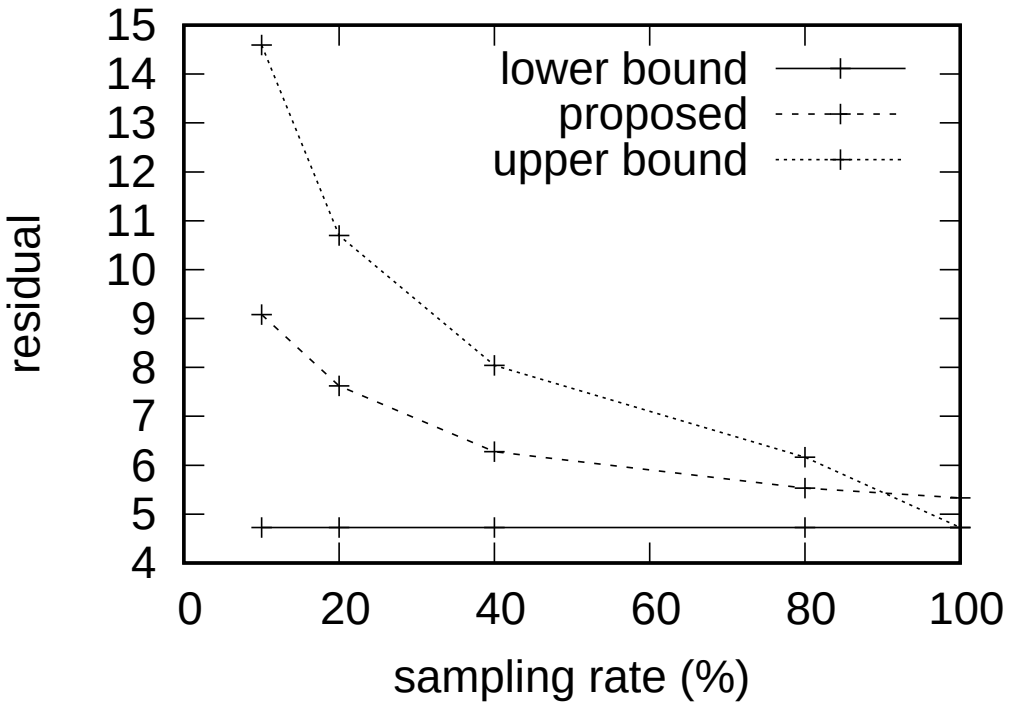


Figure 5.3: Residuals of `bikely` versus sampling rate.

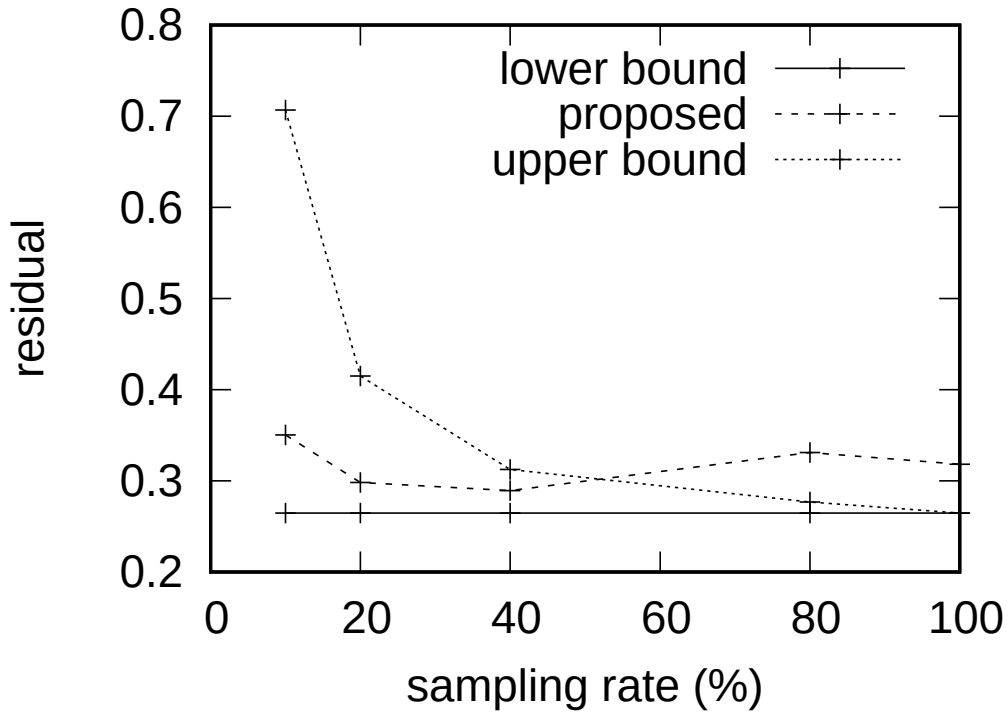


Figure 5.4: Residuals of `chicago` versus sampling rate.

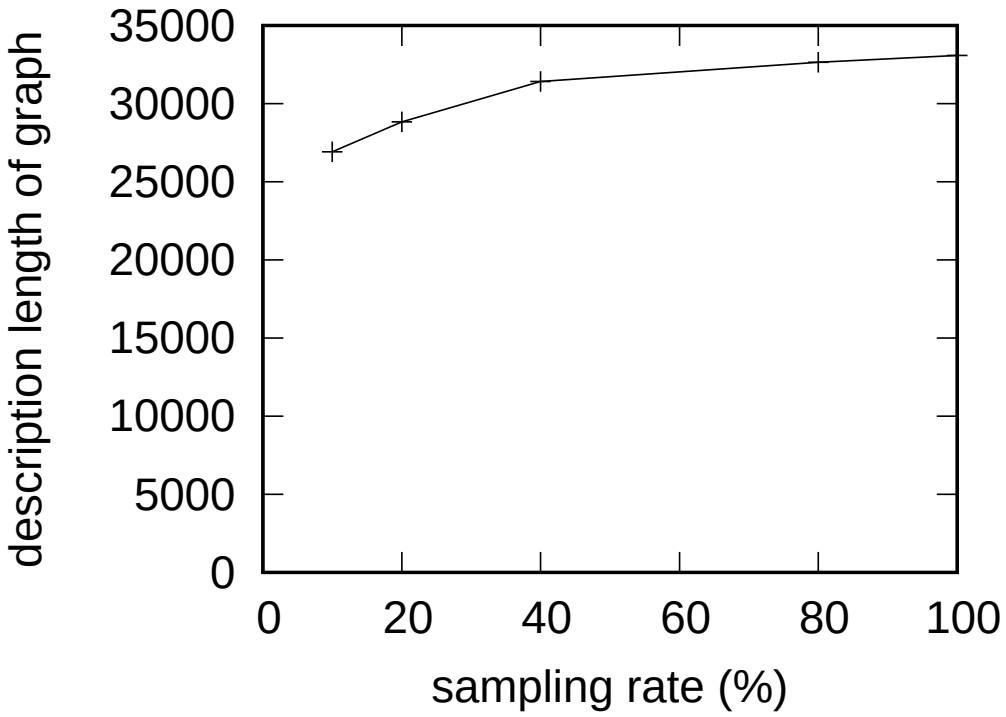


Figure 5.5: Regularization term of `icdm` versus sampling rate.

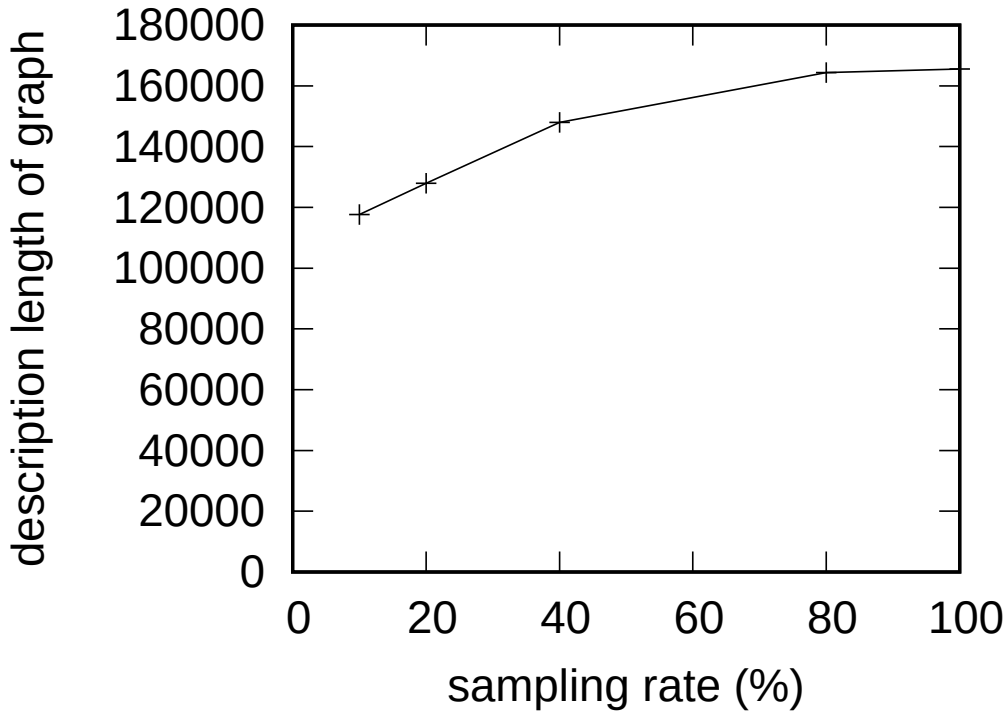


Figure 5.6: Regularization term of `bikely` versus sampling rate.

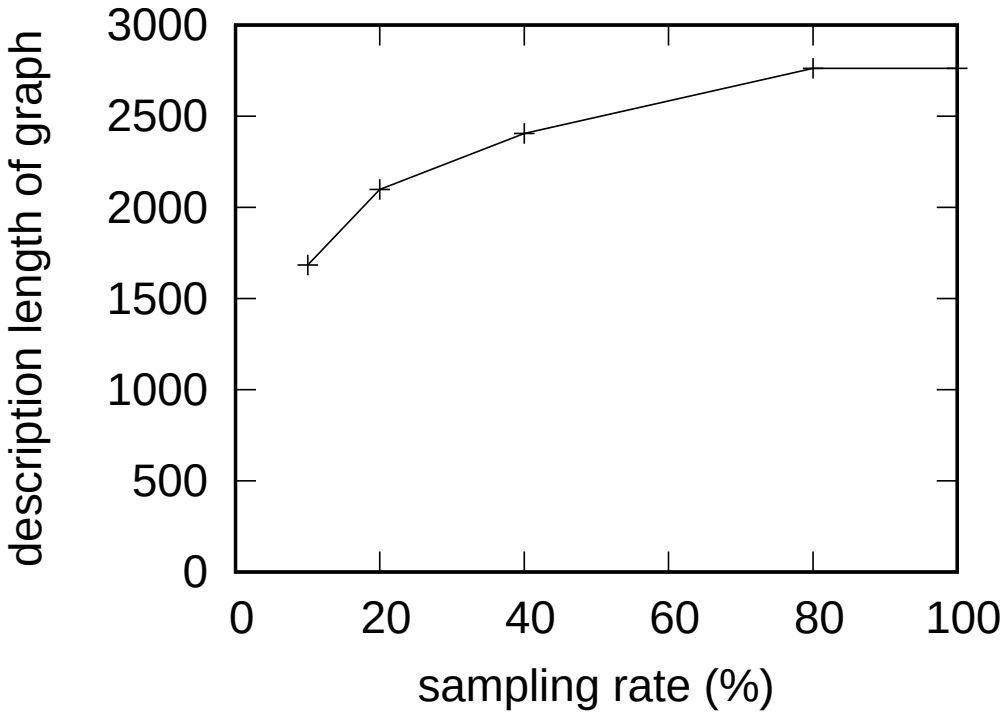


Figure 5.7: Regularization term of of `chicago` versus sampling rate.

Chapter 6

Conclusion

In this thesis, we studied geospatial mobility analysis dealing GPS observations.

Discovery of regions was extended to handle dynamic and probabilistic nature of mobility analysis. We proved the local maximum property to retain relevant cells and proposed the algorithm that compose transitive closures from remaining cells. We also introduced regularization by description length of regions to the object function. The experimental result showed that the algorithm is well applied to detecting hotspots with non-isotropic in ununiformly distributed points, and that the regularized region improves the precision by more than 20% compared to the unregularized one.

Joint map-matching was formalized as maximizing posterior problem. We proposed the algorithm that exploratory optimizes the object function with regularization term representing the number of unique edges of the mapped paths. The experimental result showed the algorithm is well applied to coarsely sampled trajectories without increasing residuals from true paths, and that the residual degradation was within 7.0% even if we map-match trajectories sparsified at a rate of 40% for the practical benchmark datasets.

For the problem of RoI discovery, this works can be continued in two directions: one involves extending it to higher dimensional data. The other is theoretical analysis. Both

the quad-tree and the regularization term are currently apriori. They need justifications based on the theory of statistical machine learning.

For the multi-track map-matching problem, we plan to continue this work in three directions: first, by realizing performance enhancements by further reducing the costly map-matching. One idea is to localize the re-routing to the disabled links without performing map-matching from the origin to destination of the trajectories. The other idea is to extend our algorithm to the incremental one that updates the route graph as trajectories arrive in sequence. The second direction is to develop more sophisticated formulations of the EM algorithm without considering the extreme case, or applying another generative model such as the Hidden Markov Model. The third direction is to apply our method to demand analysis, urban design, and other applications.

Bibliography

[1] Deepak Agarwal, Andrew McGregor, Jeff M. Phillips, Suresh Venkatasubramanian, and Zhengyuan Zhu. Spatial scan statistics: approximations and performance study. In *KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 24–33, New York, New York, USA, 2006. ACM Press.

[2] Deepak Agrawal, Jeff M. Phillips, and Suresh Venkatasubramanian. The hunting of the bump: on maximizing statistical discrepancy. In *SODA '06 Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1137–1146, Miami, Florida, 2006. Association for Computing Machinery.

[3] Helmut Alt, Alon Efrat, Günter Rote, and Carola Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262–283, 2003.

[4] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 05(01n02):75–91, March 1995.

[5] Juan A. Alvarez-Garcia, Juan A. Ortega, Luis Gonzalez-Abril, and Francisco Velasco. Trip destination prediction based on past GPS log using a Hidden Markov Model. *Expert Systems with Applications*, 37(12):8166–8171, December 2010.

[6] Luc Anselin. Thirty years of spatial econometrics. *Papers in Regional Science*, 89(1):3–25, March 2010.

[7] Marc Benkert, Joachim Gudmundsson, Florian Hübner, and Thomas Wolle. Reporting flock patterns. *Computational Geometry*, 41(3):111–125, November 2008.

[8] David Bernstein and Alain Kornhauser. An introduction to map matching for personal navigation assistants. In *The Transportation Research Board 77th Annual Meeting*, Washington, D.C, 1996.

[9] Julian Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

[10] Julian Besag and James Newell. The Detection of Clusters in Rare Diseases. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 154(1):143, 1991.

[11] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[12] Klemens Böhm, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, pages 853–864. ACM, 2005.

[13] Lili Cao and John Krumm. From GPS traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, page 3, New York, New York, USA, 2009. ACM Press.

[14] Yang Cao, Jia Zhu, and Fang Gao. An Algorithm for Mining Moving Flock Patterns from Pedestrian Trajectories. pages 310–321. Springer, Cham, September 2016.

[15] Sudarshan S. Chawathe. Segment-Based Map Matching. In *2007 IEEE Intelligent Vehicles Symposium*, pages 1190–1197. IEEE, June 2007.

[16] Ling Chen, Mingqi Lv, and Gencai Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, December 2010.

[17] Noel A. C. Cressie. *Statistics for Spatial Data.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, September 1993.

[18] Leslie Curry. A Note on Spatial Association. *Professional Geographer*, 18(2):97–99, 1966.

[19] Jonathan J. Davies, Alastair R. Beresford, and Andy Hopper. Scalable, Distributed, Real-Time Map Generation. *IEEE Pervasive Computing*, 5(4):47–54, October 2006.

[20] Mark de Berg, Otfried Chong, Marc van Kreveld, and Mark Overmars. *Computational Geometry - Algorithms and Applications.* Springer, 3rd edition, 2008.

[21] G. Derekenaris, J. Garofalakis, C. Makris, J. Prentzas, S. Sioutas, and A. Tsakalidis. Integrating GIS, GPS and GSM technologies for the effective management of ambulances. *Computers, Environment and Urban Systems*, 25(3):267–278, May 2001.

[22] David P. Dobkin, Dimitrios Gunopulos, and Wolfgang Maass. Computing the Maximum Bichromatic Discrepancy, with Applications to Computer Graphics and Machine Learning. *Journal of Computer and System Sciences*, 52(3):453–470, June 1996.

[23] Luiz Duczmal and Renato Assunção. A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. *Computational Statistics & Data Analysis*, 45(2):269–286, 2004.

[24] Luiz Duczmal, Martin Kulldorff, and Lan Huang. Evaluation of Spatial Scan Statistics for Irregularly Shaped Clusters. *Journal of Computational and Graphical Statistics*, 15(2):428–442, June 2006.

[25] Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. *Notes*, 94:64, 1994.

[26] Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

[27] Jon Froehlich and John Krumm. Route prediction from trip observations. In *SAE Technical Papers*. SAE International, 2008.

[28] Takeshi Fukuda, Yasuhiko Morimoto, Shimichi Morishita, and Takeshi Tokuyama. Data Mining with optimized two-dimensional association rules. *ACM Transactions on Database Systems*, 26(2):179–213, June 2001.

[29] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. *ACM SIGMOD Record*, 25(2):13–23, 1996.

[30] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. *Mining optimized association rules for numeric attributes*. ACM Press, New York, New York, USA, June 1996.

[31] Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining Optimized Association Rules for Numeric Attributes. *Journal of Computer and System Sciences*, 58(1):1–12, 1999.

[32] Arthur Getis. Cliff, A.D. and Ord, J.K. 1973: Spatial autocorrelation. London: Pion. *Progress in Human Geography*, 19(2):245–249, June 1995.

[33] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 330, New York, New York, USA, August 2007. ACM Press.

[34] Joshua Greenfeld. Matching GPS observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, Washington D.C., 2002.

[35] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier Inc., 2012.

[36] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining Cambridge*. Undergraduate Topics in Computer Science. Springer London, London, 2001.

[37] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2 edition, 2009.

[38] Yu-Ling Hsueh and Ho-Chian Chen. Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions. *Information Sciences*, 433-434:55–69, April 2018.

[39] Hiroya Inakoshi, Hiroaki Morikawa, Tatsuya Asai, Nobuhiro Yugami, and Seishi Okamoto. *Discovery of areas with locally maximal confidence from location data*, volume 8421 LNCS. 2014.

[40] Fumio Ishioka and Koji Kurihara. Evaluation of Hotspot Detection Method based on Echelon Structure. In *59th ISI World Statistics Congres*, page 5366, Hong Kong, 2013.

[41] Adel Javanmard, Maya Haridasan, and Li Zhang. Multi-track Map Matching. September 2012.

[42] Sophia Karagiorgou and Dieter Pfoser. On vehicle tracking data-based road network generation. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, page 89, New York, New York, USA, 2012. ACM Press.

[43] Sophia Karagiorgou, Dieter Pfoser, and Dimitrios Skoutas. Segmentation-based road network construction. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*, pages 460–463, New York, New York, USA, 2013. ACM Press.

[44] John Krumm. Where will they turn: predicting turn proportions at intersections. *Personal and Ubiquitous Computing*, 14(7):591–599, October 2010.

[45] John Krumm, Robert Gruen, and Daniel Delling. From destination prediction to route prediction. *Journal of Location Based Services*, 7(2):98–120, June 2013.

[46] Martin Kulldorff. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496, January 1997.

[47] Martin Kulldorff and Neville Nagarwalla. Spatial disease clusters: Detection and inference. *Statistics in Medicine*, 14(8):799–810, April 1995.

[48] Yang Li, Qixing Huang, Michael Kerber, Lin Zhang, and Leonidas Guibas. Large-scale joint map matching of GPS traces. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*, pages 214–223, New York, New York, USA, November 2013. ACM Press.

[49] Xuemei Liu, James Biagioni, Jakob Eriksson, Yin Wang, George Forman, and Yanmin Zhu. Mining large-scale, sparse GPS traces for map inference. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, page 669, New York, New York, USA, 2012. ACM Press.

[50] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, page 352, New York, New York, USA, November 2009. ACM Press.

[51] Marco Mamei, Alberto Rosi, and Franco Zambonelli. Automatic Analysis of Geotagged Photos for Intelligent Tourist Services. In *2010 Sixth International Conference on Intelligent Environments*, pages 146–151. IEEE, July 2010.

[52] Georges Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, December 1963.

[53] Tomáš Mikluščák, Michal Gregor, and Aleš Janota. Using Neural Networks for Route and Destination Prediction in Intelligent Transport Systems. pages 380–387. Springer, Berlin, Heidelberg, October 2012.

[54] Mehryar. Mohri, Afshin. Rostamizadeh, and Ameet. Talwalkar. *Foundations of machine learning*. MIT Press, 2012.

[55] M. Nanni, B. Kuijpers, C. Körner, M. May, and D. Pedreschi. *Spatiotemporal Data Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[56] Gonzalo Navarro and Mathieu Raffinot. *Flexible pattern matching in strings: Practical on-line search algorithms fortexts and biological sequences*. Cambridge University Press, January 2014.

[57] Daniel B. Neill and Andrew W. Moore. Rapid detection of significant spatial clusters. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 256, New York, New York, USA, 2004. ACM Press.

[58] Paul Newson and John Krumm. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, page 336, New York, New York, USA, November 2009. ACM Press.

[59] S. Panahi and Delavar. Dynamic Shortest Path in Ambulance Routing Based on GIS. *International Journal of Geoinformatics*, 5(1), March 2009.

[60] Ganapati P. Patil and Charles Taillie. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environmental and Ecological Statistics*, 11(2):183–197, June 2004.

[61] Oliver Pink and Britta Hummel. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 862–867. IEEE, October 2008.

[62] Mohammed A. Quddus, Washington Yotto Ochieng, Lin Zhao, and Robert B. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7(3):157–167, December 2003.

[63] Rajeev Rastogi and Kyuseok Shim. Mining optimized association rules with categorical and numeric attributes. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):29–50, 2002.

[64] Junichi Shigezumi, Tatsuya Asai, Hiroaki Morikawa, and Hiroya Inakoshi. A Fast Algorithm for Matching Planar Maps with Minimum Fréchet Distances. In

Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data - BigSpatial'15, pages 25–34, New York, New York, USA, 2015. ACM Press.

[65] Toshiro Tango. A spatial scan statistic scanning only the regions with elevated risk. *Advances in Disease Surveillance*, 2007.

[66] Toshiro Tango. A Spatial Scan Statistic with a Restricted Likelihood Ratio. *Japanese Journal of Biometrics*, 29(2):75–95, 2008.

[67] Toshiro Tango and Kunihiko Takahashi. A flexible spatial scan statistic with a restricted likelihood ratio for detecting disease clusters. *Statistics in Medicine*, 31(30):4207–4218, December 2012.

[68] Toshiro Tango, Kunihiko Takahashi, RJ Marshall, A Lawson, D Denison, LA Waller, CA Gotway, J Cuzick, R Edwards, J Besag, J Newell, M Kulldorff, N Nagarwalla, M Kulldorff, T Tango, T Tango, JF Viel, P Arveux, J Baverel, JY Cahn, OA Sankoh, Y Ye, R Sauerborn, O Muller, H Becher, AM Perez, MP Ward, P Torres, V Ritacco, M Kulldorff, T Tango, PJ Park, C Song, M Kulldorff, M Kulldorff, GP Patil, C Taillie, L Duczmal, R Assunção, M Dwass, K Takahashi, T Yokoyama, and T Tango. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 4(1):11, 2005.

[69] Horanont Teerayut, Apichon Witayangkurn, and Ryosuke Shibasaki. The Challenge of Geospatial Big Data Analysis. In *Open Source Geospatial Research & Education Symposium*, 2012.

[70] Vishnu S. Tiwari, Sudha Chaturvedi, and Arti Arya. Route prediction using trip observations and map matching. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 583–587. IEEE, February 2013.

[71] Roberto Trasarti, Fabio Pinelli, Mirco Nanni, and Fosca Giannotti. Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, page 1190, New York, New York, USA, August 2011. ACM Press.

[72] Suyi Wang, Yusu Wang, and Yanjie Li. Efficient map reconstruction and augmentation via topological methods. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, pages 1–10, New York, New York, USA, 2015. ACM Press.

[73] Carola Wenk, Randall Salas, and Dieter Pfoser. Addressing the Need for Map-Matching Speed: Localizing Globalb Curve-Matching Algorithms. In *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, pages 379–388, Vienna, Austria, 2006. IEEE.

[74] Zhijun Yao, Junmei Tang, and F Benjamin Zhan. Detection of arbitrarily-shaped clusters using a neighbor-expanding approach: A case study on murine typhus in South Texas. *International Journal of Health Geographics*, 10(1):23, March 2011.

[75] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, page 325, New York, New York, USA, July 2011. ACM Press.

[76] Daisuke Yoshida, Xianfeng Song, and Venkatesh Raghavan. Development of track log and point of interest management system using Free and Open Source Software. *Applied Geomatics*, 2(3):123–135, July 2010.

[77] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th*

international conference on World wide web - WWW '10, page 1029, New York, New York, USA, April 2010. ACM Press.

[78] Yu Zheng, Xing Xie, Chengyang Zhang, Yin Lou, Wei Wang, and Yan Huang. Map-Matching for Low-Sampling-Rate GPS Trajectories, November 2009.

[79] Yu Zheng and Xiaofang Zhou, editors. *Computing with Spatial Trajectories*. Springer, 2011.