Geospatial Data Science to Identify Patterns of Evasion

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Emre Eftelioglu

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advisor: Professor Shashi Shekhar

January, 2018

Acknowledgements

I thank my advisor, Professor Shashi Shekhar, for his guidance, support and patience throughout my Ph.D. He was always supportive and he always encouraged me to investigate interesting and challenging problems. I also would like to thank Professor George Karypis for having confidence in me and encouraging me with his support. I really enjoyed our personal conversations. In addition, I am grateful for the help of all the professors who helped me over the years in classes and those who served on my committee: Professor Jaideep Srivastava, Professor Vipin Kumar, Professor Christopher Uggen, and Professor John Hughes. Thank you for showing me the correct direction to shape my research.

I thank my collaborators from the National Geospatial-Intelligence Agency for giving me invaluable suggestions and helping me make my research better. Especially, I would like to thank Christopher Farah for providing me feedbacks on every step of my research. I really enjoyed our phone meetings, which improved my communication and presentation skills. I will miss them.

I thank my friends in Professor Shekhar's Spatial Computing research group for all the meetings, brainstorms on whiteboard and the great insights they provided. I will definitely miss you and will always remember you as good friends. I enjoyed our conversations on not only research but everything in life. Particularly, I thank Dev and Mike for mentoring me in my first years in the Ph.D. Both of you were great role models and mentors. I thank Xun for his great friendship. Together, Ph.D. was less stressful. I thank Kim Koffolt for reading drafts of my research papers and providing me feedback. She taught me how to write good and grammatically sound papers in English.

Lastly, I would like to express my deepest gratitude to my family. I thank my beautiful wife Tuba for all of her support. In my hardest times, she was always with me

and always encouraged me that I can do anything as long as I want. I thank my little princess, my daughter Selin for her love and our play times together. Playing with her was the best therapy I would ever get. I thank the new member of our family, Efe, for giving us hope and excitement in the most stressful times of our lives. Finally, I would like to thank my parents Mehmet and Nursel for their never-ending support. I am glad of having such supportive parents and I wish I will be a parent just like them. I hope they are proud of me.

Abstract

Over the last decade, there has been a significant growth in the availability of cheap raw spatial data in the form of GPS trajectories, activity/event locations, temporally detailed road networks, satellite imagery, etc. These data are being collected, often around the clock, from location-aware applications, sensor technologies, etc. and represent an unprecedented opportunity to study our economic, social, and natural systems and their interactions. For example, detecting hotspots (areas with unusually high concentration of activities/events) from activity/event locations plays a crucial role in epidemiology since it may help public health officials prevent further spread of an infectious disease. In order to extract useful information from these datasets, many geospatial data tools have been proposed in recent years. However, these tools are often used as a "black box", where a trial-error strategy is used with multiple approaches from different scientific disciplines (e.g. statistics, mathematics and computer science) to find the best solution with little or no consideration of the actual phenomena being investigated. Hence, the results may be biased or some important information may be missed. To address this problem, we need geospatial data science with a stronger scientific foundation to understand the actual phenomena, develop reliable and trustworthy models and extract information through a scientific process.

Thus, my thesis investigates a wide-lens perspective on geospatial data science, considering it as a transdisciplinary field comprising statistics, mathematics, and computer science. This approach aims to reduce the redundant work across disciplines as well as define scientific boundaries of geospatial data science to distinguish it from being a black box that claims to solve every possible geospatial problem. In my proposed approaches, I used ideas from those three disciplines, e.g. spatial scan statistics from statistical science to reduce chance patterns in the output and provide statistical robustness; mathematical definitions of geometric shapes of the patterns, which maintain correctness and completeness; and computational approaches (along with prune and refine framework and dynamic programming ideas) to scale up to large spatial datasets. In addition, the proposed approaches incorporate domain-specific geographic theories (e.g., routine activity theory in criminology) for applicability in those domains that are

interested in specific patterns, which occur due to the actual phenomena, from geospatial datasets. The proposed techniques have been applied to real world disease and crime datasets and the evaluations confirmed that our techniques outperform current state-of-the-art such as density based clustering approaches as well as circular hotspot detection methods.

Contents

Acknowledgements i

Abstract iii

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Geospatial Data Science 1

1.2 Geospatial Data-Driven Science 3

1.3 Illustrative Application Domain: Environmental Criminology 5

1.4 Challenges 6

1.5 Thesis Contributions 8

2 Geospatial Data Science: A Transdisciplinary Perspective 11

2.1 Introduction 11

2.1.1 Motivation 12

2.1.2 Problem Definition 13

2.1.3 Challenges 14

2.1.4 Trade-Offs 15

2.2 Background 19

2.3 Contributions, Scope, and Outline of the Chapter 20

2.4 Statistics 22

2.4.1 Traditional Statistics 22

2.4.2 Traditional Statistics versus Spatial Statistics 23
2.4.3 Spatial Statistics 25
2.5 Mathematics 28
2.5.1 Mathematics in Traditional Data Science 28
2.5.2 Limitations of Applying Traditional Mathematical Models to Spatial Data and Novel Spatial Models 30
2.6 Computer Science 34
2.6.1 Core Questions and Goals 34
2.6.2 Concepts, Theories, Models and Technologies 35
2.6.3 Limitations of Traditional Data Science for Spatial Data and Related Computer Science Accomplishments 37
2.7 Conclusion 40

3 Ring-Shaped Hotspot Detection 41
3.1 Introduction 41
3.1.1 Application Domains 42
3.1.2 Challenges 44
3.1.3 Related Work and Their Limitations 45
3.1.4 Contributions 46
3.1.5 Scope and Outline 47
3.2 Basic Concepts and Problem Statement 48
3.2.1 Basic Concepts 48
3.2.2 Problem Statement 51
3.3 Preliminary Results - Dual Grid Based Pruning Algorithm (DGP) . . . 52
3.3.1 Basic Concepts of DGP Algorithm 52
3.3.2 Dual Grid based Pruning (DGP) Algorithm 57
3.4 Proposed Approach 60
3.4.1 Algorithmic Refinements for the Prune Phase 61
3.4.2 Refine Phase: A Best Enclosing Ring Heuristic 66
3.5 Theoretical Evaluation 69
3.5.1 Correctness 70
3.5.2 Computational Analysis 70

3.6 Case Study . . . 71
3.7 Experimental Evaluation . . . 72
3.7.1 Experimental Design & Data Sets . . . 73
3.7.2 Comparative Analysis . . . 74
3.7.3 Sensitivity Analysis . . . 76
3.8 Discussion . . . 79
3.9 Conclusions and Future Directions . . . 80

4 Geographically Robust Hotspot Detection 81
4.1 Introduction . . . 81
4.1.1 Application Domain: . . . 81
4.1.2 Challenges: . . . 84
4.1.3 Related Work: . . . 84
4.1.4 Contributions: . . . 86
4.1.5 Scope and Outline: . . . 86
4.2 Basic Concepts and Problem Formulation . . . 87
4.2.1 Basic Concepts . . . 87
4.2.2 Problem Formulation . . . 88
4.3 CGC Algorithm . . . 90
4.3.1 Baseline Approach: SaTScan Algorithm . . . 90
4.3.2 Proposed Approach . . . 92
4.4 Theoretical Evaluation . . . 102
4.4.1 Correctness of the CGC Algorithm . . . 102
4.4.2 Computational Analysis of the CGC Algorithm . . . 102
4.5 Case Study . . . 103
4.6 Experimental Evaluation . . . 104
4.6.1 Experimental Design . . . 104
4.6.2 Experimental Results . . . 105
4.7 Discussion . . . 107
4.8 Conclusions and Future Work . . . 108

5 Mining Network Hotspots with Holes 109
5.1 Introduction . . . 109

5.2 Basic Concepts and Problem Statement 113
5.2.1 Basic Concepts 113
5.2.2 Problem Statement 115
5.3 Proposed Approach 116
5.3.1 Naïve NHH Generator Algorithm 116
5.3.2 Smart NHH Generator Algorithm 119
5.4 Case study 122
5.5 Experimental Evaluation 125
5.6 Discussion 126
5.7 Conclusion 127

6 Conclusions and Future Work **128**
6.1 Key Results 128
6.2 Short Term Future Directions 129
6.3 Long Term Future Directions 132

References **134**

List of Tables

2.1 Overview of the related work. 19
3.1 Geo-grid cells for $lg = 1$ generated for activity set in Figure 3.5(a). . . 55
3.2 Sample param-grid cells for $lg = 1$ generated for activity set in Figure 3.5(a). 55
3.3 Sample Output of the DGP for the activity set in Figure 3.5(a). 60
3.4 Computational complexity analysis 71
3.5 Experiment datasets generated for the Experimental Evaluation ($S = 900 \times 900$, $m = 0$). 74
5.1 An example execution trace of NaïveNHHG. 118
6.1 Thesis contributions and their related geospatial data science focus. . . 129
6.2 Patterns of evasion that could be studied in different domains. 132

List of Figures

1.1 Examples of approaches by their scientific foundations from mathematics, statistics and computer science. . . . 2
1.2 A Geospatial data driven science with the addition of domain specific theories. . . . 4
1.3 Criminals typically do not commit crime close to home (inside the dashed circle) and do not travel too far (beyond the ring) [1]. . . . 6
2.1 Steps of geospatial data science. . . . 14
2.2 The statistical bias and variance. . . . 16
2.3 The trade-off between computational scalability and statistical rigor. . . . 18
2.4 Comparison of disciplines via examples. . . . 21
2.5 Distribution of disease. . . . 24
2.6 Forms of Spatial Point Processes. . . . 26
2.7 An illustrative example showing the difference between distances in Euclidean and network spaces. . . . 31
2.8 An example of modeling the turns of a transportation network. . . . 32
2.9 An example of Time-Expanded-Graph (TEG). . . . 33
2.10 An illustrative example of dynamic segmentation. . . . 34
2.11 Intel Distribution for Apache Hadoop software components compared with the high-performance computing ecosystem. Figure adapted from [2] and [3]. . . . 36
2.12 Traditional Decision Tree versus Spatial Decision Tree Output for Classifying data from satellite imagery [4]. . . . 39
3.1 Simplified example of Ring-Shaped Hotspot Detection. . . . 42

3.2 Criminals typically do not commit crime close to home (inside the dashed circle) and do not travel too far (beyond the ring) [1]. 43
3.3 Gravity anomaly map of Chicxulub Crater shows ring-shaped anomalous regions of gravity [5]. 44
3.4 Related work classification. 45
3.5 Example Input, Output of SaTScan, and Output of Ring-Shaped Hotspot Detection (best in color). 46
3.6 An illustrative example of a 4P-CRing. Dots represent points, dashed circle represent inner circle $circle_i$ and circle with a solid line represents outer circle $circle_o$. 49
3.7 A sample geo-grid for the input activity set with 100 activity points. In this example, cell length is set to $lg = 9$. 53
3.8 Shaded cells show a sample minimal bounding geo-sub-grid(R) and a sample maximal bounded geo-sub-grid(R) used for the computation of the upper bound log likelihood ratio ($Log\ \widehat{LR_p}$). 53
3.9 Illustration of the output of DGP. 61
3.10 Illustration of missing rings due to coarse cell length (3.10(a)) and using multi cell lengths for the center (in the middle of shaded cells - shown in yellow) and width (shaded cells - shown in green/blue) of the rings (3.10(b)) to detect those. 62
3.11 Illustration of local maxima elimination approach. Param-cell pc with $tag = 5$ will be returned after the local maxima elimination. 63
3.12 Illustration of multi cell length approach. Initially cell lengths are selected as $lg_w = 9$ and $lg_c = 9$. Second row for the top right ring is shown for illustration purposes. This ring will be missed since the first iteration will prune it out. 64
3.13 Inner circles created in the first iteration of the refine phase. Blue dots represent the points in $prunedSet$, red squares represent the Voronoi vertices. Inner circles are created with the vertices inside the black square representing the center interval returned by prune phase (best in color). 67

3.14 New York Bronx Legionnaire's disease cases (collected from the map in [6]) in 2015 and the output of SaTScan and RHD. RHD output validates the actual location of the outbreak as Opera House Hotel [7]. . . 72
3.15 Experimental Design 73
3.16 Comparative Analysis. Execution time for DGP (A), DGP with Local Maxima Elimination (B), DGP with Multi Cell Size Approach (C), DGP with both Local Maxima Elimination and Multi Cell Size Approach (D), DGP with Best Enclosing Ring Heuristic (E) and DGPLMR with all algorithmic refinements (F) (best in color). 75
3.17 Sensitivity analysis of DGPLMR algorithm (best in color). 77
3.18 Two of the activity sets that were used for the experiment on the variance of activity distribution of rings (best in color). 78
3.19 DBSCAN output for a random activity set and an activity set with rings (best in color). 80
4.1 1854 London Cholera Outbreak [8]. The blue hotspot indicates the location of the infected water pump.(best in color). 82
4.2 Related work of hotspot detection. 84
4.3 Example output of Geographically Robust Hotspot Detection (GRHD) compared to DBSCAN [9] and SaTScan [10] (best in color). 85
4.4 Execution trace of SaTScan algorithm with points as centers (in color). Red rows correspond the red circles in Figure 4.4(c) (best in color). . . . 89
4.5 Count and Circle Grid cells for $l_{cell} = 5$ (best in color). 93
4.6 Illustration of minimum enclosing (MECC) and maximum fit cell collections (MFCC) shown in green which represent a circle (in red) 94
4.7 Illustration of the refine phase in CGC algorithm 98
4.8 Execution trace of the CGC algorithm (best in color). 101
4.9 Figure 4.9(a) shows 64 unarmed robbery crimes committed in San Diego in 2013 (blue dots) [11]. Figure 4.9(b) and Figure 4.9(c) compare the output of SaTScan and GRHD with the CGC algorithm(red/green circles - best in color). 104

4.10 Scalability of CGC with increasing (a), (b) number of points, (c) log likelihood ratio threshold, (d) cell length and (e) number of Monte Carlo simulation trials. 106
5.1 A path between home and work, an inner buffer zone where the activities are sparse and an activity zone (i.e. outer buffer) that we are interested in. 110
5.2 An example input and output of our proposed approach for Mining Network Hotspots with Holes. Edges represent streets and Nodes represent road intersections. 111
5.3 Output of the related work for the input in Fig.5.2(a). SaTScan (Fig.5.3(a)), Ring-Shaped Hotspot Detection (Fig.5.3(b)) and Significant Route Discovery (Fig.5.3(c)) 112
5.4 Distance based dynamic programming approach. To determine $NHH_{F4,F5,1,3}$, the set union of $NHH_{F4,F5,1,2}$ and $NHH_{F4,F5,2,3}$ is used (best in color). . 120
5.5 Edge stitching approach. To determine $NHH_{G4,G6,1,2}$ (Fig.5.2(b)); $NHH_{G4,G5,1,2}$ and $NHH_{G5,G6,1,2}$ are stitched together and then the nodes and edges of $NB_{G4,G6,1}$ are removed (best in color). 121
5.6 Case Study 1: Theft crimes in Chicago, Illinois. Fig.5.6(a) shows the input, Fig.5.6(b) shows the output of SaTScan, Fig.5.6(c) shows the output of Ring-Shaped Hotspot Detection and Fig.5.6(d) shows the output of SmartNHHG. *Log LR* values are not comparable due to the Euclidean and Network spaces. (best in color). 123
5.7 Case Study 2: Burglary crimes in Oakland, California. Fig.5.7(a) shows the input, Fig.5.7(b) shows the output of SaTScan, Fig.5.7(c) shows the output of Ring-Shaped Hotspot Detection and Fig.5.7(d) and 5.7(e) shows the output of SmartNHHG. *Log LR* values are not comparable due to the Euclidean and Network spaces. 124
5.8 Scalability of SmartNHHG with increasing 5.8(a) number of nodes, 5.8(b) number of activities 5.8(c), and unit distance (ω). 125

Chapter 1

Introduction

Over the last decade, there has been a significant growth in the availability of cheap raw spatial data in the form of GPS trajectories, activity/event locations, temporally detailed road networks, satellite imagery, etc. Geospatial datasets, already being used by over a billion people (e.g. Google Maps, Uber, etc.), are the basis of an ever growing number of more technologies and applications. For example, vehicle GPS trajectories together with engine measurement data are providing a new way to recommend environmentally friendly routes. Similarly, self-driving cars, robots, drones, and many newer mobile applications (e.g. virtual reality glasses, games, etc.) collect geospatial datasets to accomplish their tasks. Increasingly, geospatial datasets are proving their potential to revolutionize how we understand the physical world, know and communicate our relation to places in that world, and navigate through those places.

1.1 Geospatial Data Science

In order to extract useful information from geospatial datasets, many geospatial computing tools were proposed in the recent years. However, these tools are often used as a "black box" using a trial-error strategy in which multiple approaches from different scientific disciplines (e.g. statistics, mathematics and computer science) are used to find the best solution with little or no consideration of the actual phenomena being investigated. Hence, the results may be biased or some important information may be missed. There is a need for an approach with a stronger scientific foundation, namely geospatial

data science, to understand the actual phenomena, develop reliable and trustworthy models and extract information through a scientific process. Geospatial data science can be considered as a transdisciplinary field comprising statistics, mathematics, and computer science. Scientific methods from these three disciplines can be used together to leverage similar concepts to increase robustness of solutions while also reducing the redundant work across disciplines.

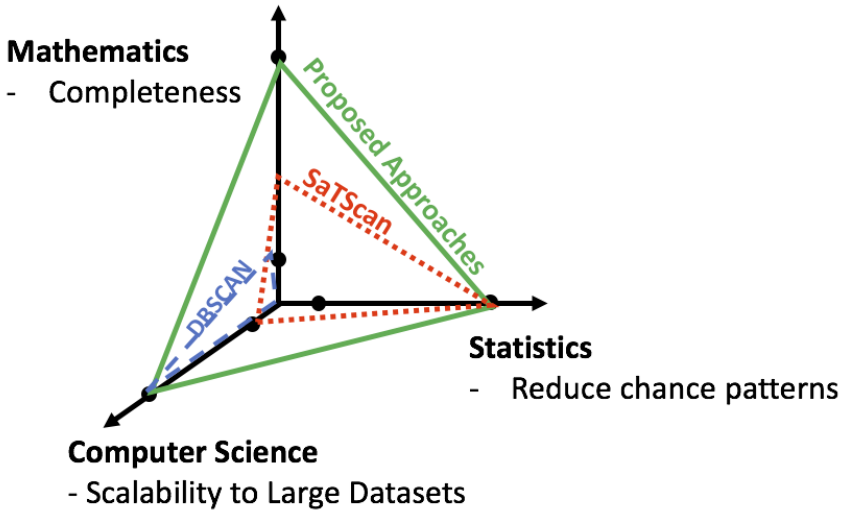


Figure 1.1: Examples of approaches by their scientific foundations from mathematics, statistics and computer science.

In Figure 1.1, representative approaches (e.g. SaTScan, DBSCAN), that are often used for hotspot detection are compared relative to their emphasis in different scientific domains. The figure depicts the three disciplines, namely mathematics, statistics and computer science as three axes and summarizes their differing objectives (from a hotspot detection perspective):

- Mathematics is focused on the correctness and completeness of approaches,
- Statistics is focused on eliminating chance patterns,
- Computer science is focused on scalability to large datasets.

The proposed approaches in this thesis aim to achieve the objectives of these scientific disciplines by ensuring solution correctness and completeness, providing statistical significance testing to reduce chance patterns, and making algorithms scalable to large geospatial datasets. As can be seen from the figure, some existing approaches may not be able to achieve the goals of all three disciplines. For example, SaTScan, a state-of-the-art software package that uses spatial scan statistics to detect circular hotspots, gives importance to statistics for reducing chance patterns but its computational scalability to large datasets is limited to a couple of thousand points and its completeness (i.e. the robustness of its definitions) is limited due to its circle enumeration strategy (a point at center and a point on the circumference of a circle). Similarly, DBSCAN, a representative density based clustering method, uses an efficient and computationally scalable algorithm to detect dense clusters in a point dataset. However, its mathematical completeness depends on user input variables (e.g. "Eps" and "k" values) and it doesn't use robust statistical approaches to eliminate chance patterns.

1.2 Geospatial Data-Driven Science

Geospatial problems are often the focus of research in fields such as epidemiology, criminology or precision agriculture. We may call these geospatial data-driven sciences. Geospatial data driven sciences can be seen as another way to approach geospatial problems. Often geospatial datasets are collected from a variety of sources by domain scientists trying to solve a specific problem in a specific application domain. Thus, in addition to the considerations of geospatial data science, domain-specific theories and insights may help shape the geospatial tools that domain scientists require in data driven sciences. For example, in hotspot detection, there are many tools to detect the locations with significant concentrations of activity.

- In criminology, domain specific theories, e.g. routine activity theory and crime pattern theory, provide insights for the shape of an interesting hotspot pattern, which leads to the definition of ring-shaped hotspots in Chapter 3.
- Diffusion theory in epidemiology may be said to emphasize hotspots of diseases around the source of the disease, making circular hotspots more interesting than

arbitrary shapes.

- In the agriculture domain, one may be interested in the locations in a field where crop health is significantly lower than other locations. For such cases, rectangular hotspot detection, where each rectangle corresponds to a segment of the field, may be interesting.

- Another example from transportation planning domain is that since pedestrian fatalities occur on a road segment, using tools for circular or ring-shaped hotspot in Euclidean space may not make sense for transportation planners. Thus, a linear/path hotspot detection method on a road network may be preferable to identify those road segments with significantly high pedestrian fatalities as described in [12, 13].

Therefore, domain specific theories may help domain scientists understand the actual phenomena under study and often such tools may not make sense for another application domain. In Figure 1.2, we update Figure 1.1 by adding domain specific theories as another axis to emphasize the importance of applying domain specific theories and constraints.

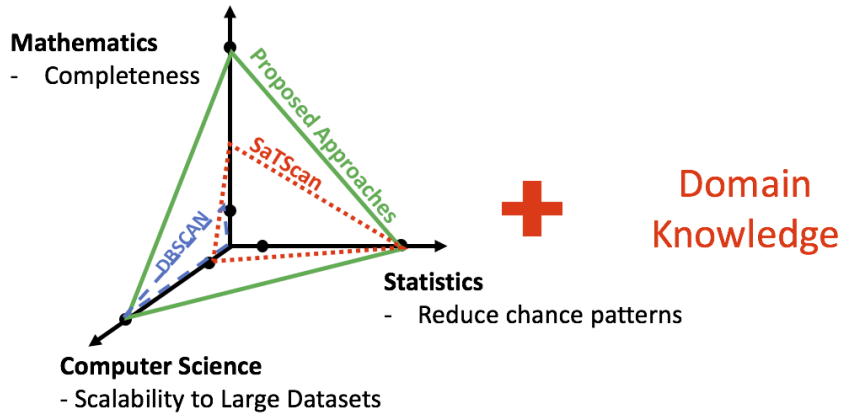


Figure 1.2: A Geospatial data driven science with the addition of domain specific theories.

1.3 Illustrative Application Domain: Environmental Criminology

Environmental criminology is the study of crime, criminality, and victimization as they relate to particular places and to the way that individuals and organizations shape their activities spatially [14]. Two important spatial theories in environmental criminology are routine activity theory, which suggests that the location of a crime is related to a serial criminal's frequently visited areas [15], and crime pattern theory, which extends routine activity theory to a spatial model [16]. These theories are the basis for geographic profiling by crime analysts, who use the locations of connected crime sites and then try to analyze this spatial information to identify the underlying factors of a crime [17].

For this purpose, several analytical tools have been widely adopted by crime analysts [18, 19]. For example, crime hotspot analysis helps detect locations where crime incidents tend to be concentrated [18]. These tools are useful for the prevention of crime by deploying new police forces near hotspots, but they are not designed to predict the potential residence of a serial criminal.

Geographic profiling, as defined in environmental criminology, aims to use the spatial information of crime incidents to identify the most probable location of a criminal. It can be thought of as a decision support mechanism for solving serial crimes. Its main purpose is to decrease the area of interest, improve police reaction time and reduce the overall effort to find a criminal. Decreasing the area of interest means identifying a smaller area used by the criminal as his/her base of operations; this base can be the criminal's home, work or some other area that he/she visits frequently. The method for decreasing the area of interest depends on two concepts, (i) distance decay and (ii) buffer zone. Distance decay refers to the idea that crimes will decrease in frequency the farther away an offender travels from home. In other words, distance decay is a geographical expression of least effort (time, money, etc.) and results when an offender shows preference for closer crime sites [1]. For example, 70% of arsons occur within two miles of an arsonist's home [20].

On the other hand, criminals tend not to commit crimes too close to home as this may constitute a threat to their anonymity. This area around a criminal's home is referred to as the buffer/comfort zone [1, 17]. Geographic profiling uses the opposing

effects of the buffer zone and distance decay to predict the home of a criminal and usually denotes these crime patterns as doughnut hole patterns as shown in Figure 1.3.

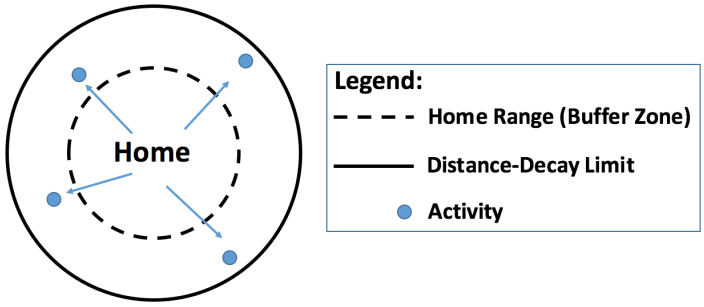


Figure 1.3: Criminals typically do not commit crime close to home (inside the dashed circle) and do not travel too far (beyond the ring) [1].

Currently, finding doughnut hole patterns in environmental criminology is a labor-intensive task since these are often searched for manually (i.e. with paper maps and pins) or using risk mapping tools (e.g. Rossmo's formula [17], Rigel Software [21]), which are not computationally efficient. Nevertheless, recently data volume is growing much faster than the number of human analysts [22]. For example, crime reports in a large city often exceed one million per year [23]. Thus, there is a growing need for geospatial tools to assist human analysts at large agencies (e.g. state and federal level).

1.4 Challenges

Handling geospatial data with an interdisciplinary approach raises a number of challenges.

Computational Challenges: Spatial data imposes huge dataset sizes (e.g. petabytes of GPS tracks and satellite imagery) due to the finer resolution of data that are collected by ever-improved data collection devices. These datasets impose an exorbitant computational cost for the current approaches. Thus, the proposed approaches should incorporate computer science to scale up to large datasets.

Statistical Challenges: Analytics on spatial data may output too many spatial patterns (e.g. hotspots) that may include false positives (e.g. patterns that occurred by

chance). This is not desirable in some application domains. For example, falsely identifying a location as a hotspot of a malicious event (e.g. disease, fatality, crime, etc.) may result in stigmatizing, economic loss, along with social/political challenges. Therefore, geospatial data science tools need to use statistics to eliminate chance patterns. Using traditional statistical techniques to eliminate chance patterns is non-trivial since these are designed for high dimensional vector space, not geographic space. Another approach may be to use geostatistical tools to eliminate chance spatial patterns. However, such tools assume the study area is an isotropic Euclidean space (i.e., every location is treated equally). Yet, human activities/events often violate this isotropic Euclidean space assumption due to underlying geographic features (e.g., road networks that humans travel on). For example, people mostly travel on roads, which means the road network is an underlying geographic feature that should be accounted for. Moreover, the isotropic space assumption does not hold in certain domain-specific geographic theories (e.g., routine activity theory in criminology). For applicability in specific domains, geospatial data science methods need to be able to accommodate such theories.

Mathematical Challenges: Geospatial data science tools should take the correctness and completeness of the output into account. Correctness generally refers to the output that aligns with the input thresholds and requirements. In other words, correctness means all output satisfy the input criteria. Completeness means that all patterns that satisfy the criteria are returned. Thus, if there exists a pattern that satisfies the input criteria, it is not missed. Since completeness is hard to accomplish, many approaches use some simplifying pattern definitions which may cause missing patterns. For example, although the mathematical definition of a circle has three parameters, namely two center coordinates (x, y) and a radius (r), SaTScan uses a simple enumeration consisting of just two points, one at the center, and one on the perimeter, to reduce the infinite number of possible circles. Since this approach may miss some patterns which satisfy the given thresholds, it is not complete. Therefore, there is a need for tools which can use the mathematical parameters of pattern shapes to ensure solution completeness. Finally, current tools are affected by the scale of a study area. For example, a hotspot detection method may miss spatial patterns at a county level but not at a state level even when the input activities/events are the same. Thus, there is a need for tools which are not affected by different scales.

1.5 Thesis Contributions

In this thesis, geospatial data science problems are investigated from a transdisciplinary data science perspective encompassing mathematics, statistics and computer science disciplines which usually work in a siloed manner. In addition, the proposed approaches, i.e. ring-shaped hotspot detection; geographically robust hotspot detection; and network hotspots with holes incorporate some domain specific theories. The algorithms are not only robust, but they also address the limitations of applying "black box" solutions to domain-specific problems. Next, each chapter is briefly introduced.

- Chapter 2 takes a wide-lens perspective on geospatial data science, considering it as a transdisciplinary field comprising statistics, mathematics, and computer science. This approach aims to reduce the redundant work across disciplines as well as to begin to define scientific boundaries of geospatial data science. Too often geospatial tools are seen as black box solutions for any kind of geospatial problem. This chapter also investigates the challenges arise from black box approach as well as the trade-offs across the three disciplines. The chapter begins with introducing the geospatial data science as a term, discusses previous problems as well as the issues that are related to individual scientific disciplines. Then it provides insights for geospatial data science to solve those problems.

- Chapter 3 discusses a novel hotspot detection technique called "Ring-Shaped Hotspot Detection" which uses ideas from our three disciplines: e.g. "ring shape" from geometry and a mathematical definition of rings to maintain correctness and completeness, spatial scan statistics from statistics, and a prune and refine framework from computer science. This approach also incorporates domain specific theories i.e. routine activity and crime pattern theory. Informally, given a set of activity points (e.g., crime, disease locations), Ring-Shaped Hotspot Detection (RHD) finds ring-shaped areas where the concentration of activities inside is significantly higher than that outside. RHD is societally important for applications such as environmental criminology, epidemiology and biology to investigate patterns of evasion. RHD is computationally challenging because of the large number of candidate rings, non-monotonic interest measure and cost of the statistical significance test. Previous approaches (e.g. spatial scan statistics tools) focus on

simply-connected shaped areas (e.g. circles, rectangles) and can not detect statistically significant rings. In the chapter, a novel algorithm is proposed to discover statistically significant ring-shaped hotspots based on the ideas of dual grid based pruning and best enclosing ring refining. Theoretical evaluation proves that the proposed approach is a correct approach (i.e. all outputs satisfy input thresholds) to detect ring-shaped hotspots. A case study on real disease data shows that the proposed approach finds ring-shaped hotspots which were not detected by the existing techniques. Cost analysis and experimental results on synthetic data show that the proposed approach with algorithmic refinements yields substantial computational savings.

- Chapter 4 takes state-of-the-art circular hotspot detection methods a step further to provide a robust way of generating candidates by using the mathematical definition of circles in its enumeration method. The proposed approach uses diffusion theory and is not affected by the sparseness of data and thus provides a richer enumeration space compared to the related work. More specifically, given a set of points in two dimensional space, a minimum radius, a minimum log likelihood ratio and a significance threshold, Geographically Robust Hotspot Detection (GRHD) finds hotspot areas where the concentration of points inside is significantly high. The GRHD problem is societally important for many applications including environmental criminology, epidemiology, etc. GRHD is computationally challenging due to the difficulty of enumerating all possible candidate hotspots and the lack of monotonicity property for the interest measure, namely the log likelihood ratio test. Related work may miss hotspots when hotspots are divided by geographic barriers (the road network, rivers etc.) or when hotspot centers are close to parks, lakes, mountains, etc. To address these limitations, a novel approach is proposed based on two ideas: cubic grid circle enumeration and a grid log likelihood ratio upper bound. A case study on real crime data shows that the proposed approach finds hotspots which cannot be discovered by the related work. Experimental results show that the proposed algorithm yields substantial computational savings compared to the related work.

- Chapter 5 investigates ways to overcome the issues that assume the study area

as an isotropic Euclidean space (i.e., every location is treated equally). Human activities/events often violate this isotropic Euclidean space assumption due to underlying geographic features. For example, people mostly travel on roads, which means the road network is an underlying geographic feature that should be accounted for. From a domain perspective, network hotspots with holes may occur when a path is used frequently by the activity/event source (e.g. a serial criminal) to travel to the location of the activities, resulting in a sparser concentration of activities on the path but higher concentration of activities on the road segments around the path. Informally this problem can be defined as follows; given a spatial network and a collection of activities (e.g. crime locations), the problem of Mining Network Hotspots with Holes (MNHH) finds network hotspots with doughnut shaped spatial footprint, where the concentration of activities is unusually high (e.g. statistically significant). MNHH is important for societal applications such as criminology, where it may focus the efforts of officials to identify a crime source. MNHH is challenging because of the large number of candidates and the high computational cost of statistical significance testing. Previous work focused either on geometry based hotspots (e.g. circular, ring-shaped) on Euclidean space or connected subgraphs (e.g. shortest path), limiting the ability to detect statistically significant hotspots with holes on a spatial network. This chapter proposes a novel Network Hotspot with Hole Generator (NHHG) algorithm to detect network hotspots with holes. The proposed algorithm features refinements that improve the performance of a naïve approach. Case studies on real crime datasets confirm the superiority of NHHG over previous approaches. Experimental results on real data show that the proposed approach yields substantial computational savings without reducing result quality.

- Finally, chapter 6 concludes the thesis findings and identifies related areas that remain open for future research.

Chapter 2

Geospatial Data Science: A Transdisciplinary Perspective

2.1 Introduction

Despite the significant growth in geospatial data science applications, there is a limited progress on scientific foundations of the geospatial data tools. Due to the lack of scientific approaches, it is often more difficult to develop reliable and trustworthy geospatial models and tools. We propose a definition of geospatial data science as a scientific process of extracting valuable information from raw geospatial data with reasonable effort. The specific properties of geospatial data, its volume, variety and velocity and the implicit but complex spatial relationships limit the applicability of traditional data science methods. In this chapter, we explore the emerging field of geospatial data science from a transdisciplinary perspective across the three closely related scientific disciplines of statistics, mathematics and computer science. Our proposed definition aims to reduce the redundant work across siloed disciplines and promote better understanding of the limits of geospatial data science as well as the expectations via examples.

This chapter provides a transdisciplinary scientific perspective for the geospatial data science which promises to create new frontiers for the geospatial problems which were previously studied with a trial and error approach. A well-known example from the past illustrates how rigorous scientific methods may change a field. Alchemy, the medieval forerunner of chemistry, once aimed to transform matter into gold [24]. Alchemists

worked tirelessly for years trying to combine different matter and observe their effects. This trial and error process was successful for finding new alloys (e.g., brass, bronze, etc.) but not for creating another metal, i.e., gold. Later, the science of chemistry showed the chemical reactions and their effects on elements, and successfully proved that an element cannot be created by simply melting and combining other elements. We see similar unrewarded efforts [25, 26] in the current trial and error approach to geospatial data science. We believe that research in the field needs to be conducted more systematically using methods scientifically appropriate for the data at hand. This chapter investigates geospatial data science from a transdisciplinary perspective to provide such a systematic approach with the collaboration of scientific disciplines namely mathematics, statistics and computer science.

2.1.1 Motivation

Over the last decade, there has been a significant growth of cheap raw geospatial data in the form of GPS trajectories, activity/event locations, temporally detailed road networks, satellite imagery, etc. [27, 28]. These data, which are often collected around the clock from location-aware applications, sensor technologies, etc., represent an unprecedented opportunity to study our economic, social, and natural systems and their interactions.

Consequently, there has also been rapid growth in geospatial data science applications. Often, geospatial information retrieval tools have been used as a type of "black box", where different approaches are tried to find the best solution with little or no consideration of the actual phenomena being investigated. Such approaches can have unintended economic and social consequences. An example from computer science was Google's "Flu Trends" service, begun in 2008, which claimed to forecast the flu based on people's searches. The idea was that when people have flu, they search for flu-related information (e.g., remedies, symptoms). Google claimed to be able to track flu trends earlier than the Center for Disease Control. However, in 2013, the approach failed to identify the flu season, missing the peak time by a large margin (e.g., 140 percent) [29, 30, 31].

This fail is but one example of how the availability of a computational tool does not mean the tool is suitable for every problem. A recent New York Times article discussed

similar issues in big data analysis from the statistics perspective, concluding, "[Statistics is] an important resource for anyone analyzing data, not a silver bullet." [32].

Similarly, geospatial data science applications need a strong foundation to understand scientific issues (e.g., generalizability, reproducibility, computability and prediction limits error bounds), which often makes it difficult for users to develop reliable and trustworthy models and tools. Moreover, we need a transdisciplinary scientific approach that considers not only one scientific domain but multiple scientific domains for discovering and extracting interesting patterns in them to understand past and present phenomena and provide dynamic and actionable insights for all sectors of society [33].

2.1.2 Problem Definition

The term geospatial data science implies the process of gaining information from geospatial data using a systematic scientific approach that is organized in the form of testable scientific explanations (e.g., proofs and theories, simulations, experiments, etc.). A good example is USGS and NOAAs analysis of geospatial and spatiotemporal datasets, e.g., satellite imagery, atmospheric data sensors, weather models, etc. to provide actionable hurricane forecasts using statistics, machine learning (computer science) and mathematical models [34, 35].

The most important aspect of a scientific process is objectivity [36], meaning the results should not be affected by peoples perspectives, interests, or biases . To achieve objectivity, scientific results should be reproducible [37, 38]. In other words, using the claims in a scientific study, the results should be consistent and thus give the same results every time.

Although they vary by domain [39], for geospatial data science we provide the following steps (depicted in Figure 2.1), which can provide objectivity and reproducibility.

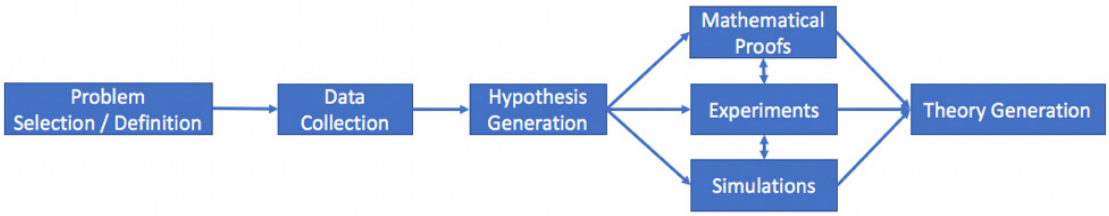


Figure 2.1: Steps of geospatial data science.

The first step is the selection of a phenomenon to explain scientifically. In other words, we decide which problem we want to explain. Next, sufficient data about the phenomenon is collected to generate a hypothesis. The important aspect of this step is that hypothesis generation should be objective and not biased by scientists' perspectives or interests. Experiments and simulations are then done to test the hypothesis. If the hypothesis survives these tests, then a theory can be generated. Note that in some domains, theories can be validated by mathematical proofs, and then confirmed by experiments and simulations. Thus, scientific methods differ slightly from one scientific domain to another. This scientific process will also draw boundaries of predictability just as chemistry drew boundaries for creating matter (i.e. gold). Depending on the data in hand, non-stationarity in time may impact the success of predictability. Thus, past events may not always help predict the future. Similarly, black swan events, where the occurrence of a current event deviates from what is expected, may escape the notice of individual disciplines [40]. The proposed transdisciplinary approach encourages us to investigate such events for better understanding the cause and predictability of black swan events with a scientific approach.

2.1.3 Challenges

Geospatial data science poses several significant challenges to both current data scientific approaches as well as individual scientific disciplines. First, the increasing size, variety and update rate of geospatial data exceed the capacity of commonly used data science approaches to learn, manage, and process them with reasonable effort [41, 42]. For example, vehicle trajectory datasets that are openly published on the Planet GPX web site include trillions of GPS points, each of which carries longitude, latitude, and time

information [43].

Second, geospatial data often violate fundamental assumptions of individual traditional scientific disciplines. For example, in statistics, the independent and identically distributed (i.i.d.) assumption of random variables, and the stationarity assumption (whereby the mean, variance and autocorrelation are assumed to be stationary) do not hold for geospatial data [42]. Similarly, in mathematics, regions with indeterminate boundaries may not be represented with traditional topology and geometry although in a geographical space indeterminate boundaries are needed since neighborhoods or urban areas often do not have determinate (strict) boundaries [44, 45]. Also, graphs in mathematics cannot be used to represent spatial networks (e.g., road networks, rivers, etc.) since these networks have location information as well as node specific constraints (e.g., turns, traffic lights, etc.) [46]. In addition, computer science often deals with one dimensional data while geospatial data often has two, three or more dimensions. A simple example is "sorting". In computer science, sorting may be done in one-dimensional vectors. However, there is no simple notion of sorting multidimensional geospatial data [47].

A third challenge is that, due to imperfect data collection devices, geospatial datasets often include missing or erroneous data [48]. To make things more complicated, there are concerns from users about geo-privacy [49]. Thus, it is hard to provide robust approaches that are generalizable.

Finally, the siloed nature of statistics, mathematics, and computer science research leads to redundant and often incomplete work on data science problems.

2.1.4 Trade-Offs

Taking a transdisciplinary view of geospatial data science means we must deal with the well-known trade-offs within individual disciplines, as well as with the many trade-offs across disciplines.

Intra-Disciplinary Trade-offs: An example in statistics is the trade-off between bias and variance, as shown in Figure 2.2. A bias error occurs when wrong assumptions are used with the training dataset. In other words, during model learning we may be overly cautious, causing our model to under-fit the data, which in turn leads to a high prediction error rate. Variance error comes from the fact that even small variances in the training data are considered for model building. Such an approach may cause

overfitting as well as unnecessarily complex model building and thus poor prediction performance.

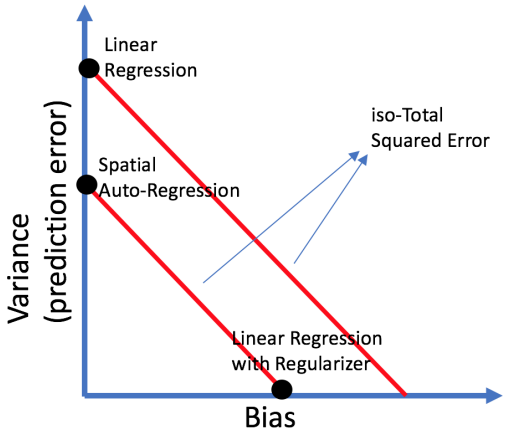


Figure 2.2: The statistical bias and variance.

An example within the discipline of computer science is the trade-off between memory storage and response time. For example, a shortest path computation using Dijkstra's algorithm [50] iteratively traverses the nodes and edges of the graph to compute the shortest path. An alternative approach may be based on pre-computing and storing the shortest paths in a database with an index on the pairs of start-node and destination. This approach will simply use the index to retrieve the pre-computed shortest path to quickly answer queries. The computation cost for shortest paths will be much lower; however, it will require much larger storage for the database of pre-computed paths. The computer science literature includes many algorithms, e.g., hierarchical Routing and contraction hierarchies [51], which explore the trade-off between storing a subset of pre-computed paths and on-the fly computation. Another computer science example from distributed systems in computer science is the CAP theorem [52], which states that one must choose between consistency and availability where the third concern is the partition tolerance.

Beyond the trade-offs within individual disciplines, there are new transdisciplinary trade-offs to consider across mathematics, statistics, computer science and data-driven sciences (referred to as domain sciences).

Data-Driven Domain Science Interpretation and Statistics (Uncertainty Quantification): Data-driven domain science interpretation and statistical uncertainty quantification have different objectives. For example, in the land cover classification problem, a decision tree [53,54] or random forest [55] approach may be used to classify remote sensing imagery to land cover type (e.g. wetland, dryland, forest, urban, rural, etc.) since the resulting models (i.e. decision trees or random forests) are relatively easy for domain scientists to interpret. However, neither the decision tree nor random forest approaches quantify uncertainty or provide a statistical confidence level for predicted land-cover classes. The alternative method is using statistical approaches such as Bayesian classifiers [56]. These may provide uncertainty quantification and statistical confidence but the results are not as easy to interpret due to their numerical nature. Thus, there is a need for approaches that will provide uncertainty quantification as well as ease of domain interpretation.

Computer Science and Statistics: Computational approaches such as data mining and machine learning tools often provide computational scalability but they may not quantify uncertainty as depicted in Figure 2.3. For example, the K-means algorithm [57] for clustering, is computationally efficient as it converges quickly to a local minimum on the error surface. However, it does not quantify statistical confidence in the discovered clusters. For example, it cannot determine whether the clusters discovered by K-means are better than those achieved by a random partitioning of the data set. In addition, it does not provide guarantees on the solution quality. For example, it doesn't tell us how the quality of a local minimum recommended by the K-means procedure compares with the quality of a global minimum on the error surface? On the other hand, the expectation maximization (EM) approaches [58] may iteratively converge to a global optimum solution; however, they seldom provide guarantees on computational cost. They cannot answer questions such as, "Is it guaranteed to terminate in a reasonable time (or will it run for an infinite time)? What is the computational complexity of the EM algorithm?" In addition, statistical approaches which aim to provide probability distributions as well as evaluate the results with statistical significance levels often require hypothesis testing [59], which increases the computational cost. Therefore, new research is required to provide computational scalability and statistical uncertainty quantification at the same time.

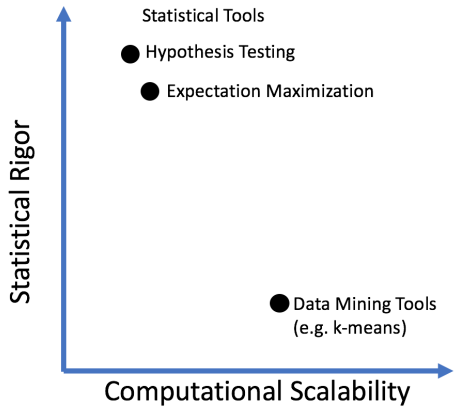


Figure 2.3: The trade-off between computational scalability and statistical rigor.

Mathematics and Statistics: A pure mathematical optimization approach to estimate parameters of a statistical (or machine learning) model may lead to overfitting [60], which may cause the model to perform poorly on generalization for prediction on unseen datasets. Moreover, it may cause many statistical models (e.g., regression, decision trees) to become excessively complex and hard to interpret. For example, in a regression model, given any set of data points, it is possible to find a polynomial function that exactly passes through each point. This may cause overfitting and reduce the prediction power of the model, since the dataset may have noisy points that bias the results. In summary, there is a need for tools that preserve statistical interpretation and mathematical completeness as well as prevent statistical models from becoming overly complex.

Mathematics, Computer Science and Statistics: Mathematics and statistics often have conflicting objectives. Basically, statistical inferences often involve quantifying the uncertainties with confidence intervals and statistical significance values. On the other hand, mathematics often deals with results' completeness, optimality, etc. Many statistical methods do not guarantee mathematical properties, e.g., completeness and optimality. For example, consider SaTScan [10], an algorithm to find hotspots, i.e., circular areas within which the density of a phenomenon (e.g., disease, crime) is much higher than the density outside. This method uses a statistical measure, e.g., a likelihood ratio and p-value, to reduce chance patterns and quantify uncertainty. This software,

widely-used in epidemiology for hotspot detection, enumerates circles using pairs of points, where one point defines the center and the distance between the points defines the radius. However, this approach doesn't enumerate many other circles, such as those defined by subsets of three points. It is likely to miss circular hotspots with empty centers as it gives up mathematical completeness to reduce computational cost. There is a need for approaches that preserve mathematical completeness while providing computationally feasible and scalable solutions.

2.2 Background

Previous attempts to define geospatial data science (Table 2.1) often focused on pairs of disciplines, e.g., statistics-computer science, mathematics-computer science, etc. We argue that all three disciplines should be considered to provide an understanding of naturally-occurring phenomena. Moreover, these disciplines should operate together so that all may benefit from conceptual advances of common interest. For example, analytics on hyperspectral remote sensing imagery, which is used by earth science applications (e.g., agronomy, geology, hydrology, etc.), applies computationally efficient and statistically robust algorithms for those high dimensional (e.g., hyperspectral) geospatial datasets [61].

Table 2.1: Overview of the related work.

Approach	High Dimensional Data	Spatial Data
Siloed / Multi-Disciplinary	Statistics, Mathematics, Computer Science	Spatial statistics, Spatial Data Mining, Machine Learning
Transdisciplinary	Theoretical Foundations of Data Science Workshop	Proposed Approach

Recently, an NSF workshop on "Theoretical Foundations of Data Science (TFoDS)" [31] attempted to provide a definition of "data science" that brings these three disciplines together. The workshop identified fundamental areas where collaboration among computer scientists, mathematicians, and statisticians is necessary to achieve significant progress. However, the focus of the workshop was not geospatial data generally

but high dimensional data, and most of the discussion centered on very specific topic areas i.e. computation-statistics tradeoff, randomized numerical linear algebra, signal processing/harmonic analysis on graphs, nonconvex statistical optimization, combining physical and statistical models, mixed type and multi-modality data, applied representation theory and non-commutative harmonic analysis, topological data analysis and homological algebra, security, privacy, and algorithmic fairness [31].

Another recent development was the NSF Workshop on Geospatial Data Science in the Era of Big Data and CyberGIS [62]. Its focus was high performance computing and the computational aspects of geospatial data science. Topics included geospatial big data capabilities (e.g., LiDAR, remote sensing, location-based social media) for novel applications (e.g., urban sustainability), cloud computing, and tools for scalable geospatial data analytics. One of the goals was to formulate a core set of questions and problems of geospatial data science around these themes. The workshop addressed the geospatial data science problem from a high-performance computing perspective but did not address the broader set of questions that led us to our attempt here to define geospatial data science.

2.3 Contributions, Scope, and Outline of the Chapter

This chapter takes a wide-lens perspective on geospatial data science. We believe that geospatial data science is a transdisciplinary field comprising statistics, mathematics, and computer science, and that it should be formally considered the foundation of geospatial science. The aim is both to reduce redundant work across disciplines as well as to define the scientific boundaries of geospatial data science so it is no longer seen as a black box solution to every possible geospatial problem. In addition, we aim to lay out some of the challenges that arise from the geospatial nature of the data. Hence, in the following sections we investigate individual disciplines, their objectives as well as the challenges they face to investigate the transdisciplinary definition of geospatial data science.

Scope and Outline: In this chapter, we present geospatial data science as a transdisciplinary scientific process. The proposed approach provides a discipline-of-disciplines

perspective towards reducing redundant work and providing a more robust way to create information from raw geospatial data. In addition, our approach aims to identify the limits of geospatial data science predictability.

To emphasize the transdisciplinary perspective of geospatial data science, in the following sections we provide examples from each discipline, namely statistics, mathematics and computer science that are cross-cutting with geospatial data science. As summarized in Figure 2.4 for example, the study of indeterminate regions is both a mathematics and a spatial statistics problem. Similarly, randomized algorithms can be considered as a problem in computer science but also one that uses fundamental ideas from spatial statistics. Finally, representative problem examples that all three disciplines tackle are explained in more detail.

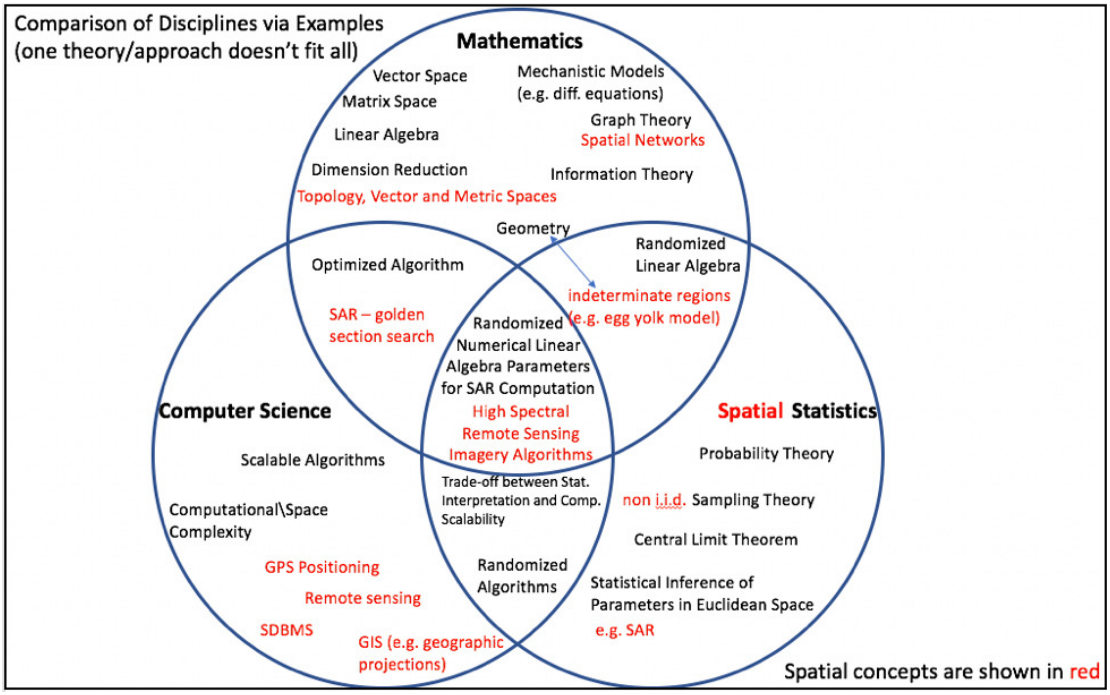


Figure 2.4: Comparison of disciplines via examples.

2.4 Statistics

2.4.1 Traditional Statistics

Statistics studies data in the context of uncertainty, and it serves as an important foundation of many data science tasks such as pattern recognition, prediction, and classification. Given observations collected from a part of a population, statistics reduces uncertainty by making inferences on the entire population. It differs from probability theory in that probability theory works with known probabilistic distributions to estimate the probability of future observations while statistics starts with a collection of past observations and estimates the unknown parameters of a probabilistic distribution to make inferences.

In statistics, data collection is performed based on sampling theory [63], which provides a scientific framework to decide the population of concern, sampling approach (e.g., random sampling), sampling size, etc. The collected observations are then used to estimate parameter values of a target distribution model (e.g., Gaussian distribution). The estimation can be performed using either a frequentist or a Bayesian approach [64]. A frequentist approach analyzes data as an integrated whole. It assumes each parameter has a fixed value that does not change over time and that can be accurately estimated as the number of observations increases to infinity. However, in real world scenarios, the number of observations is limited and there is always an uncertainty associated with the analysis given the incomplete data. In order to express this uncertainty, a frequentist approach uses a confidence interval [65] to claim a minimum expected probability (e.g., 95%) that the estimated parameters are true.

By contrast, a Bayesian approach assumes that each parameter comes from a prior distribution. It considers data as a sequence of observations and continues to update the estimation of parameters as new observations are available. Unlike a frequentist approach, a Bayesian approach captures the change or evolution of parameters over a sequence (e.g., time) of observations, and thus can further reduce the uncertainty in an inference. However, a Bayesian approach requires an appropriate prior distribution as input; otherwise it cannot give correct inferences.

2.4.2 Traditional Statistics versus Spatial Statistics

One of the most common assumptions in traditional statistics is that observations are identically and independently distributed (i.i.d.) [66]. The i.i.d. assumption is an important foundation of many data science methods. For example, in machine learning, maximum likelihood estimation [67] is used to estimate the parameter values of a given model, and the expressions of likelihood functions are often obtained based on this i.i.d. assumption (e.g., Nave Bayes classifier, Expectation-Maximization). In fact, many classic statistics theorems come from the i.i.d. assumption, such as the well-known central limit theorem [68], which states that the mean of a set of samples is approximately equal to the mean of an entire population given a sufficiently large sample size.

Although it offers great convenience in traditional statistics, the i.i.d. assumption is often violated in the geospatial domain. As the first law of geography states: "Everything is related to everything else, but nearby things are more related than distant things" [69]. This fundamental observation on geospatial data breaks the i.i.d. assumption of non-spatial data in traditional statistics. Spatial statistics deals with the phenomenon of spatial auto-correlation through careful modeling of spatial relationships among data samples. The following discusses two motivating examples of spatial statistics.

Example 1: Pearson correlation on geospatial data: Figure 2.5a shows a distribution of three types of diseases, abbreviated as TF, VSD, and ALL. Each instance of each disease has a unique ID as marked in Figure 2.5a. From the distribution, we can see each ALL instance has a nearby TF instance and VSD instance. For example, ALL1 is adjacent to TF1 and VSD1. To measure the spatial correlation among the three types of diseases, we need some parameters to express the spatial distribution. Figure 2.5b shows a boundary fragmenting the study area. For each type of disease, we can consider each fragment as a property of its spatial distribution, and each property value as the count of instances of this disease within the fragment. Suppose the fragments are concatenated into a vector following column-wise order (top-left $\rightarrow$ bottom-left $\rightarrow$ top-right $\rightarrow$ bottom-right). Thus, the vector of properties for ALL is $[0, 0, 1, 1]$, TF is $[1, 2, 0, 0]$ and VSD is $[0, 0, 1, 1]$. With this spatial modeling based on boundary fragmentation, the Pearson correlation ratio is -0.91 between TF and ALL, and 1 between VSD and ALL. This negative correlation between TF and ALL contradicts

to our observation since their spatial adjacency is broken by the boundary between fragments (Figure 3 2.5b). By contrast, the correlation between VSD and ALL is positive because the spatial adjacency between VSD and ALL instances is preserved by the arbitrary partitioning. These mutually contradictory correlations reveal the uncertainty of results when traditional statistics is trivially applied to the geospatial domain.

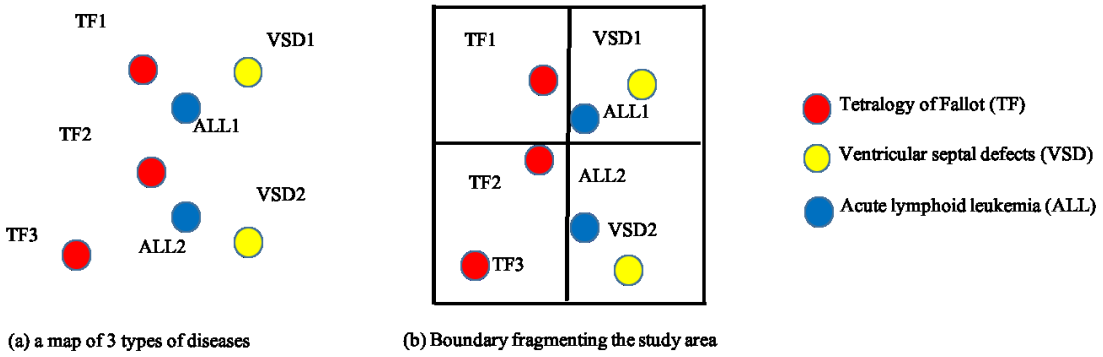


Figure 2.5: Distribution of disease.

Example 2: Agronomic field experiment design: Field experiments are used by agricultural scientists to evaluate the performance and properties of crops under different conditions (e.g., water, fertilizer) [25]. Traditional experiment designs assume observations are independent and that an expected value stays the same at different spatial locations. However, in field experiments, these assumptions are often violated since closer plants exhibit more similar properties, and soil properties vary at different locations, which lead to non-stationary expectations [25]. To address this problem, blocks are used in field experiment design to reduce the effect of spatial auto-correlation and heterogeneity. A block is a large spatial unit containing a set of plots. With a properly chosen block size, the spatial-related properties (e.g., soil type) can be assumed to be uniform within a block. Distances are added between blocks so that the spatial auto-correlation between blocks is reduced. The choice of block size and distance between blocks are critical parameters to reduce the errors caused by spatial effects. In practice, they can be determined by spatial statistical analysis.

2.4.3 Spatial Statistics

Geostatistics: Geostatistics [70] is concerned with point-reference data, which contains a set of points with fixed locations and a set of attribute values. The goal of geostatistics is to model the distribution of the attribute values and make predictions on uncovered locations. Point-reference data has several inherent properties: 1) isotropy/anisotropy; 2) second order stationarity; and 3) continuity. In the context of isotropy, uniformity is assumed in all directions, while under anisotropy, some statistical properties may vary by direction. Second order stationarity is a weaker form of strong stationarity so it is also referred to as weak stationarity. Instead of assuming a strong stationarity with invariant density of distribution, second order stationarity assumes only invariant moments (e.g., mean, variance) across a spatial domain but covariance between locations depends on the distance. The continuity property indicates the existence of spatial dependence on the data. The degree of dependence can be quantitatively measured with input distance and direction using a variogram or semivariogram. If we further assume isotropy, then the variogram simplifies to a function of distance only. With the base assumptions on point-reference data, the distribution of attribute values can be effectively modeled. A set of statistical tools are provided by geostatistics and one of the most popularly used methods is Kriging [71]. Kriging is a statistical model of interpolation that predicts attribute values at unsampled locations (e.g., water quality estimation based on observations from a set of monitoring sites). Co-Kriging [72], provides a multivariate extension of ordinary Kriging. For a set of highly correlated attributes, Co-Kriging can improve the prediction quality on a poorly sampled attribute using well sampled ones. Besides spatial auto-correlation, spatial heterogeneity also needs careful consideration in many applications (e.g., different types of underlying landscape). Special models, such as GWR (geographically weighted regression) and spline, are available in geostatistics to reflect the changes in statistical properties given the presence of spatial heterogeneity. These models deploy a local view on the data and assign higher weights to neighboring points to reduce the effect of heterogeneity.

Spatial point process: Unlike geostatistics, a spatial point process is not concerned with attribute values but with the locations of points [73], specifically, their distribution. Locations of a set of points can be generated based on different statistical assumptions (e.g., random, clustered). The most common model assumed for a spatial point process

is a homogeneous Poisson distribution, also known as complete spatial randomness (CSR). In CSR, the total number of points follows a Poisson distribution and each point is identically and independently distributed in a pre-defined spatial domain. A variant of CSR is a binomial point process, in which the only difference is a fixed total number of points. In many application domains, CSR or binomial point process is not an appropriate assumption since points may have spatial autocorrelation or inhibition characteristics. In such cases, other specialized models should be applied to better approximate the exact distribution as shown in Figure 2.6. For spatial inhibition, a Poisson hardcore process is widely used to generate a distribution that enforces mutual repulsion among points. For spatial autocorrelation, a Matern cluster process can be chosen to reflect the clustering characteristics. Similar cluster processes include Poisson cluster process, Cox cluster process, Neyman-Scott process, etc. One of the most well-known applications of a spatial point process is spatial scan statistics [74] in hotspot detection. In spatial scan statistics, chance hotspots are removed through a statistical significance test under a null hypothesis based on CSR. CSR is also used as a null hypothesis for significance testing in Ripleys K function [75], which estimates the overall clustering degree of a point distribution.

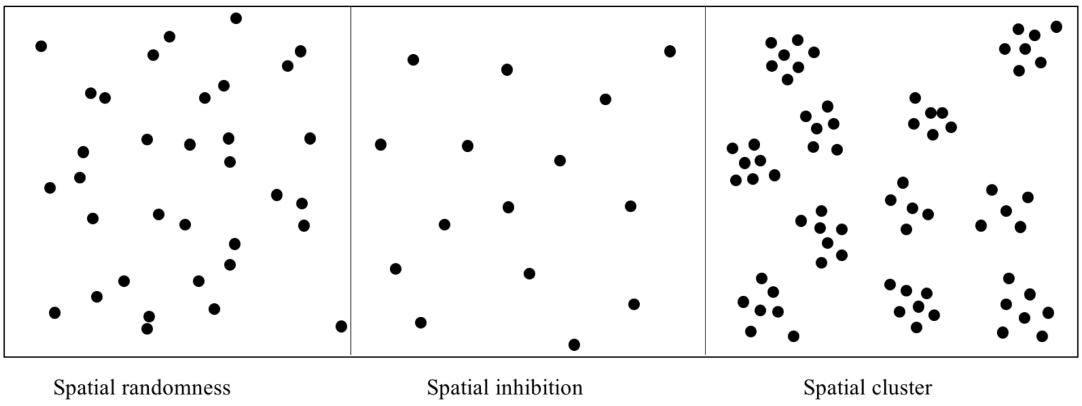


Figure 2.6: Forms of Spatial Point Processes.

Lattice statistics: A lattice is a representation of a discrete space, which is a finite collection of grid cells in a spatial domain. In this case, lattice statistics concerns statistical processes in the field model. For continuous data (e.g., polygon), a W-matrix

(continuity matrix) can be computed to transform the original data into a discretized representation based on their spatial adjacency or proximity. Lattice statistics provides a set of models [76, 77], such as Moran's I, Getis-Ord Gi*, Geary's C, Gamma index and LISA, to evaluate spatial autocorrelation on the field model. For example, Moran's I outputs an I-value within $[-1, +1]$ to reflect a positive, none or negative spatial autocorrelation in the input dataset. For value estimation and prediction, spatial autoregressive models [78] are applied on discrete data, such as Markov random fields (MRF), the simultaneous autoregressive model (SAR), and the conditional autoregressive model (CAR). MRF models the evolution process of a phenomenon based on the assumption that the property of a spatial location is spatiotemporally determined by its neighbors with additional randomness. In the CAR model, a Markov property is implied and the state of a location is affected by its direct neighbors, but not neighbors of its neighbors. This property of CAR is called spatial memoryless. By contrast, SAR does not assume any non-transitive spatial influences and considers autocorrelation in a larger spatial domain. Therefore, CAR is a more appropriate choice for use cases influenced by a local spatial process and SAR is a better assumption for a more global spatial process. Another critical issue in lattice statistics is the impact of scale on spatial analysis. With different aggregation levels of scale, the statistical analyses may have distinct results. For example, variance of income aggregated on a neighborhood level could be much smaller than that on a county level within the same state.

Spatial network statistics: A spatial network is a graph-based model with enriched spatial information (e.g., turn, capacity). In a spatial network, events or objects are mutually accessed through a set of connected edges instead of straight lines in the Euclidean space. Statistics on spatial networks is a newly emerging area which has not been as extensively studied as statistics on Euclidean space. In recent work, some statistical models for object data, such as spatial autocorrelation, interpolation, and clustering approaches have been extended to spatial networks. Spatial network statistics, as an extension of spatial statistics on Euclidean space, can better model processes in urbanized places where objects and events spread along network edges (e.g., roads, rivers). For example, in transportation planning, statistically significant hotspots of accidents needs to be identified based on network space [79].

2.5 Mathematics

Mathematics plays a critical role in all science and technology. It is fundamental to a variety of traditional subjects such as physics, chemistry, and agriculture. In data science, mathematics provides its core value in data representation and modeling as well as the logic and proofs used to validate data science approaches. In this section, we first introduce how mathematics is applied in traditional data science with a collection of examples. Then, we discuss the limitations of applying traditional mathematical models to spatial data and novel spatial models with examples.

2.5.1 Mathematics in Traditional Data Science

Data science utilizes a variety of subjects for accomplishing different data modeling and processing tasks. Many types of data can be represented using linear algebra models. Aligned two-dimensional data are typically modeled as a matrix. For example, an image channel is represented as a matrix where each element indicates the value of a pixel in the corresponding location. This representation is widely applied in precision agriculture [80, 81, 82], which uses the remote sensing data consisting of multiple image channels. Similarly, a graph can be represented as a neighborhood matrix as well, where each row corresponds with a node in the graph and the elements in that row indicate the connection from this node to all other nodes. A vector is always used to model an object that has a set of features where each features is quantized as an element in the vector. The operations on matrices and vectors also apply on the represented data. For example, the similarity between two feature vectors can be measured by the distance computed by the norms and the angle between them. Eigenvalue and Eigenvector are used for studying the behavior of Markov chains [83, 84] which has been the core idea of many approaches such as PageRank [85]. Principal Component Analysis (PCA) [86] uses Eigenvalues and Eigenvectors for reducing the dimensionality of the data. Another important application of linear algebra in data science is machine learning [87]. For example, linear regression can be modeled as a linear system which can be solved by multiple linear algebra approaches such as Gaussian elimination and multiplying by its inverse [88]. Many data science approaches are derived based on linear algebra. As an example, low rank matrix approximation based on Singular Value Decomposition

(SVD) [89, 90] is applied in data compression, classification, regression, clustering, and signal processing, etc.

Another subject in mathematics that is widely used in data science is information theory. Entropy is a concept that originally comes from thermodynamics [91] which measures the number of microscopic configurations that a thermodynamic system has. Based on the essence of entropy, entropy in information theory [92] measures the expected value of the information contained in a message or the uncertainty of the data. Data classification approaches such as decision trees use entropy to measure the information gain [93, 94, 95] between two levels of the tree which offers a quantitative guide of how the tree should grow. For example, a good growth of the tree is expected to decrease the overall entropy.

Optimization is a highly interdisciplinary subject related to both mathematics and computer science. It is applied to many critical societal applications. For example, precision agriculture researchers need to allocate each field with a type of product to achieve the optimal environmental and economic outcome, which requires to solve a multi-variable optimization problem [96, 97]. Many machine learning approaches use optimization techniques to achieve their goals such as finding the minimal value of the cost function [98, 99]. For example, gradient descent, a popular technique used for machine learning approaches, finds the minimum value of a cost function by iteratively moving along the direction of the slope [100,101,102]. Finding the slope requires solving differential equations [103] which is an important subject in mathematics. Differential equations have many other applications in data science especially for spatial data since they can be naturally differentiated into variations over space and time. For example, the Soil and Water Assessment Tool (SWAT) [104, 105] is a software that embraces a variety of environmental and agricultural models about the variations over space and time which apply differential equations.

Another topic from mathematics, which is tightly related to computer science and data science, is graph theory since many real phenomena can be naturally modeled by a graph where the vertices represent the objects, and the edges represent the relationship between objects. For example, internet can be modeled as a graph where [106,107] each web site can be a vertex and the links can be edges outgoing from this vertex. In a social network model [108, 109], vertexes represent individuals, and edges represent the

relationship between two individuals. There are also spatial data models based on graph theory. Traditionally, road networks are modeled such that the intersections are vertices and the roads are edges. A similar framework also applies to flight networks [110] and oil pipeline networks [111], but the edges become the air routes and the pipelines.

Topology studies the properties that are preserved under deformations, including stretching, twisting, and bending. Topological Data Analysis (TDA) [112] is an example of applying topology in data science whose main goal is to study the geometric characteristic of data via topology. For spatial data, they are largely used in modeling a collection of relationships between real-world spatial objects. For example, Minneapolis is inside of Minnesota state is an "inside" topological relationship.

2.5.2 Limitations of Applying Traditional Mathematical Models to Spatial Data and Novel Spatial Models

We reviewed the mathematical subjects that have been applied in data science. However, they have many non-negligible limitations when dealing with spatial data. An example comes from the metric of objects. Suppose there are two spatial objects on a two-dimensional plane, each presented by a two-dimensional coordinate; how do we order them? One straightforward way is using their distance to the origin. Another popular way is sorting by the angle between the line connecting the points and the origin and an axis (i.e., x-axis or y-axis). However, there is no natural metric to order spatial points. Thus, developing meaningful and efficient ordering metrics for spatial objects is an important and challenging research topic.

In traditional topology, spatial regions are always modeled with determinate boundaries [45, 113]. It turns out that the traditional topological relationship models always rely on the boundary. For example, the relationship inside is determined by whether a spatial region falls completely within another region, and the relationship touches is determined by whether the boundaries of two spatial regions are overlapped but not their inside such as two neighbor states. However, in real-world scenarios, many spatial regions are surrounded by indeterminate boundaries. For example, it is impossible to clearly define the boundary between urban and rural areas. Research has been done to narrow the gap between real-world relationships between spatial regions and traditional topological models. One of the most popular models is the "Egg-Yolk" [45] model

which provides a representation of regions with indeterminate boundaries based on the framework of "RCC-theory" [113, 114]. It is a logically consistent and computationally tractable model that represents a spatial region with indeterminate boundary by pairs of regions with determinate boundaries (i.e. crisp regions).

Traditionally, spatial data have always been modeled on Euclidean space. This works well for many problems such as those related to air and ocean. However, there are many types of events/activities that can be associated with the transportation networks such as traffic and crimes. Using traditional models based on Euclidean space significantly affects the precision of the model and thereby the quality of the solution. As an illustration, Figure 2.7 shows a map of the campus of University of Minnesota. The east and west banks are connected by a bridge over the Mississippi river. The Euclidean distance between the two red dots is short, yet the network distance computed from the shortest path is much longer [50]. Suppose we want to approximate the travel time between these two dots, the error using Euclidean distance will be huge.

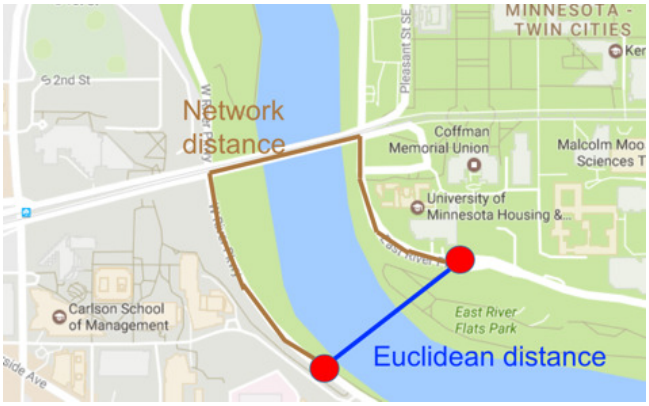


Figure 2.7: An illustrative example showing the difference between distances in Euclidean and network spaces.

Models based on network space can give a better distance approximation to some extent. In the simplest way, a transportation network can be represented as a graph, where each intersection is a vertex and each road segment is an edge associated with a value representing the travel cost of that edge. The travel cost could be assigned various values such as road distance, travel time, or fuel consumption. However, traditional graph models have several major limitations dealing with the massive information

contained in spatial networks. For example, the traditional models simply treat intersections as vertices but do not model the turns. However, left turns are usually less costlier than right turns if driving on the left side [115]. This cost difference can have a large impact for real-world applications. For example, UPS saved 10 million gallons of fuel, emits 22 thousand tons less carbon dioxide, and delivers 350 thousand more packages every year by avoiding left turn since the year of 2004 [115]. Figure 2.8 shows an example of modeling the turns. The left figure shows a patch of map in Dinkytown, Minneapolis. The middle figure shows a traditional model describing the streets where the vertices are the intersections and the directed edges are the roads. The right figure shows an example of modeling the intersection at N5 while keeping the turn information by a set of connects. The other approaches include using hyper-edges along with hyper graphs and annotating the graph with turn information.

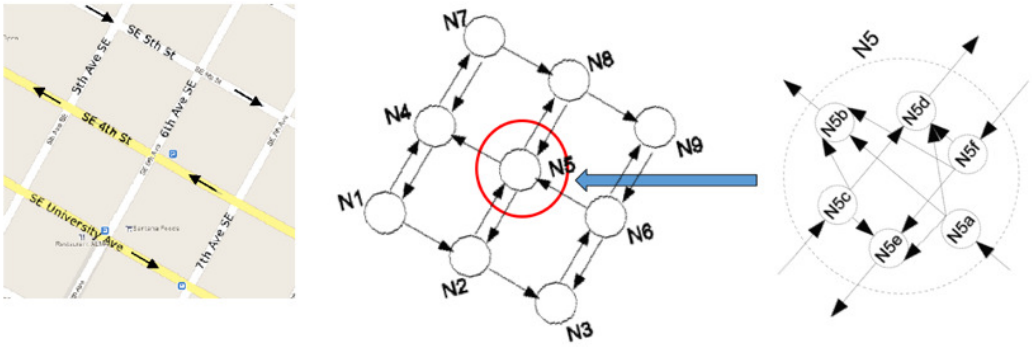


Figure 2.8: An example of modeling the turns of a transportation network.

Another consideration is that in traditional graph models, each edge is associated with a static value, which is not enough for modeling dynamically changing travel costs. For example, the travel time for a highway around downtown varies a lot during rush hour and non-rush hours. A Time-Expanded-Graph (TEG) [116] is one of the approaches that is capable of modeling the dynamically changing weights on edges. Figure 2.9 shows an example of TEG of a graph consisting of four nodes. The left side shows the varying travel times associated with each edge in four timestamps. The right side shows the TEG modeling this graph where each column represents the set of vertices in one timestamp. Each edge connects the nodes that are reachable within

a certain time. For example, edge $< A1, B3 >$ indicates that if departing from Node A at timestamp 1, one can arrive to Node B at timestamp 3. As can be seen, TEG is much more complex compared to traditional static graph model, and thus leads to harder computational challenges.

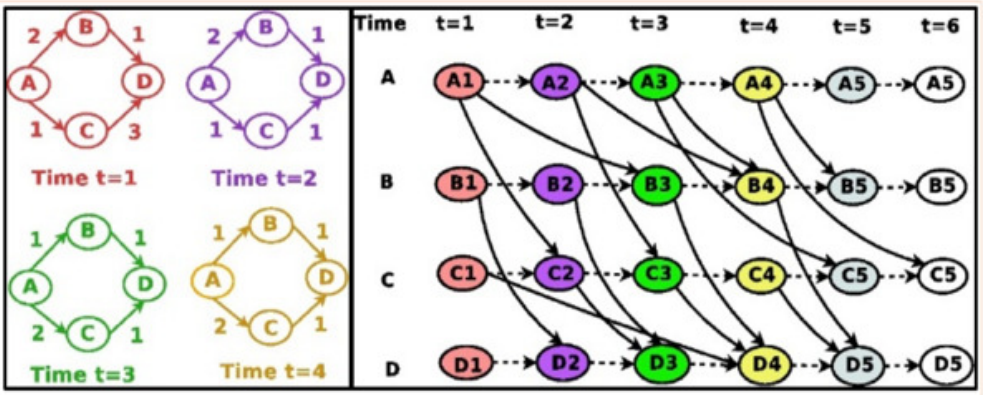


Figure 2.9: An example of Time-Expanded-Graph (TEG).

Moreover, in a traditional graph, the edges are considered atomic which cannot be further fragmented. This feature works when using graphs to model non-spatial networks such as webpage networks and social networks since the edges virtually represent the connection between objects. However, for spatial data models, the edges represent roads on which the activities happen. If we treat edges as atomic, the location information of the activities will be lost. A novel model called dynamic segmentation [117, 118] has been proposed to handle this limitation. In dynamic segmentation, the original graph is segmented based on the locations of activities on the edges. Figure 2.10 shows an illustrative example, using traditional graph model, edge $< N1, N2 >$ is atomic and the location information of activities $A1$, $A2$, $A3$, $A4$ are lost. In dynamic segmentation, edge $< N1, N2 >$ is segmented to $< N1, A1 >$, $< A1, A2 >$, $< A2, A3 >$, $< A3, A4 >$, $< A4, N2 >$, and thus the locations of the activities are kept. Dynamic segmentation outperforms traditional graph models especially when dealing with activities located on a portion of a long road segment such as an highway. For example, in linear hotspots detection, dynamic segmentation helps increase the precision of the hotspots and reveal the ones that are missed using traditional models.

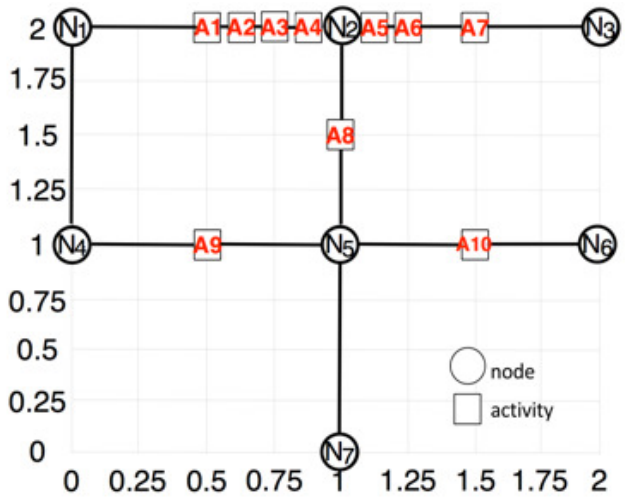


Figure 2.10: An illustrative example of dynamic segmentation.

2.6 Computer Science

In this section, we start by discussing core questions and goals of computer science. We then present some of the concepts, theories, models and technologies that computer science has contributed to the field of data science. Finally, we discuss the limitations of traditional data science with respect to spatial data and the computer science accomplishments that have attempted to address these limitations towards the realization of geospatial data science.

2.6.1 Core Questions and Goals

Computer Science is both a scientific and an engineering discipline [119]. Hence, Computer Science contributions encompass both theory (e.g. studying properties of computational problems) and practice (e.g. systems design, data mining). However, the scientific aspect of Computer Science is different from physical disciplines and is closer to mathematics where the goal is to create representation models and study their properties. Many traditional data science questions are studied within the field of computer science. Examples of these questions include: Is a given problem decidable (i.e. computable)? Is there a polynomial time algorithm to solve a given problem? What is the most efficient algorithm to perform the computations? Can the algorithm scale to large

datasets? What is the trade-off between the computational scalability of the algorithm and the statistical rigor?

2.6.2 Concepts, Theories, Models and Technologies

Many computer science concepts are leveraged in data science. Two major concepts are the design of appropriate data structures and algorithms. Data structures are ways of storing data so that they can be efficiently used. Examples of common data structures include arrays, queues, linked lists, trees, and graphs. Algorithms [120] are well-defined computational procedures that take a value (or a set of values) as input, and produce a value (or a set of values) as output to solve a given problem (e.g., searching, sorting, finding the shortest path between a source and a destination node in a transportation graph).

In addition, Computer Science theories are also leveraged in traditional data science. For instance, the Computational Complexity theory [121] focuses on classifying computational problems according to their inherent difficulty. The theory introduces mathematical models and techniques for studying computational problems and is usually used to establish proofs that for a given problem, no algorithm can run faster than the current one.

Another major accomplishment relevant to data science is the development of database management systems (DBMS), general-purpose software systems that facilitate the processes of defining, constructing, manipulating, querying and sharing databases among users and applications [122]. The most common type of DBMS is relational database management systems (RDMS), which adopt the relational data model first introduced in [123]. In this model, the database is represented as a collection of relations (i.e., tables), based on the concept of mathematical relations. Each row (i.e., tuple) typically represents information about a real-world entity or relationship, while each column represents a given attribute describing that entity. SQL is the standard query language for commercial RDBMSs and is based on relational calculus. Relational algebra is also used as the basis of query processing and optimization in RDBMS [122]. Examples of popular commercial RDBMS' include IBM's DB2, Oracle, Sybase DBMS, SQLServer, Access, MySQL and PostgreSQL.

Cloud computing platforms make possible the processing of large data volumes in

an efficient manner. Existing approaches to cloud computing provide a general framework for a distributed file systems (e.g., Google file [124] system and HDFS [125]) and processing these data sets based on replicas of data blocks (e.g., map-reduce [126], Hadoop [125] and Spark [127]. For instance, the left side of Figure 2.11 shows the Intel distribution for Apache Hadoop software components [2]. It also shows many components running on top of the HDFS for distributed processing (MapReduce), workflow (Oozie), scripting (Pig), machine learning (Mahout), SQL queries (Hive), and column store storage (HBase). In addition to cloud computing platforms, there are also many existing high-performance scientific computing cluster technologies as depicted in the right side of Figure 2.11. These computing technologies include parallel file systems (e.g., Lustre), batch schedulers (e.g., SLURM), MPI and OpenMP for internode and intra-node parallelism, and numerical and domain specific libraries, on-top of which applications are usually developed using languages such as FORTRAN and C/C++ [3].

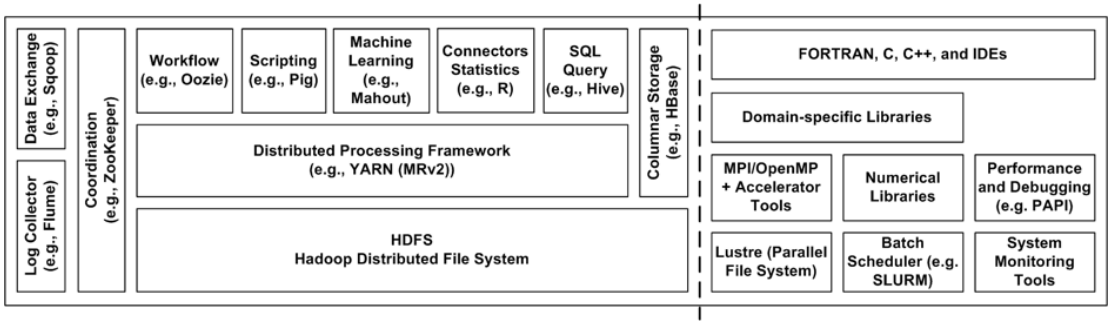


Figure 2.11: Intel Distribution for Apache Hadoop software components compared with the high-performance computing ecosystem. Figure adapted from [2] and [3].

Another major area of interest in computer science is data mining. Data mining refers to the discovery and extraction of new and useful information (e.g., patterns or rules) from large amounts of data. Typically, data mining has been mainly concerned with the computational complexity of proposed discovery algorithms and less concerned with the statistical robustness of these algorithms (e.g., bias, inference confidence, etc.). Common data mining tasks include the discovery of association rules (e.g., which grocery store items are frequently bought together?). Algorithms such as Apriori [128] and FP-growth [129] have been proposed for efficiently mining association patterns. Data mining

tasks also include the classification problem (e.g. classifying a pixel in a picture as dry land versus wetland based on other pixel properties). Popular classification models include decision trees for which computational algorithms such as ID3 [94] have been proposed.

2.6.3 Limitations of Traditional Data Science for Spatial Data and Related Computer Science Accomplishments

Now we review the limitations of traditional data science with respect to spatial data by focusing on three main areas of accomplishments, namely, spatial databases, spatial cloud-computing platforms and spatial data mining.

Spatial Databases: Applications such as precision agriculture require special database support to store, process and query spatial data (e.g., storing and querying the polygons representing farm plots). Before the development of spatial databases, spatial queries (e.g., Which galaxy pairs are within 30 arc seconds of each other? Which houses are most likely to be flooded by global warming-induced sea-level rise?) required extensive programming and suffered from long computation times due to the mismatch between 2D spatial data and 1D data types (e.g., number) and indexes used by traditional database systems (such as B+ Tree) [42]. In addition, a naive collection of spatial data types is inadequate for multistage queries since the result of some queries (such as the union of disjoint polygons) cannot naturally be represented as a point, line, or polygon. Spatial databases (such as Oracle Spatial and PostGIS) introduced spatial data types (such as OGIS simple features [130]), operations (such as inside and distance), spatial data structures (such as Voronoi diagrams), and algorithms (such as shortest-path, nearest-neighbor, and range query) to represent and efficiently answer multistage concurrent spatial queries [42]. The reduced programming effort resulted in more compact code and quicker response times. In addition, spatial indexes have also been added. Representative indexes for point objects include Grid files, multidimensional grid files [131], Point-Quad-Trees, and Kd-trees [132]. Representative indexes for extended objects include the R-tree structures [133]. The R-tree is a height balanced natural extension of the B+ tree for higher dimensions [134]. Objects are represented in the R-tree by their minimum bounding rectangles (MBRs). Non-leaf nodes are composed of entries of the form (R, child-pointer), where R is the MBR of all entries contained in the child-pointer.

Leaf nodes contain the MBRs of the data objects. To guarantee good space utilization and height-balance, the parent MBRs are allowed to overlap. Many variations of the R-tree structure exist whose main emphasis is on discovering new strategies to maintain the balance of the tree in case of a split and to minimize the overlap of the MBRs in order to improve the search time.

Spatial Computing Platforms: Support for spatial data (e.g., spatial indexes) was also needed in cloud computing platforms to improve the I/O cost of spatial queries (e.g., retrieving a set of farm polygons within a given spatial range). Representative efforts for supporting spatial data in existing cloud computing platforms include [135]: 1) SpatialHadoop [136], which is a MapReduce extension to Apache Hadoop designed specially to work with spatial data by providing specialized spatial data types, spatial indexes, and spatial operations and 2) Hadoop GIS, a high performance spatial data ware- housing system over MapReduce [137]. 3) GeoSpark [138], the spatial extension for Apache Spark. Research on parallel R-tree construction on a GPU is also ongoing [139]. At the Hadoop Distributed File System (HDFS) level, SpatialHadoop [136] and Hadoop GIS [137] have added spatial indexes. At the scripting layer (e.g., Pig), SpatialHadoop has added Open Geodata Interoperability Specification (OGIS) data types and operators. GIS on Hadoop (Pang et al. 2013) has also added OGIS data types and operators at the SQL query level (e.g., Hive). In addition to the spatial extensions of Hadoop, the GeoSpark [138] system has also extended Apache Spark with a set of Spatial Resilient Distributed Datasets (SRDDs) that can efficiently load, process, and analyze SBD. GeoSpark also introduced spatial indexes, spatial geometric operations that follows the Open Geospatial Consortium (OGC) standard, and spatial query operations for SBD.

Spatial Data Mining: Spatial data mining [140] is the process of discovering interesting and potentially useful patterns from spatial databases. For example, in precision agriculture, given a UAV-captured image of a farm, one may want to classify the set of pixels in the image based on crop type (e.g., corn, soybean, etc.).

However, the complexity of spatial data and implicit spatial relationships limits the usefulness of conventional data mining techniques for extracting spatial patterns [28]. Specific features of geographical data that preclude the use of general purpose data mining algorithms are: (1) the spatial relationships among the variables; (2) the spatial

structure of errors; (3) the presence of mixed distributions as opposed to commonly assumed normal distributions; (4) observations that are not independent and identically distributed (i.i.d.); (5) spatial autocorrelation among the features; and (6) nonlinear interactions in feature space. Figure 2.12 [4] illustrates an example of these limitations, namely the existence of spatial auto-correlation, by comparing the output of traditional decision trees with spatial decision trees for classifying wetland and dryland pixels in a satellite image taken in the city of Chanhassen, MN. The classification model used 12 continuous explanatory features as input including multi-temporal spectral information (R, G, B, NIR bands), and Normalized Difference Vegetation Index (NDVI) for the years 2003, 2005, and 2008. Figure 2.12a shows the output of the traditional decision tree algorithm. The legend of the prediction maps is shown in Figure 2.12c. The green and red colors represent correctly classified wetland and correctly classified dryland. The black and blue colors represent false wetland and false dryland.

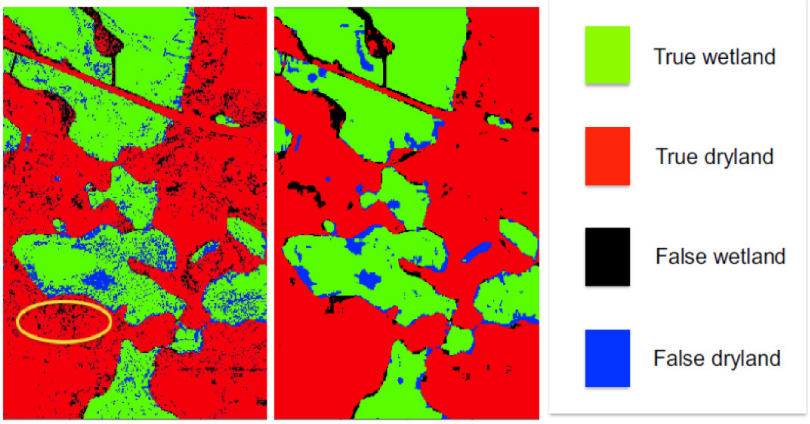


Figure 2.12: Traditional Decision Tree versus Spatial Decision Tree Output for Classifying data from satellite imagery [4].

As shown in Figure 2.12a, the prediction of the traditional decision tree model has lots of salt-and-pepper noise due to high local variation of features within patches of the same class. For example, the area in the yellow circle is a dry land area consisting of trees. The black salt-and-pepper noise pixels inside the yellow circle corresponds to locations without tree coverage. These pixels are mis-classified as wetland here due to

the i.i.d assumption. In contrast, the spatial decision tree employs a model where the tree traversal for a location is based on not only local bus also focal (i.e. neighborhood) properties of the location thus accounting for spatial auto-correlation. Hence, as shown in Figure 2.12b, the spatial decision tree model captures the local variations results in much less salt-and-pepper noise in the same area.

In addition, the spatial data mining literature includes spatial hotspot analysis [10, 141, 142], discovering spatial co-location and co-occurrence patterns [143, 144, 145, 146, 147], network summarization [12,148,149], GPS track mining [150,151,152,153,154,155, 156,157,158,159], spatial outlier detection [160,161], spatial classification and regression [54, 162, 163] and change footprint detection [164].

2.7 Conclusion

The specific properties of geospatial data; its volume, variety and velocity; and the implicit but complex nature of spatial relationships are non-trivial considerations in all geo-related research. We believe the current practice of independent research in siloed fields is counterproductive and likely untenable in the long term. We are proposing therefore that statistics, mathematics, and computer science all be considered integral to geospatial data science. This chapter explored the emerging field of geospatial data science from such a transdisciplinary perspective where these three closely related scientific disciplines are considered as integral parts of geospatial data science rather than individual siloed disciplines. Our proposed definition aims to reduce the redundant work being done across silos and to understand the limits of geospatial data science.

In the future, we envision that geospatial data science will accomplish its tasks while addressing users' privacy and confidentiality concerns. In addition, there are other issues that will need to be considered such as "the trade-offs across disciplines"; "when to use high-dimensional tools and approaches for geospatial datasets"; "how to apply spatial statistics, which assumes isotropic Euclidean space, on geospatial network datasets (e.g., road networks affect isotropy in space)"; and "how to determine the statistical distribution of geospatial datasets (e.g., GPS trajectories) in a study area?" Finally, predictability and prediction error bounds should be considered since these will provide confidence limits to future approaches of geospatial data science.

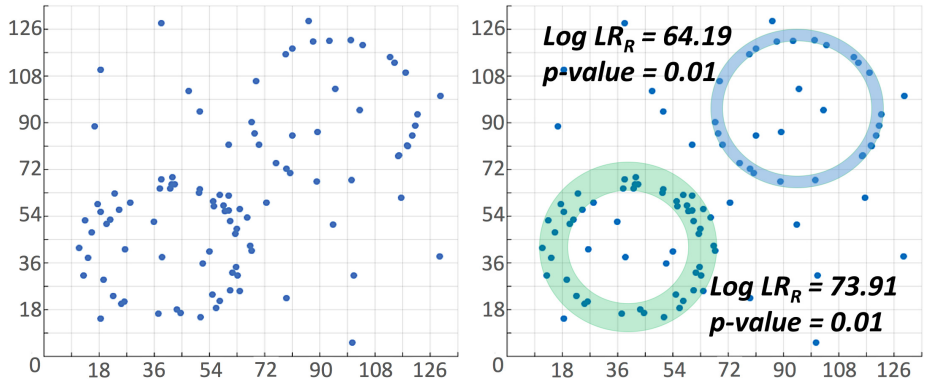
Chapter 3

Ring-Shaped Hotspot Detection

3.1 Introduction

Given a collection of geo-located activities (e.g., crime reports, disease locations), ring-shaped hotspot detection (RHD) finds ring-shaped areas where the concentration of activities inside the ring is significantly higher than outside.

Informally, the RHD problem can be defined as follows: given a set of geo-located activities and thresholds on the size, interest measure (e.g. log likelihood ratio) and statistical significance of a ring (e.g. p-value), find ring-shaped hotspot areas which satisfy the thresholds. A simplified input and output of RHD is shown in Figure 3.1. In this example, a set of 100 activities was generated in a study area of 135×135 unit space (shown as dots). The other inputs are log likelihood ratio threshold $\theta = 50$, statistical significance threshold $\alpha_p = 0.01$, minimum ring width $(r_o - r_i)_{min} = 0.5$, minimum inner radius $(r_i)_{min} = 10$, and the number of Monte Carlo simulation trials is set to $m = 99$. In Figure 3.1(b), the output is two ring-shaped hotspots (e.g. shaded areas) that contain significantly high density of activities relative to outside (e.g. non-shaded areas).



(a) Input (b) Output

Figure 3.1: Simplified example of Ring-Shaped Hotspot Detection.

3.1.1 Application Domains

Ring-shaped hotspot detection is important for a variety of application domains where finding a ring-shaped hotspot may help focus domain users' efforts to a specific region. For example, finding a ring-shaped hotspot may focus public security officials' efforts to the inner circle of a ring when searching for a possible crime source. Following are some example application domains.

Environmental criminology: Analysts create geographic profiles of serial criminals by using the crimes sites to identify the most probable locations of serial criminals and other crime sources. One of the geographic profiling methods [14,15,17] depends on two concepts: (i) distance decay and (ii) buffer zone. Distance decay is a geographical expression of least effort (time, money, etc.) and results when a serial criminal shows preference for closer crime sites [1]. Conversely, serial criminals often do not commit crimes too close to home as this may constitute a threat to their anonymity [1]. This area around a serial criminal's home is referred to as the buffer/comfort zone [1, 14, 16, 17]. Geographic profiling uses the opposing effects of the buffer zone and distance decay to predict the location of a serial criminal [14,16]. Sometimes called a blackhole or doughnut, the pattern is approximated in this paper as a ring-shaped hotspot as illustrated in Figure 3.2.

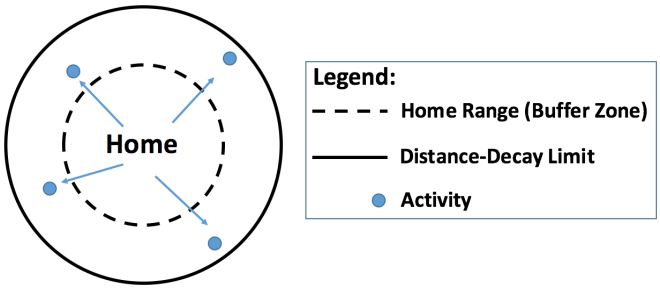


Figure 3.2: Criminals typically do not commit crime close to home (inside the dashed circle) and do not travel too far (beyond the ring) [1].

Epidemiology: If the source of the disease is a tower (or a chimney) which emits the disease causing material (e.g. carcinogenic smoke, disease causing bacteria, etc.), the number of disease cases may be less in closer locations to the source. Conversely, the probability of transmission declines with distance from the source [165, 166]. The interplay between these two may create a ring-shaped hotspot around the possible source of a disease.

Animal Foraging: There are similarities between criminal behavior and predator animal behavior. Predator animals do not travel too far for hunting due to the required effort and time. Predator animals (e.g. bats) also do not hunt too hunt close to their nests [167, 168]. The interplay between these two may give rise to ring-shaped areas of hunting.

Invasive Species Biology: Upon arrival to a new habitat, many invasive species (e.g. bumblebees) establish a nest and a territory around it. Over time, some individuals (e.g. the offspring) leave to establish new nests outside the territory of the original nest [169, 170, 171]. However, they do not go too far due to the least effort principle. Thus, the interplay between these may create a ring-shaped region around the source population of these species.

Geophysics: Meteor impact crater sites often have uplifted centers, broad flat shallow crater floors, and terraced walls creating a shape similar to a ring. However, those are often buried and hidden to the plain sight due to post impact sediments [172, 173]. Such meteor impact crater sites can be detected by interpreting the rings of ground gravity and magnetic anomalies as shown in Figure 3.3 [5]. Although, those

may be detected by plain eye for smaller sites of interest (as shown in Figure 3.3 [5]), computational methods may be required for planetary scale analysis of gravitational data or for regions where numerous meteor impacts overlap.

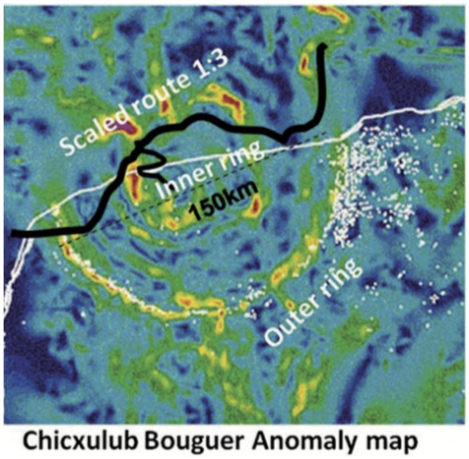


Figure 3.3: Gravity anomaly map of Chicxulub Crater shows ring-shaped anomalous regions of gravity [5].

3.1.2 Challenges

Ring-Shaped Hotspot Detection (RHD) is a challenging problem since the number of hotspots (e.g. reported crime cases may be the result of more than one criminal), their locations and their sizes are not known beforehand, resulting in a prohibitively large candidate enumeration space. In addition, a candidate ring with a smaller area may have more activities than a candidate ring with a larger area (i.e. not monotonic). Thus, pruning techniques that assume monotonicity are inapplicable for RHD. Finally, statistical significance testing to remove chance patterns multiplies the cost.

Finally, geographic profiling and generating hypothesis on the possible source of the activities (e.g. criminal's location) is a challenging task due to the lack of datasets describing the geographic distribution of activities/events. Therefore, this task is often carried out by generating and ranking hypothesis using the data in hand.

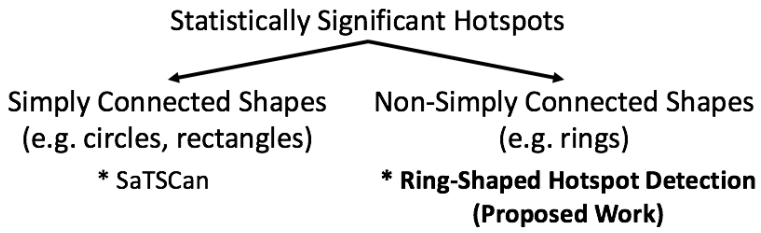
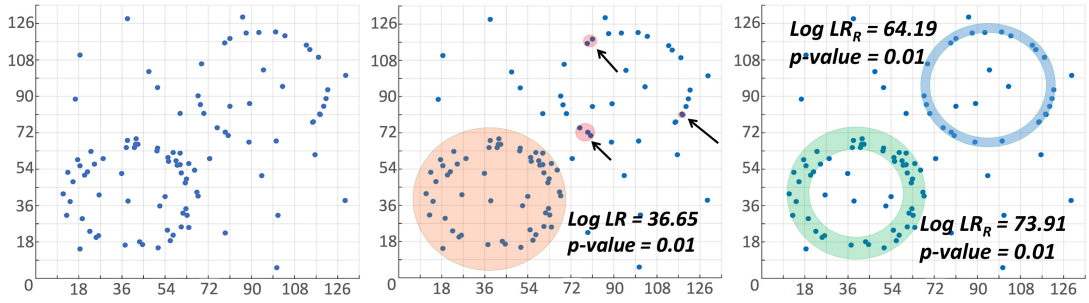


Figure 3.4: Related work classification.

3.1.3 Related Work and Their Limitations

Methods for hotspot detection can be roughly divided into two groups: (i) those that do not test for statistical significance and (ii) those do test for statistical significance. Hotspot detection methods which do not take statistical significance into account are widely used in many domains [28, 164]. However, they may produce false positives. While not a concern in all domains, those false positives may lead to social stigma and major economic loss in some application domains (e.g. criminology, epidemiology). In this paper, the aim is to remove the chance patterns and focus on techniques which take statistical significance into account.

Figure 3.4 classifies statistically significant hotspot detection techniques in two groups namely simply connected shapes and non-simply connected (e.g. rings) shapes. The former used spatial scan statistics [10,174,175] to detect simply connected shapes [176] such as circles and rectangles without any holes. In contrast, this paper explores techniques to identify statistically significant hotspots with non-simply connected shapes, i.e. rings with holes.



(a) Input (b) SaTScan Output (c) RHD Output

Figure 3.5: Example Input, Output of SaTScan, and Output of Ring-Shaped Hotspot Detection (best in color).

Figure 3.5 shows an example input and output of RHD comparing it with SaTScan [10], which is a widely used application for circular hotspot detection in epidemiology. Figure 3.5(a) visualizes our example activity set that is previously shown in Figure 3.1. Figure 3.5(b) shows the output of SaTScan with four circular hotspots which have low log likelihood ratios. Since SaTScan is not designed to detect hotspots with holes, the output does not align with the ring-shaped hotspots in the study area. In contrast, RHD outputs two ring-shaped hotspot areas with high log likelihood ratios and the shapes of these hotspots align with the ground truth.

Our preliminary work [141] proposed a Dual Grid Based Pruning Algorithm (DGP) for the RHD problem. DGP was experimentally validated and a case study comparing its output with SaTScan found that it yielded better results. This paper extends our previous work by proposing algorithmic refinements that improve the scalability of our preliminary DGP approach [141].

3.1.4 Contributions

Our new contributions are as follows:

- We present prune phase algorithmic refinements featuring different grid cell length for the center and width of rings along with a multi cell length approach (DGP-M) and local maxima elimination (DGP-L) (Section 3.4.1).

- We also propose a new refine phase based on a best enclosing ring heuristic (DGP-R)(Section 3.4.2).
- We analytically prove the correctness of the proposed algorithmic refinements (Section 3.5.1) and we provide a cost analysis (Section 3.5.2).
- We present a new case study on real dataset that shows that the proposed **DG-PLMR** (**D**ual **G**rid Based **P**rune with **L**ocal maxima elimination, **M**ulti cell length and **R**efine with best enclosing ring) algorithm outperforms the related work (e.g., SaTScan)(Section 3.6).
- We present experimental results on synthetic data that validate the performance of the algorithmic refinements. The results show that the proposed DGPLMR algorithm is almost two orders of magnitude faster than the DGP algorithm in preliminary work [141] (Section 3.7).

3.1.5 Scope and Outline

This work presents a computational approach for statistically significant ring-shaped hotspot detection. Rings should be considered as an approximation of actual doughnut patterns (generated by the interplay between distance decay and buffer zone) which consist of nested closed curves. Therefore, non-concentric rings and nested closed curves are outside the scope of our work. In this paper, underlying population and other variables associated with an activity set are not considered. Also, input parameters are governed by domain experts and this paper does not provide guidance on parameter (e.g. minimum radius, etc.) value selection. The study area is assumed to be a two dimensional isotropic Euclidean space and geographic barriers and transportation network are not considered. In addition, techniques to detect linear [12], network or top-k hotspots are not comparable to the proposed approach.

It is also worth noting that proposed approach uses simplifying assumptions (e.g. isotropic Euclidean space) and approximations (e.g. approximating doughnut patterns by rings) to scale up to large datasets (e.g. one million activity points). The results should be reviewed by analysts in the context of additional domain information to prevent potentially misleading results (e.g. biased and/or missed rings due to approximations). However, the post-processing steps (e.g. review of the output and removal of

potentially misleading results) by analysts are out of the scope of this paper.

The paper is organized as follows: Section 3.2 presents basic concepts and problem statement of Ring-Shaped Hotspot Detection (RHD). Section 3.3 reviews our preliminary approach towards addressing the RHD problem [141]. Section 3.4 details the proposed DGPLMR algorithm. Section 3.5 presents a theoretical and cost analysis. Section 3.6 presents a case study comparing DGPLMR to a diffusion theory based output (e.g., SaTScan) on a real disease dataset. The experimental evaluation is covered in Section 3.7. Section 3.8 presents a discussion on techniques (e.g. DBSCAN [9]) without statistical significance as well as a discussion on the post-processing of the output. Section 3.9 concludes the paper and previews future work.

3.2 Basic Concepts and Problem Statement

3.2.1 Basic Concepts

Definition 1 ***An activity set*** *A is a collection of geo-located activities. An* ***activity*** *$a \in A$ is associated with a pair of coordinates (x, y) representing its spatial location in the study area S.*

Definition 2 ***Study area*** *S is the minimum bounding rectangle of A in the Euclidean space.*

Definition 3 *From mathematical perspective, a concentric* ***Ring*** *is defined by four parameters: the coordinates of its center ($cent_R = [x, y]$), an inner circle radius r_i, and an outer circle radius r_o.*

Mathematical definition of rings may introduce an infinite number of rings in a study area which is computationally infeasible to enumerate. Therefore, for implementation reasons, we introduce an implementation-specific (computational) 4P-CRing definition next.

Definition 4 *A 4P-CRing (denoted by R in this paper) is a shape formed by the set difference of two concentric circles, i.e. $R = circle_o - circle_i$. A $circle_i$ can be uniquely identified by any non-collinear three points (a_1, a_2, $a_3 \in A$). A $circle_o$ can be identified*

for any activity point $a_4 \in A$ outside $circle_i$, where a_4 is on $circle_o$ and $circle_o$ is concentric with $circle_i$. An illustration of a 4P-CRing is shown in Figure 3.6.

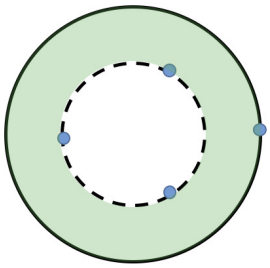


Figure 3.6: An illustrative example of a 4P-CRing. Dots represent points, dashed circle represent inner circle $circle_i$ and circle with a solid line represents outer circle $circle_o$.

Spatial Scan Statistics

In hotspot detection, there are several questions that need to be addressed: (1) What is the location of a hotspot? (2) Is there any difference between the distribution of points inside the candidate hotspot vs. outside? (3) How can we make sure that the candidate hotspot is not a chance pattern? Spatial scan statistics was developed to answer such questions. It was proved to be the most powerful statistical test to detect hotspots in an activity set [177].

The first question is addressed by enumerating the candidate hotspot locations. These candidate hotspots may be rings, circles, rectangles, or jurisdictions (i.e. zip codes). In this paper, our candidates are 4P-CRings as defined in Definition 4.

The second question is addressed by evaluating the candidate rings with a test statistic function. In this paper, we use a "Log Likelihood Ratio" function similar to related work [10].

The third question is answered by creating Hypothesis test and evaluating the candidate rings against a statistical significance threshold using a "Randomization Test". These tests are defined as follows (more details can be found in [178]):

Definition 5 ***Test Statistic: Log Likelihood Ratio*** *(Log LR_R) is the interest measure used as the test statistic for a candidate ring [177, 178]. The equation can be shown as:*

$$Log\ LR_R = Log\left(\underbrace{\left(\frac{c}{B}\right)^{c}}_{Int.\ of\ the\ Ring} \times \underbrace{\left(\frac{|A|-c}{|A|-B}\right)^{|A|-c}}_{Ext.\ of\ the\ Ring} \times I()\right) \tag{3.1}$$

$B = \frac{|A| \times area(R)}{area(S)}$. $I() = \begin{cases} 1, & if\ c > B \\ 0, & otherwise \end{cases}$

where B denotes the expected number of activities, c denotes the observed number of activities in a candidate ring, $|A|$ denotes the number of activities of set A. The term $I()$ is an indicator function. $I() = 1$ when the ring has more activity points than expected ($c > B$), otherwise it is set to 0 to prevent the detection of low activity ring areas [10].

For example, the area of the bottom left ring in Figure 3.5(c) is 1105 and $area(S) = 135 \times 135 = 18225$. Thus $B = \frac{1105 \times 100}{18225} = 6.06$. In this ring, there are $c = 50$ activities and $I = 1$ since $50 > 6.06$.

Using Equation 3.1, $Log\ LR_R = Log\left(\left(\frac{50}{6.06}\right)^{50} \times \left(\frac{100-50}{100-6.06}\right)^{100-50} \times 1\right) = 73.91$

Definition 6 ***Monte Carlo Simulation (MCS)*** *is a randomization test that is used to get the distribution of the test statistic (Log LR_R). First, m random point sets $(A_1, A_2, ..., A_m)$ are generated in the study area S. For each $A_1, A_2, ..., A_m$, new rings are enumerated and the maximum Log LR_R of each $A_1, A_2, ..., A_m$ is stored in decreasing order in a list (Log LR_R^{MCS}) which is used as the distribution of the test statistic under the null hypothesis.*

Definition 7 ***Hypothesis Test:*** *In RHD, the null hypothesis (H_0) states that the points are distributed randomly according to a homogeneous Poisson process over the study area S. The alternative hypothesis (H_1) states that the inside of a ring has a higher number of points than outside [178]. Using the test statistic (Log LR_R) of a ring and the distribution of the test statistic (acquired by MCS), the statistical significance of that ring is determined. The statistical significance (p-value) of R is computed by finding the order of its Log LR_R in the distribution of the test statistic (Log LR_R^{MCS}) and dividing that position by $m+1$. Given a desired significance level (α_p), if p-value $\leq \alpha_p$, then H_1 cannot be rejected.*

3.2.2 Problem Statement

The Ring-Shaped Hotspot Detection problem is formally stated as follows:
Given:

1. A set of activities A where each activity $a \in A$ has x, y coordinates in a Euclidean space,
2. A minimum ring width ($(r_o - r_i)_{min}$) and a minimum ring inner radius ($(r_i)_{min}$)
3. A log likelihood ratio threshold (θ),
4. A spatial scan statistic *p-value* threshold (α_p) and
5. A number of Monte Carlo simulation trials (m)

Find:

Ring-shaped hotspots (enumerated as 4P-CRings) in S with $(r_o - r_i)_R \geq (r_o - r_i)_{min}$, $(r_i)_R \geq (r_i)_{min}$, $p\text{-}value_R \leq \alpha_p$ and $Log\ LR_R \geq \theta$.
Objective: Computational efficiency.
Constraints:

1. Correctness of the result set.
2. Inner circles of the rings should have at least three points and outer circles should have at least one point on their circumference.
3. Data which describes the underlying distribution of activities is not available.
4. Cardinality of A is large enough ($|A| >> 4$) to provide statistical significance.

Activity set A and activities $a \in A$ are defined in Definition 1. θ indicates the minimum desired log likelihood ratio and α_p is the desired level of statistical significance and m is the number of Monte Carlo simulations trials to determine the statistical significance. Finally, the outputs are ring-shaped hotspots with the desired level of significance and log likelihood ratio.

It is worth mentioning that the parameters should be chosen by domain users. Using $(r_i)_{min}$ and $(r_o - r_i)_{min}$ domain users can select the radii of generated rings depending

on the nature of the activity. For example, in environmental criminology, crime analysis shows that 82% of arsonists live within 0.5 to 2 miles of the crime sites [179]. This domain knowledge may be used to determine $(r_i)_{min}$ and $(r_o - r_i)_{min}$. The above problem statement includes parameters that limit the geographic size of a ring (e.g. $(r_i)_{min}$ and $(r_o - r_i)_{min}$). However, a user may also need to limit the number of points inside a ring for domain specific purposes. This can be supported by a post-processing step that filters out the rings with higher/lower than the number of points required.

Example: Given the activity set depicted in Figure 3.5(a), the aim is to find ring-shaped hotspots with a level of significance $\alpha_p = 0.01$, a log likelihood ratio $\theta = 50$, and a minimum inner radius $(r_i)_{min} = 10$. Figure 3.5(c) shows two rings returned by RHD which satisfy the desired thresholds. Note that the result set includes many ring-shaped hotspots, but in order to reduce the visual clutter, only one generated ring is shown for each hotspot.

3.3 Preliminary Results - Dual Grid Based Pruning Algorithm (DGP)

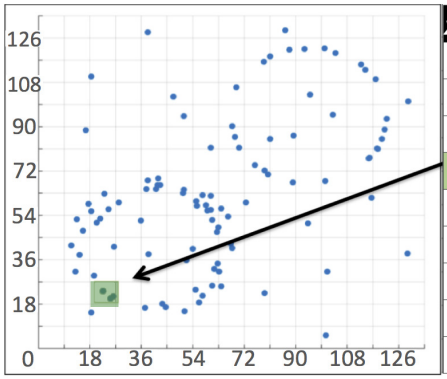
This section reviews our previous dual grid based pruning (DGP) approach to Ring-Shaped Hotspot Detection (RHD) problem [141]. The intuition behind the DGP algorithm's pruning phase is to enumerate ring families (instead of individual rings) in parametric space and eliminate those that do not satisfy the input thresholds (i.e. $(r_o - r_i)_{min}$, $(r_i)_{min}$, θ). Finally, activity points that are associated with a parametric grid cell, which satisfy the thresholds, are used to enumerate actual 4P-CRings. Once actual 4P-CRings are enumerated using those activity points, if they satisfy the input thresholds, they are evaluated for their statistical significance. In the next subsections, the basic concepts of the dual grid based pruning (DGP) algorithm are presented; then an upper bound for the log likelihood ratio is shown and its correctness is proven. Finally, the details and the execution trace of the DGP algorithm is presented.

3.3.1 Basic Concepts of DGP Algorithm

Definition 8 ***A geometric space grid (geo-grid)*** *with cell length lg is a partitioning of the study area S into two-dimensional geometric grids where each geo-grid cell (gc) is*

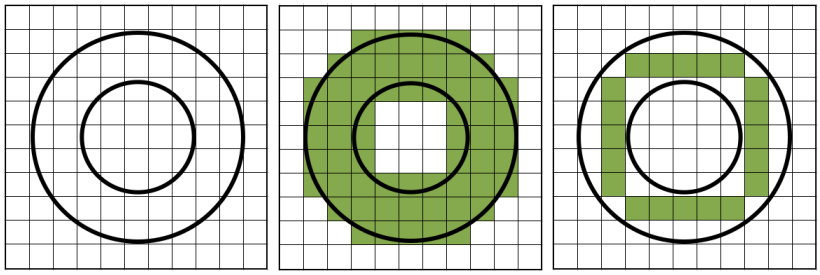
represented by its ***coordinate intervals*** $([x_{min}, x_{max}], [y_{min}, y_{max})$ *and the* ***count*** *of the points inside. The number of gc in S is* $N \times N$*, where* $N = floor(side\_length(S)/lg)$*. Note that each activity point is associated with exactly one geo-grid cell gc.*

Figure 3.7 shows our example activity set with 100 points that includes 2 ring-shaped hotspots (with 50 and 25 activity points respectively). Suppose cell length is set to $lg = 9$. The geo-grid cells will keep the count of activity points. For example, the shaded geo-grid cell in Figure 3.7 has an interval of $([x_{min} = 18, x_{max} = 27], [y_{min} = 18, y_{max} = 27])$ and a count of 3 as shown on the fourth row of the table.



x_{min}	x_{max}	y_{min}	y_{max}	count
0	9	0	9	0
0	9	9	18	0
...	...	...	...	...
18	27	18	27	3
18	27	27	36	1
...	...	...	...	...
72	81	63	72	1
72	81	72	81	2
...	...	...	...	...

Figure 3.7: A sample geo-grid for the input activity set with 100 activity points. In this example, cell length is set to $lg = 9$.



(a) A ring R (b) Min. bounding geo-sub-grid(R) (c) Max. bounded geo-sub-grid(R)

Figure 3.8: Shaded cells show a sample minimal bounding geo-sub-grid(R) and a sample maximal bounded geo-sub-grid(R) used for the computation of the upper bound log likelihood ratio ($Log\ \widehat{LR_p}$).

Definition 9 ***Minimal bounding geo-sub-grid:*** *Given a ring and a geo-grid, a minimal bounding geo-sub-grid(R) is defined as the collection of geo-grid cells gc that overlap with R (e.g., shaded cells in Figure 3.8(b) represent a minimal bounding geo-sub-grid for the ring in Figure 3.8(a)).*

Definition 10 ***Maximal bounded geo-sub-grid:*** *Given a ring and a geo-grid, a maximal bounded geo-sub-grid(R) is defined as the collection of geo-grid cells gc that is completely inside the ring (e.g., shaded cells in 3.8(c) represent a maximal bounded geo-sub-grid for the ring in Figure 3.8(a)).*

Suppose there is a ring in the study area. A *minimal bounding geo-sub-grid*(R) and a *maximal bounded geo-sub-grid*(R) are valid for that ring if (a) at least one geo-grid cell is completely inside the inner circle of a ring, and at least one geo-grid cell is completely inside the area in between inner circle and outer circle of a ring and if (b) the *minimal bounding geo-sub-grid*(R) has less than $N \times N$ geo-grid cells. Note that (a) can be satisfied by selecting the cell length as the smaller of $lg \leq (r_i)_{min}/\sqrt{2}$ and $lg \leq (r_o - r_i)_{min}/\sqrt{2}$, where $(r_i)_{min}$ is the minimum inner circle radius of a ring and $(r_o - r_i)_{min}$ is the minimum thickness of a ring, as defined in the problem statement in Section 3.2.

Definition 11 ***A parametric space grid (param-grid)*** *is defined by four parameters, namely two-dimensional center coordinates $([x_{min}, x_{max}], [y_{min}, y_{max}])$ (where $x_{max} = x_{min} + lg$ and $y_{max} = y_{min} + lg$), an inner radius $[r_i^{min}, r_i^{max}]$, and an outer radius $[r_o^{min}, r_o^{max}]$. Given a cell length lg, the number of geo-grid cells is $N \times N$, and the number of param-grid cells is $N \times N \times N \times N$. In other words, a param-grid discretizes the four dimensional parameter space into param-grid cells (pc) with $\left([x_{min}, x_{max}], [y_{min}, y_{max}], [r_i^{min}, r_i^{max}], [r_o^{min}, r_o^{max}]\right)$. A pc represents a collection of rings if the conditions for minimal bounding geo-sub-grid(R) and maximal bounded geo-sub-grid(R) are satisfied and $\{(x_{min} \leq x \leq x_{max}), (y_{min} \leq y \leq y_{max}), (r_i^{min} \leq r_i \leq r_i^{max}), (r_o^{min} \leq r_o \leq r_o^{max})\}$.*

Table 3.1: Geo-grid cells for $lg = 1$ generated for activity set in Figure 3.5(a).

$[x_{min}, x_{max}]$	$[y_{min}, y_{max}]$	Count
1,2	1,2	0
17, 18	27, 28	0
18, 19	14, 15	1
79, 80	72, 73	1
. . .	. . .	. . .

Table 3.2: Sample param-grid cells for $lg = 1$ generated for activity set in Figure 3.5(a).

x_{min}, x_{max}	y_{min}, y_{max}	r_i^{min}, r_i^{max}	r_o^{min}, r_o^{max}	$Log\ \widehat{LR_p}$
40, 41	40, 41	22, 23	29, 30	92.69
38, 39	41, 42	23, 24	26, 27	92.34
94, 95	94, 95	26, 27	28, 29	65.62
40, 41	35, 36	18, 19	31, 32	58.18

Table 3.2 illustrates several *pc*s for the activity set in Figure 3.5(a) when grid cell length is set to $lg = 1$. For example, the top row in the table shows a *pc* with $([40, 41], [40, 41], [22, 23], [29, 30])$.

Definition 12 *Given a param-grid cell pc, the* ***upper bound log likelihood ratio*** *(Log* $\widehat{LR_p}$*) can be computed as follows:*

$$
\begin{aligned}
&Log\ \widehat{LR_p} = Log\left(\widehat{LR_{int}} \times \widehat{LR_{ext}} \times \widehat{I()}\right),\ \textit{where}\\
&\widehat{LR_{int}} = \left(\frac{U(c)}{L(B)}\right)^{U(c)},\ \textit{and}\\
&\widehat{LR_{ext}} = \begin{cases} \left(\frac{|A|-L(c)}{|A|-U(B)}\right)^{(|A|-U(c))}, & \textit{if } L(c) \geq U(B) \\ 1, \textit{otherwise} \end{cases}\\
&\widehat{I()} = \begin{cases} 1, & \textit{if } U(c) > L(B) \\ 0, \textit{otherwise} \end{cases}
\end{aligned}
\tag{3.2}
$$

It can be stated that $U(c)$ is an upper bound of c, $L(c)$ is a lower bound of c, $U(B)$ is an upper bound of B, and $L(B)$ is a lower bound of B.

$U(c) = \#$ *of points in minimal bounding geo-sub-grid*(p)
$L(c) = \#$ *of points in maximal bounded geo-sub-grid*(p)
$U(B) = \frac{area(minimal\ \ bounding\ \ geo\text{-}sub\text{-}grid(p))\times|A|}{area(S)} < |A|$
$L(B) = \frac{area(maximal\ \ bounded\ \ geo\text{-}sub\text{-}grid(p))\times|A|}{area(S)} > 0$

The upper bound log likelihood ratio is applied if and only if the maximal bounded geo-sub-grid(R) *has at least one gc and the minimal bounding geo-sub-grid*(R) *has less than* $N \times N$ *geo-grid cells (gc).*

$\widehat{LR_{int}}$ and $\widehat{LR_{ext}}$ are the upper bounds of the two components in Equation 3.2, representing the interior and the exterior of a ring respectively. The second component of $\widehat{LR_{ext}}$ has two cases, conditioned on whether $L(c) \geq U(B)$. It is proved in Theorem 1 that this upper bound always gives a value higher than the actual *Log* LR_R. The indicator function $\widehat{I()}$ is similar to the indicator function in Equation 3.2 and is set to $I() = 1$ when the upper bound of activity points is higher than the lower bound of the expected activity points ($U(c) > L(B)$).

For example, on the top row of Table 3.2, upper bound log likelihood ratio *Log* $\widehat{LR_p}$ is computed by the following: $U(B) = 7.11$, $L(B) = 5.31$, $L(c) = 37$ and $U(c) = 50$. Finally, *Log* $\widehat{LR_p} = Log\left(\left(\frac{50}{5.31}\right)^{50} \times \left(\frac{100-37}{100-7.11}\right)^{100-50} \times 1\right) = 92.69$.

Lemma 1 *For any ring* $R(x, y, r_i, r_o)$*, Equation 3.2 is well defined if the cell length is selected as* $lg \leq (r_o - r_i)_{min}/\sqrt{2}$.

Proof 1 *In Equation 3.2,* $L(B)$ *and* $L(c)$ *are defined by the maximal bounded geo-sub-grid cells, and* $U(B)$ *and* $U(c)$ *are defined by the minimal bounding geo-sub-grid cells. If* $L(B) = 0$*, then Log* $\widehat{LR_p}$ *can't be computed since* $\widehat{LR_{int}}$ *in Equation 3.2 will be undefined. Thus, in order for* $L(B) > 0$*, there needs to be at least one gc in the collection of maximal bounded geo-sub-grid cells. Therefore, lg should be set to* $lg \leq (r_o - r_i)_{min}/\sqrt{2}$.

Lemma 2 *If a ring has* $c > B$*, then* $\widehat{LR_{int}} \geq \left(\frac{c}{B}\right)^c$.

Proof 2 *If* $c > B$*, then* $\frac{c}{B} > 1$*. Since* $B \geq L(B)$*,* $\left(\frac{c}{B}\right)^c \leq \left(\frac{c}{L(B)}\right)^c$*. Also,* $U(c) \geq c$*,* $\left(\frac{c}{L(B)}\right)^c \leq \left(\frac{U(c)}{L(B)}\right)^c \leq \left(\frac{U(c)}{L(B)}\right)^{U(c)}$ *and* $U(c) > L(B)$*, thus* $\widehat{LR_{int}} \geq \left(\frac{c}{B}\right)^c$.

Lemma 3 *If $c > B$, then $\widehat{LR_{ext}} \geq \left(\frac{|A|-c}{|A|-B}\right)^{(|A|-c)}$.*

Proof 3 *In Likelihood Ratio equation [178], if $c > B$ then $(|A|-c) \leq (|A|-B)$, implying $\frac{|A|-c}{|A|-B} > 1$. Now consider two cases:*
Case 1: *$L(c) \geq U(B)$*
$\left(\frac{|A|-c}{|A|-B}\right)^{(|A|-c)} \leq \left(\frac{|A|-L(c)}{|A|-B}\right)^{(|A|-c)}$ *(Since $c \geq L(c)$)*
$\leq \left(\frac{|A|-L(c)}{|A|-L(B)}\right)^{(|A|-c)}$ *(Since $B \geq L(B)$)*
$\leq \left(\frac{|A|-L(c)}{|A|-L(B)}\right)^{(|A|-U(c))} = \widehat{LR_{ext}}$ *(since quotient is less than 1, and it is raised to a smaller exponent)*
Case 2: *$L(c) < U(B)$*
Since $\left(\frac{|A|-c}{|A|-B}\right) < 1$, then $\left(\frac{|A|-c}{|A|-B}\right)^{(|A|-c)} < 1 = \widehat{LR_{ext}}$.
To sum up, if $c > B$, then $\widehat{LR_{ext}} \geq \left(\frac{|A|-c}{|A|-B}\right)^{(|A|-c)}$.

Lemma 4 *If $c > B$, then $\widehat{I()} = I()$.*

Proof 4 *If $c > B$, then $I() = 1$. Since $U(c) \geq c$ and $L(B) \leq B$ then $U(c) \geq L(B)$. Using Definition 12, $\widehat{I()} = 1$ and it can be concluded that $\widehat{I()} = I()$.*

Theorem 1 *For any ring R where $c > B$, Log $\widehat{LR_p} \geq$ Log LR_R.*

Proof. According to Lemma 2, Lemma 3 and Lemma 4, if $c > B$, then $\widehat{LR_{int}} \geq \left(\frac{c}{B}\right)^c$, $\widehat{LR_{ext}} \geq \left(\frac{|A|-c}{|A|-B}\right)^{(|A|-c)}$, and $\widehat{I()} = I()$. Therefore, *Log* $\widehat{LR_p} \geq$ *Log* LR_R.

3.3.2 Dual Grid based Pruning (DGP) Algorithm

Algorithm 1 shows the three phases of the DGP algorithm. The **prune phase** uses a dual-grid which returns param-grid cells (*pc*s) and the associated activity points in *prunedSet*s. This information is then sent to the **refine phase** in order to find the candidate rings. Finally, candidate rings are tested for statistical significance in the **Monte Carlo simulation phase**.

A. Prune Phase: First, the study area is discretized into geo-grids and param-grids (lines 1-2) and geo-grid cell count values are computed (line 3). Since all *gc*s are equal-sized, baseline B_{gc} for an individual *gc* is then determined (line 4). For every

param-grid cell (pc), the $Log\ \widehat{LR_p}$ is computed (lines 5-8). In Line 6, it can be seen that r_o^{cell} can be as high as $N/2$. This restriction is to limit the rings' radii inside S. Also, number "3" in Line 6 is selected to prevent cell collection inside maximal bounded geo-sub-grid from exceeding the cell collection inside minimum bounding geo-sub-grid which may cause spurious upper bound likelihood ratios. Finally, if the upper bound log likelihood ratio $Log\ \widehat{LR_p} \geq \theta$, then those pcs and $prunedSet$s are sent to the refine phase (lines 9-10).

Algorithm 1 Dual Grid Based Pruning Algorithm

Inputs are the same as in Problem Statement

A. Prune Phase:

Discretize S into $[N \times N]$ geo-grids using lg

Create param-grids with $[N \times N \times N \times N]$

Associate counts c to geo-grid cells(gc)

Baseline (Expected) $B_{gc} \leftarrow |A|/(N \times N)$

for each param-grid cell pc **do**

  for each $r_i^{cell} \leftarrow$ 1 *to* $N/2 - 3$ **do**

    for each $r_o^{cell} \leftarrow (r_i + 3)$ *to* $N/2$ **do**

      $pc \leftarrow Log\ \widehat{LR_p}$

      if $Log\ \widehat{LR_p} \geq \theta$ **then** $prunedSet \leftarrow$ *all* $a \in pc$

Send all pcs and $prunedSet$s to refine phase

B. Refine Phase

$circle_i \leftarrow$ activity triplets $\in A$

if $r_i \geq (r_i)_{min}$ **then**

  $circle_o \leftarrow$ each a outside $circle_i$ thus create R

if $(r_o - r_i) \geq (r_o - r_i)_{min}$ **then**

  Compute $Log\ LR_R$

  if $Log\ LR_R \geq \theta$ **then** Add R to $candidateRings$

C. Monte Carlo Simulation Phase:

$significantRings \leftarrow$ rings R with $p\text{-}value \leq \alpha_p$

Return $significantRings$

B. Refine Phase: The DGP Refine Phase begins by enumerating every three non-linear points to generate the inner circle ($circle_i$) of a 4P-CRing (line 11). All points outside $circle_i$ are used to generate the outer circle $circle_o$, and thus R (lines 12-13). If R is thicker than the specified ring width ($(r_o - r_i)_R \geq (r_o - r_i)_{min}$), and r_i is greater than the required minimum inner circle radius ($(r_i)_R \geq (r_i)_{min}$), *Log* LR_R value for a R is computed using its area and the number of activity points it contains (line 14). If the *Log* $LR_R \geq \theta$, then R is added to *candidateRings* (line 15).

Lemma 5 *Given a set of activity points, DGP Refine phase is complete. An algorithm is considered complete if it discovers any 4P-CRing that qualifies the input thresholds.*

Proof 5 *The proof is trivial. In Line 11 of Algorithm 1, all possible triple combinations of points are traversed to create* $circle_i$ *of 4P-CRings. Similarly, in Line 13, for each* $circle_i$*, if* $r_i \leq (r_i)_{min}$*, all points outside* $circle_i$ *is used to enumerate* $circle_o$*s. Thus, according to the Definition 4, since all possible 4P-CRings, which exceed the thresholds, are enumerated, DGP Refine phase is complete.*

Corollary 1 *Using Lemma 5, DGP Algorithm is complete if prune phase is not used and the entire input (i.e. activity set and thresholds) is given to the refine phase.*

C. Monte Carlo Simulation Phase: A *p-value* is determined for the *candidateRings* using m trials of Monte Carlo simulation. First, m random activity sets are generated. For each random activity set, DGP runs and stores the highest *Log* LR_R of every trial (if no pc survives the prune phase for an A_{rand}, its highest *Log* LR_R is set to 1). These m log likelihood ratio values are then ordered and compared with the *Log* LR_R of *candidateRings*. The order of *Log* LR_R of *candidateRings* is divided by $m + 1$ to get the *p-value* of a 4P-CRing.

DGP Execution Trace: Figure 3.5(a) shows our input activity set with $|A| = 100$. The *Log* LR_R threshold θ is set to 50 and cell length is set to $lg = 1$ (for illustration purposes). First, the prune phase creates geo-grids with 135×135 geo-grid cells (gc) and param-grids with $135 \times 135 \times 68 \times 68$ param-grid cells (pc). Next, for each gc, the number of activity points inside is determined. For example, the geo-grid cell $([18, 19], [14, 15])$ has 1 activity point (Table 3.1). Then, B is determined by dividing the cardinality of A by the total number of gcs. In this example $B = 100/(135 * 135) = 0.0055$. Next,

$Log\ \widehat{LR_p}$ for each pc is computed. Table 3.2 shows sample pcs. For example, the param-grid cell $([40,41],[40,41],[22,23],[29,30])$ has $Log\ \widehat{LR_p} = 92.69$. This is done for each pc $(135 \times 135 \times 68 \times 68)$. Finally, all pcs with $Log\ \widehat{LR_p} \geq \theta$ and their associated activity points are saved in $prunedSets \subset A$ and sent to the refine phase. In Table 3.2, the top row which is defined by a pc and the associated points are sent to the refine phase as $prunedSet$.

In the refine phase, 4P-CRings are enumerated with the activity points in each $prunedSet$. Next, actual $Log\ LR_R$ are computed and 4P-CRings with $Log\ LR_R \geq \theta$ are saved in $candidateRings$. Table 3.3 shows sample $candidateRings$ with $Log\ LR_R \geq \theta$ returned by refine phase.

Finally, $candidateRings$ are tested for statistical significance and the $p\text{-}value$ of the $candidateRings$ determined. Only Rs with $p\text{-}value \leq \alpha_p$ are saved as $significantRings$. Figure 3.9 shows two output 4P-CRings with $Log\ LR_R = 73.91$ and 64.19 respectively and both of which have $p\text{-}value = 0.01$.

Table 3.3: Sample Output of the DGP for the activity set in Figure 3.5(a).

x	y	r_i	r_o	$area(R)$	B	c	$LogLR_R$
40.72	40.84	22.33	29.17	1106	6.06	50	73.91
40.72	40.84	22.33	29.17	1106	6.07	48	68.50
94.55	94.51	27.05	27.93	151.17	0.83	25	64.19
94.64	94.48	27.09	27.95	150.08	0.82	23	57.09
94.53	94.49	27.06	27.93	150	0.82	21	50.05

3.4 Proposed Approach

We found in our preliminary approach that in the prune phase of DGP algorithm, selecting a coarse cell size may cause rings to be missed, while fine cell size incurs high cost. Thus, we propose refinements for the prune phase of the DGP. Similarly, the refine phase of the DGP algorithm uses an extensive candidate enumeration strategy which causes many almost identical candidates to be enumerated and evaluated for each prunedSet (e.g. $|prunedSet|^4$ 4P-CRings). Therefore, we also propose a new refine phase

algorithm which addresses scalability issues by preventing enumeration of all possible 4P-CRings from a *prunedSet*.

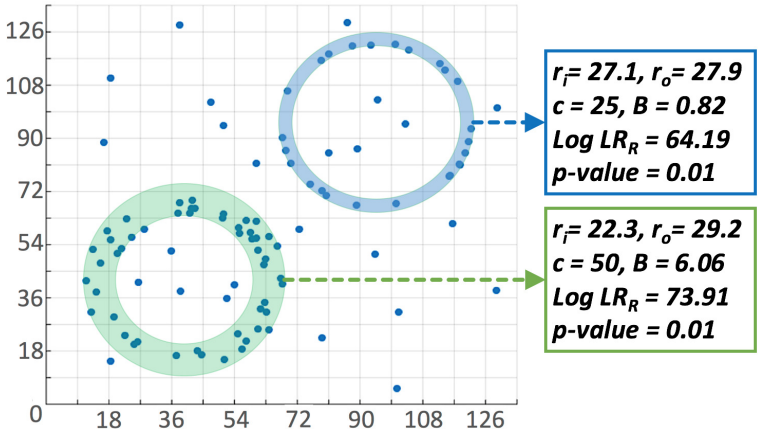


Figure 3.9: Illustration of the output of DGP.

The proposed DGPLMR (**D**ual **G**rid Based **P**rune with **L**ocal maxima elimination, **M**ulti cell length and **R**efine with best enclosing ring) algorithm incorporates the following new algorithmic refinements: Different cell sizes for the center and width of a ring, multi cell size for the width of a ring, local maxima elimination for pruned cells and a new refine phase using a best enclosing ring algorithm.

3.4.1 Algorithmic Refinements for the Prune Phase

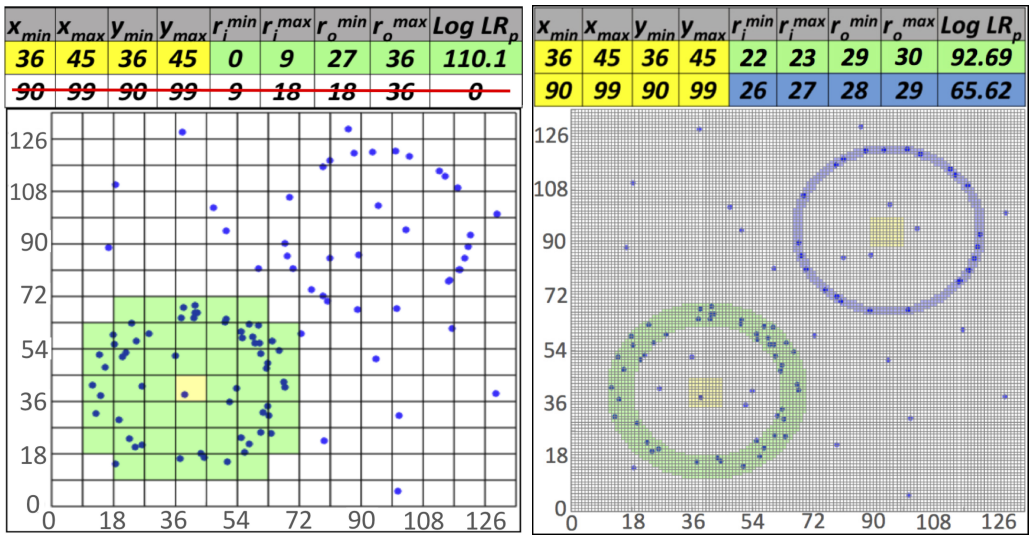
Different cell sizes for the center and width of the ring

In DGP, the cell length lg is constant. If lg is set coarser, due to the generation of rings on the parametric grid space, rings may be missed if their actual widths are thin. In Figure 3.10(a), $lg = 9$ and the upper bound log likelihood ratio computed for the second row is $0 < \theta$. Thus the top right ring will be pruned by DGP. On the other hand, if the lg is set finer, the generated number of cells increases. Since pruning returns all parametric grid cells (pc) with $Log\ \widehat{LR} \geq \theta$, even for the same ring, many pcs and *prunedSet*s are sent to the refine phase. Therefore, selecting a too small cell length causes redundant computation. For the same example, if the lg was selected as 1, both of the rings would be detected by the algorithm by trading the speed due to the high

number of cells to be created. This makes it hard to find a balance between a fine and a coarse cell length and the user is forced to decide between speed and ring detection.

Our new approach uses different cell lengths for the center (lg_c) and the width of the ring (lg_w). When creating rings using parametric space, lg_c for the center of the ring is kept coarse but lg_w for the ring width is selected fine. In Figure 3.10(b), cell lengths are selected as $lg_c = 9$ and $lg_w = 1$.

Note that in order for a ring to be represented by a *minimal bounding geo-sub-grid*(R) there should be at least one geo-grid cell completely inside the inner circle of a ring (Definition 9). Thus, cell length for the center (lg_c) of the ring should be $lg_c \leq (r_i)min/\sqrt{2}$. Similarly, in order for the *maximal bounded geo-sub-grid*(R) to have cells completely inside the ring (Definition 9), cell length for the center (lg_w) of the ring should be $lg_w \leq (r_o - r_i)min/\sqrt{2}$.



(a) Coarse cell length ($lg = 9$) (b) Multi cell lengths ($lg_c = 9$ and $lg_w = 1$)

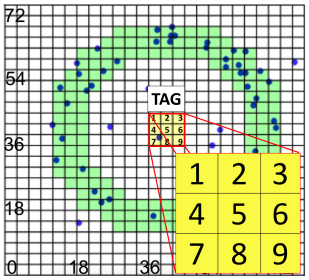
Figure 3.10: Illustration of missing rings due to coarse cell length (3.10(a)) and using multi cell lengths for the center (in the middle of shaded cells - shown in yellow) and width (shaded cells - shown in green/blue) of the rings (3.10(b)) to detect those.

Local Maxima Elimination

In DGP, if the cell length (lg) is smaller than the r_i of the R, multiple param-grid cells and multiple *prunedSet*s may be returned for a single ring, causing redundant enumerations. To handle this issue, a local maxima elimination approach is used to remove redundant *pc*s from the result set of the prune phase.

A local maxima is a value of a function which is relatively maximum within a neighborhood. In local maxima filtering, a technique used in aerial imagery, a local maxima is defined as a pixel/pixel group which has a value greater than its surrounding neighbors within a specified neighbor size [180].

In Figure 3.11(a), the cells tagged as $1, 2, 3, 4, 6, 7, 8, 9$ are the neighbors of the cell tagged as 5. Tags are added to the figure for convenience but not used in the algorithm. *Log* $\widehat{LR_p}$ of these 9 cells (Table 3.11(b)) show that all have *Log* $\widehat{LR_p} \geq \theta$. In the previous DGP approach, these 9 cells would be returned to the refine phase even though they all represent the same ring. In the new approach, local maxima cells are identified after every iteration of the prune phase and the rest of the cells are eliminated from the result set. In Figure 3.11(a), only *pc* tagged as 5 will be returned to the refine phase.



(a) Tags of the cells inside the ring-shaped hotspot.

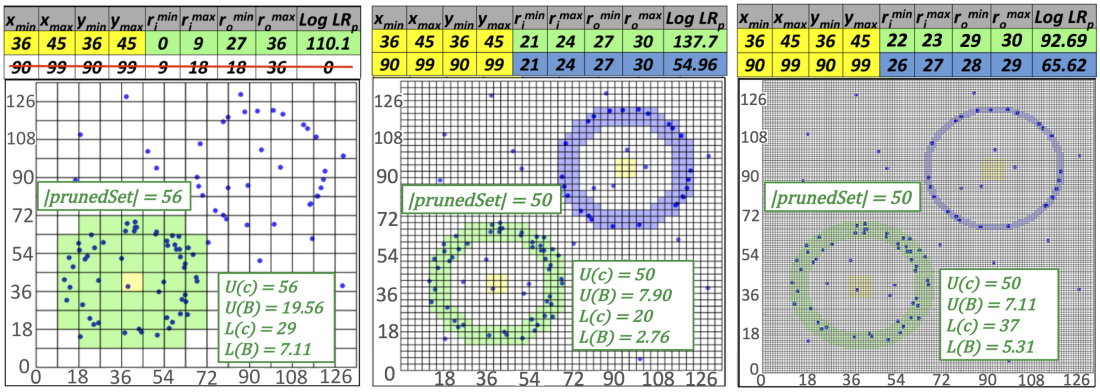
Tag	x_{min}	x_{max}	y_{min}	y_{max}	r_i^{min}	r_i^{max}	r_o^{min}	r_o^{max}	Log LR_p
5	42	45	42	45	21	24	27	30	137.7
6	45	48	42	45	21	24	27	30	123.7
8	42	45	39	42	24	27	30	33	108.5
2	42	45	45	48	21	24	27	30	103.6
4	39	42	42	45	21	24	30	33	101.7
3	45	48	45	48	18	21	24	27	95.06
7	39	42	39	42	21	24	27	30	89.47
9	45	48	39	42	21	24	27	30	89.27
1	39	42	45	48	21	24	30	33	84.02

(b) Result of pruning phase (for a single run with $lg_w = 3$ and $lg_c = 3$ and $\theta = 50$

Figure 3.11: Illustration of local maxima elimination approach. Param-cell *pc* with $tag = 5$ will be returned after the local maxima elimination.

Lemma 6 *When there are multiple cells inside the* $circle_i$ *of a ring, the cell which is closer to (or includes) the actual center of the ring will have higher Log* $\widehat{LR_p}$.

Proof 6 *Without loss of generality, let the cell with Tag 1 have the highest Log* $\widehat{LR_p}$ *in Figure 3.11. Then in order to have* $U(c)^{Tag1} \geq U(c)^{Tag5}$ *and* $L(c)^{Tag1} \leq L(c)^{Tag5}$, *the inner radius should be* $r_i^{Tag\ 1} \leq r_i^{Tag\ 5}$ *or the outer radius should be* $r_o^{Tag\ 1} \geq r_o^{Tag\ 5}$ *(suppose* r_i *and* r_o *are multiples of grid cells). Since Log* $\widehat{LR_p}$ *is computed by* $U(c)$, $L(c)$, $U(B)$ *and* $L(B)$, *either case will also require* $U(B)^{Tag1} \geq U(B)^{Tag5}$ *and* $L(B)^{Tag1} \leq L(B)^{Tag5}$. *But if* $r_i^{Tag\ 1} \leq r_i^{Tag\ 5}$ *or* $r_o^{Tag\ 1} \geq r_o^{Tag\ 5}$ *then it is impossible to have* $U(B)^{Tag1} \geq U(B)^{Tag5}$ *and* $L(B)^{Tag1} \leq L(B)^{Tag5}$ *due to the area of the rings computed using their radii. Thus the lemma is proved by contradiction. Therefore, when there are multiple cells inside the* $circle_i$ *of a ring, the cell which is closer to (or includes) the actual center of the ring will have the highest Log* $\widehat{LR_p}$.



(a) Output of the first iteration with $lg_w = 9$ and $lg_c = 9$

(b) Output of the second iteration with $lg_w = 3$ and $lg_c = 9$

(c) Output of the final iteration with $lg_w = 1$ and $lg_c = 9$

Figure 3.12: Illustration of multi cell length approach. Initially cell lengths are selected as $lg_w = 9$ and $lg_c = 9$. Second row for the top right ring is shown for illustration purposes. This ring will be missed since the first iteration will prune it out.

Algorithm 2 DGP Algorithm with new Prune Phase

Inputs are the same as in Problem Statement

Select lg_c and lg_w using $(r_o - r_i)_{min}$ and $(r_i)_{min}$

Discretize S to $[N_c \times N_c]$ grids using lg_c

Discretize S to $[N_w \times N_w]$ geo-grids using lg_w and set B_{cell}

Create param-grid $[N_c \times N_c \times N_w \times N_w]$ with lg_w

for each param-grid cell pc in $p_{1...n}$ **do**

for each $r_i \leftarrow$ 1 *to* $N_w/2 - 3$ **do**

for each $r_o \leftarrow (r_i + 3)$ *to* $N_w/2$ **do**

$pc \leftarrow Log\ \widehat{LR_p}$

for each pc with $Log\ \widehat{LR_p} \geq \theta$ **do**

$prunedSet_p \leftarrow a_{1...|A|} \in$ min.bou.geo-sub-grid

4.1.2 Local Maxima Elimination

for each pc with $Log\ \widehat{LR_p} \geq \theta$ **do**

Define neighborhood of parametric cell pc

if $\widehat{LR}_{neighbor} \leq Log\ \widehat{LR_p}$ **then**

Remove $pc_{neighbor}$ and $prunedSet_{neighbor}$

4.1.3 Multi Cell Size for the Width of the Ring

for each pc with $Log\ \widehat{LR_p} \geq \theta$ and $prunedSet_p$ **do**

$prunedSet_{prev} = prunedSet_p$ and $prunedSet_{new} = \emptyset$

while $lg_w \geq (r_o - r_i)_{min}$ **do**

if $|prunedSet_{prev}| == |prunedSet_{new}|$ **then** break;

$lg_w = lg_w/3$

Select S of $prunedSet_p$ as S_p

Discretize S_p to $[N_w \times N_w]$ geo-grids

Create param-grid for $[1 \times 1 \times N_w \times N_w]$

for each $r_i \leftarrow$ 1 *to* $(N_w - 3)/2$ **do**

for each $r_o \leftarrow (r_i + 3)$ *to* $N_w/2$ **do**

$pc_{new} \leftarrow Log\ \widehat{LR_{p_{new}}}$

if $Log\ \widehat{LR_{p_{max}}} < \theta$ **then**

Remove pc and $prunedSet_p$;

$prunedSet_{new} \leftarrow a_{1...|A|} \in$ min.bou.geo-sub-grid

$prunedSetFinal \leftarrow prunedSet_{new}$

if $|prunedSetFinal| \geq 4$ **then**

Send $prunedSetFinal$ to the Refine Phase

Multi cell size for the width of the ring

In DGP, the cardinality of *prunedSet*s depends on the cell length. When there are activity points which will not contribute to the actual ring, the *prunedSet* may have a larger size that will cause the refine phase to enumerate unnecessary rings. Our new approach runs *prunedSet*s through the prune phase multiple times. On every iteration, lg_w is divided by a constant (e.g. 1/3 to keep the center the same on each iteration), the prune phase is re-run, and the cardinality of the *prunedSet* is compared with the previous iteration. Once cardinality stabilizes or $lg_W \leq (r_o - r_i)_{min}$, the iteration ends and the *prunedSet* is sent to the refine phase if $|prunedSet| \geq 4$. Figure 3.12 shows sample iterations of this multi cell size approach. Since the cardinality of the *prunedSet* in the third iteration (Figure3.12(b)) is the same for the second iteration (Figure3.12(c)), the algorithm terminates. Note that the top right ring was missed in the first iteration and did not appear in the result set; it is shown to illustrate the change of $Log\ \widehat{LR_p}$ on each iteration.

3.4.2 Refine Phase: A Best Enclosing Ring Heuristic

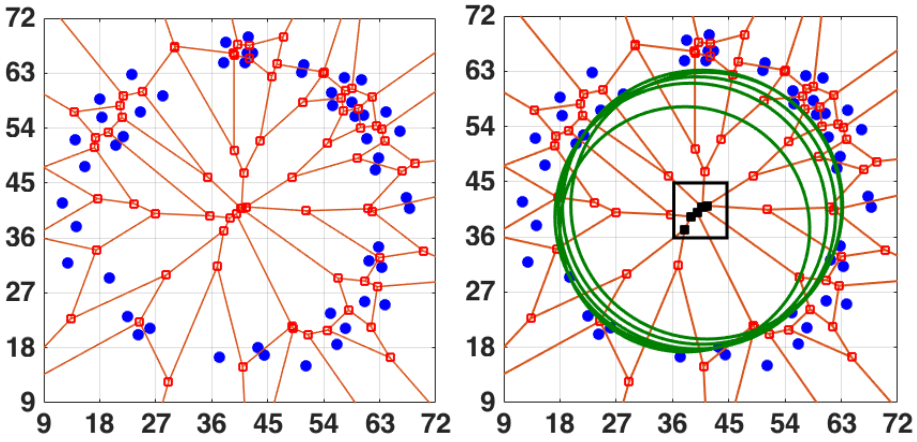
The DGP refine phase has a high computational cost since it enumerates 4P-CRings using all possible triplets of activities in *prunedSet*s (i.e, $O(|prunedSet|^4)$). In order to reduce this cost, we used a best enclosing ring heuristic for each *prunedSet*. In this approach, we use the basic property of Voronoi diagrams that states for each Voronoi vertex, there are 3 points that are equidistant to that vertex. In other words, those Voronoi vertices are the centers of the circles that can be enumerated using those 3 points. Since the prune phase also returns a center range, we simply determine the Voronoi vertices that fall into those center ranges and create inner circles with those vertices and point triplets. In addition, we iteratively remove points and reconstruct the Voronoi diagram to enumerate 4P-CRings not only with whole of the points but also with the subsets of the points in prunedSet.

Note that there is a trade-off between scalability and quality of result set. This approach is used to improve scalability by sacrificing the guarantee of having rings which may have higher log likelihood ratio.

Definition 13 ***Voronoi Diagram:*** *Given a prunedSet with* $|prunedSet| \geq 4$*, a*

Voronoi diagram is a division of a 2 dimensional Euclidean space into $Region_{i=1...|prunedSet|}$ *around each point* $a_i \in prunedSet$ *such that any arbitrary point* (x) *in* $Region_i$ *is closer to* a_i *than any point in prunedSet [181]. Formally,* $Vor(prunedSet) = \{x : dist(a_i, x) < dist(a_j, x)\ \forall\ a_i, a_j \in prunedSet\}$.

For any Voronoi vertex v_i*, there are* 3 *points* a_1, a_2, a_3 *that are equidistant to* v_i*. A circle passing through* $a_{1,2,3}$ *has a center at* v_i*. If any* 4 *points are co-circular, there is a duplicate circle (same circle is enumerated twice). If any* 3 *points are collinear,* v_i *is a line (no circle is enumerated).*



(a) *prunedSet* and the corresponding Voronoi diagram

(b) Inner circles created in the first iteration of the refine phase

Figure 3.13: Inner circles created in the first iteration of the refine phase. Blue dots represent the points in *prunedSet*, red squares represent the Voronoi vertices. Inner circles are created with the vertices inside the black square representing the center interval returned by prune phase (best in color).

Algorithm 3 starts by creating a Voronoi diagram of the *prunedSet* (line 1). The Voronoi vertices v_i which lie on the center intervals returned by the prune phase are then used as the center of 4P-CRings (lines 3-4). For these vertices v_i, three closest points $a_{1,2,3} \in prunedSet$ are determined and used to create the inner circle ($circle_i$) (line 5). The points in the outer radius interval r_o^{min}, r_o^{max}, returned by the prune phase, are used to create the outer circle ($circle_o$) (lines 7-8). If $circle_i$ and $circle_o$ satisfy the minimum ring width and minimum inner radius thresholds, the $Log\ LR_R$

of the 4P-CRing is computed and saved in the *ringList* (lines 9-10). Once all Voronoi vertices v which lie on the center intervals are traversed to enumerate the 4P-CRings, the points associated with these vertices are removed from the *prunedSet*, the Voronoi diagram is reconstructed(lines 12-13) and the algorithm re-runs with the new Voronoi vertices. Finally, the R with the highest $Log\ LR_R$ is returned (line 14).

Figure 3.13 illustrates the first iteration of the refine algorithm. Once the *prunedSet* is returned by the prune phase (points are shown as blue dots), a Voronoi diagram is created as shown in Figure 3.13(a) (voronoi vertices are shown as red squares). Inner circles ($circle_i$) of 4P-CRings are created using the Voronoi vertices which are inside center intervals defined by pc. In Figure 3.13(b), the black squares represent the vertices and the circles represent the $circle_i$. Once $circle_i$s are created, $circle_o$s are created using the points beyond these. Next, the Voronoi vertices which were used to enumerate $circle_i$ are removed from the *prunedSet* and the Voronoi diagram is reconstructed. If any of the new vertices fall inside the center interval, the same process is repeated to create new 4P-CRings. Once the *prunedSet* is empty or there is no vertex inside center interval, the algorithm terminates and returns the R with the highest $Log\ LR_R$.

Algorithm 3 Refine Phase and Monte Carlo Sim. of DGPLMR

Input:

1) Param-grid cell pc and $prunedSet$,

2) A minimum ring width $(r_o - r_i)_{min}$ and inner radius $(r_i)_{min}$,

3) A log likelihood ratio threshold θ

Output:

Output is the same as Algorithm 1 (DGP

Refine Phase:

Create the Voronoi diagram of $prunedSet$

while $prunedSet \neq \emptyset$ **do**

for each Vertex v of the Voronoi diagram **do**

if $x_{min} \geq v_x \geq x_{max}$ and $y_{min} \geq v_y \geq y_{max}$ **then**

$circle_i \leftarrow$ three closest $a \in prunedSet$

while $dist(a, v) \geq r_o^{min}$ **do**

$circle_o \leftarrow$ farthest $a \in prunedSet$

Ring $R \leftarrow circle_i$, $circle_o$ and $Log\ LR_R$

if $(r_o - r_i)_R \geq (r_o - r_i)_{min}$ and $(r_i)_R \geq (r_i)_{min}$ and $Log\ LR_R \geq \theta$ **then**

Save R in $ringList$

Remove a from $prunedSet$

Remove points associated with v

Reconstruct Voronoi diag. with new $prunedSet$

$candidateRings \leftarrow R$ with highest $Log\ LR_R$

Monte Carlo Simulation Phase:

$significantRings \leftarrow R$ if $p\text{-}value \leq \alpha_p$

Return $significantRings$

3.5 Theoretical Evaluation

In this section we analyze the correctness, and computational complexity of DGPLMR. Prune Phase Algorithm (Algorithm 2) does not return an output and simply returns prunedSets for the refine phase algorithm. Therefore, correctness of DGPLMR is proved by the correctness of the Refine phase algorithm (Algorithm 3). A proof of completeness

is not given since Local Maxima Elimination and the Best Enclosing Ring heuristic trade completeness for scalability.

3.5.1 Correctness

Lemma 7 *DGPLMR is correct. Correctness means that DGPLMR returns only the 4P-CRings with* $(r_o - r_i)_R \geq (r_o - r_i)_{min}$ *and* $(r_i)_R \geq (r_i)_{min}$*, Log* $LR_R \geq \theta$ *and p-value* $\leq \alpha_p$*.*

Proof 7 *In Algorithm 3, each 4P-CRing is evaluated against* $(r_o - r_i)_{min}$*,* $(r_i)_{min}$ *and* θ *thresholds on line 9, and any satisfying these is saved in candidateRings. Once the refine phase is done, Monte Carlo simulation phase starts and candidateRings are evaluated against the* α_p *threshold on line 15. Finally, DGPLMR returns the candidateRings with p-value* $\leq \alpha_p$*. Thus DGPLMR is correct.*

Lemma 8 *Given a finite activity set* $A = a_1, a_2, ..., a_n$*, DGPLMR terminates in finite time.*

Proof 8 *In Algorithm 2, the prune phase iterations on lines 5-7 of DGPLMR are bounded by the number of param-grid cells and the algorithm will terminate after all param-grid cells are visited. Also Multi Cell Size pruning will terminate once* $lg_w < (r_o - r_i)_{min}$ *since* lg_w *is divided by a constant (e.g.* 3*) on every iteration (line 19). In Algorithm 3, the refine phase of DGPLMR removes points* $a \in prunedSet$ *on each iteration of the loop (lines 6-11) and terminates when all a in the range of* r_o^{min} *and* r_o^{max} *are visited. Also the prunedSet size decreases on each iteration (lines 2-13) and it terminates when* $prunedSet = \emptyset$*. Since both phases terminate, DGPLMR terminates in finite time.*

3.5.2 Computational Analysis

Lemma 9 *Given an activity set* $|A|$*, DGPLMR has a lower computational cost than DGP, if* $N_c^2 \times N_w^2 \leq N^4$ *and* $N \geq N_w$*.*

Proof 9 *The worst case computational cost of DGPLMR for a single run is* $O(N_c^2 \times N_w^2 + |A|^2\ log\ |A|)$ *whereas the worst case cost of DGP is* $O(N^4 + |A|^5)$*. Thus, DGPLMR*

prune phase is less costly than DGP prune phase. Moreover, $|A|^2\ log\ |A| \leq (|A-1|)^5$*, meaning DGPLMR's refine phase cost is less than DGP's. Therefore, DGPLMR has a lower computational cost than DGP if* $N_c^2 \times N_w^2 \leq N^4$.

The DGP algorithm prune phase uses a dual grid based ring enumeration on a geometric grid space of N^2 and a parametric space of N^4, where N^2 is the number of geo-grid cells created using the grid cell size lg. The worst case cost of DGP is $O(m \times (N^4 + |A|^5))$, if no pruning occurs and all activity points are returned to the refine phase. In the best case, the cost will be $O(m \times N^4)$ as the prune phase will not return any *prunedSet* and the algorithm will terminate.

DGPLMR has a worst case cost of $O(m \times (N_c^2 \times N_w^2 + |A|^2\ log\ |A|)$, that is, when the algorithm does not prune any points and the refine phase uses $|A|$ to generate an actual 4P-CRing. In the best case, the cost will be $\Omega(m \times N_c^2 \times N_w^2)$ as the prune phase will not return any *prunedSet* and thus the refine phase will not run and the algorithm will terminate.

Note that N_c and N_w are defined by lg_c and lg_w which are selected using the minimum inner radius $(r_i)_{min}$ and minimum ring width $(r_o - r_i)_{min}$ inputs. Thus, a careful selection of these inputs is important since they affect the execution times. A summary of the computational complexity of DGP and DGPLMR is given in Table 3.4.

Table 3.4: Computational complexity analysis

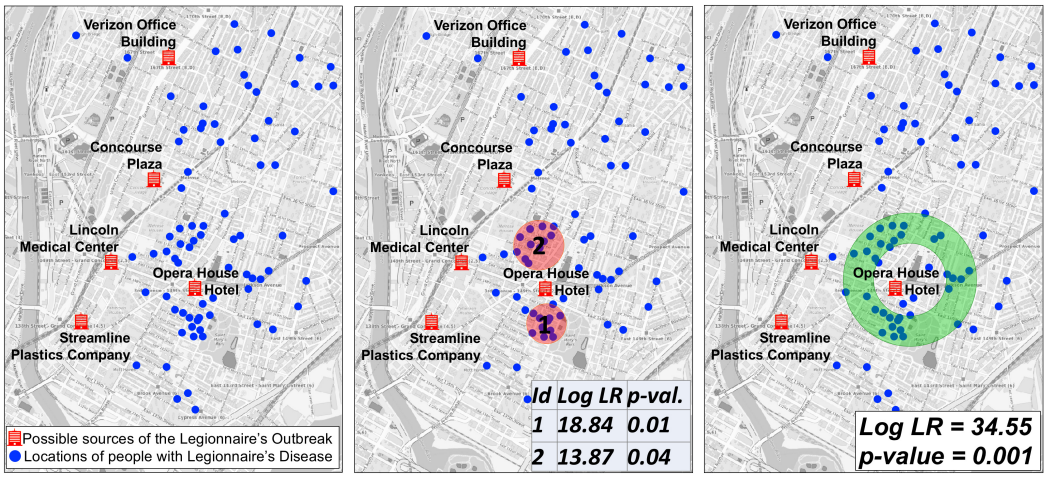
Algorithm	Complexity (best case)	Complexity (worst case)
DGP [141]	$\Omega(m \times N^4)$	$O(m \times (N^4 + \|A\|^5))$
DGPLMR (proposed approach in this chapter)	$\Omega(m \times N_c^2 \times N_w^2)$	$O(m \times (N_c^2 \times N_w^2 + \|A\|^2\ log\ \|A\|))$

3.6 Case Study

Statistically significant hotspot detection techniques based on diffusion theory, such as SaTScan, find hotspots with simply connected (e.g. circles) shapes. However, in some domains, activities may occur in a ring-shaped pattern. To evaluate our approach, we

compared the output of SaTScan and DGPLMR on real world data using a continuous Poisson process [10].

We conducted our case study on a New York Bronx Legionnaire's disease outbreak [6] dataset as shown in Figure 3.14(a). The input activity set included 78 infected individuals' locations as well as the possible sources of the outbreak which were hand digitized using the map in [6]. SaTScan returned two small circular hotspots (Figure 3.14(b)), while RHD returned a single ring-shaped hotspot (Figure 3.14(c)) with a low p-value (i.e. p-value= 0.001) and a high $Log\ LR_R$ (i.e. $Log\ LR_R = 34.55$). Later, we validated that the Opera House Hotel inside the inner circle of the ring was indeed the source of the outbreak [7].



(a) Legionnaire's in New York (2015)

(b) Output of SaTScan

(c) Output of RHD

Figure 3.14: New York Bronx Legionnaire's disease cases (collected from the map in [6]) in 2015 and the output of SaTScan and RHD. RHD output validates the actual location of the outbreak as Opera House Hotel [7].

3.7 Experimental Evaluation

The goal of the experiments was twofold: (a) *a comparative analysis* to evaluate the effect of the new algorithmic refinements of DGPLMR compared to DGP. (b) *a sensitivity analysis* to determine the scalability and potential bottlenecks in DGPLMR.

In the comparative analysis, five questions were asked: (1) What is the effect of the activity set ($|A|$) size? (2) What is the effect of the number of activities in R? (3) What is the effect of the selected cell size (lg_c and lg)? (4) What is the effect of the number of rings? (5) What is the effect of the log likelihood ratio threshold (θ)? The following candidate algorithms were included in the comparative analysis:

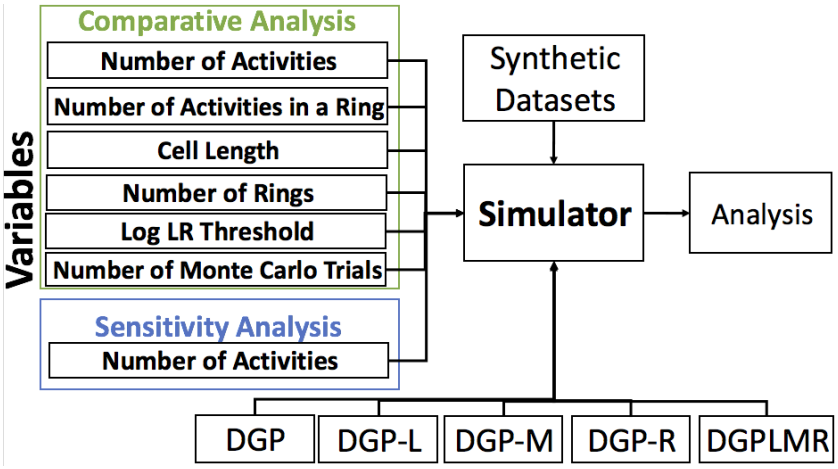


Figure 3.15: Experimental Design

In sensitivity analysis we asked: "Where is the bottleneck in DGPLMR?" What is the effect of number of activities in A, number of activities in R and cell length (lg_c) on the memory cost of DGPLMR?

3.7.1 Experimental Design & Data Sets

Activity sets were generated under a null hypothesis (Complete Spatial Randomness), and ring-shaped hotspots were added (20% of the activity set size). The experimental design is illustrated in Figure 3.15 and the parameter details are listed in Table 3.5. The likelihood ratio threshold was user defined and set to $\theta = 2500$. Experiments were performed on a computer with Intel Xeon 2.67 GHz CPU and 4GB Java VM memory.

Table 3.5: Experiment datasets generated for the Experimental Evaluation ($S = 900 \times 900$, $m = 0$).

Experiment (Effect of)	$\|A\|$	$\|R\|$ (% of $\|A\|$)	# of R	θ	$(r_i)_{min}$	$(r_o - r_i)_{min}$
$\|A\|$ (Comp)	Vary	20%	1	2500	50	20
$\|R\|$	5000	Vary	1	2500	50	20
lg, lg_c	5000	20%	1	2500	Vary	Vary
# of R	5000	20%	Vary	2500	50	20
θ	5000	20%	1	Vary	50	20
$\|A\|$ (Sens.)	Vary	20%	1	50000	50	20
σ^2 (Sens.)	Vary	Vary	1	2500	50	20

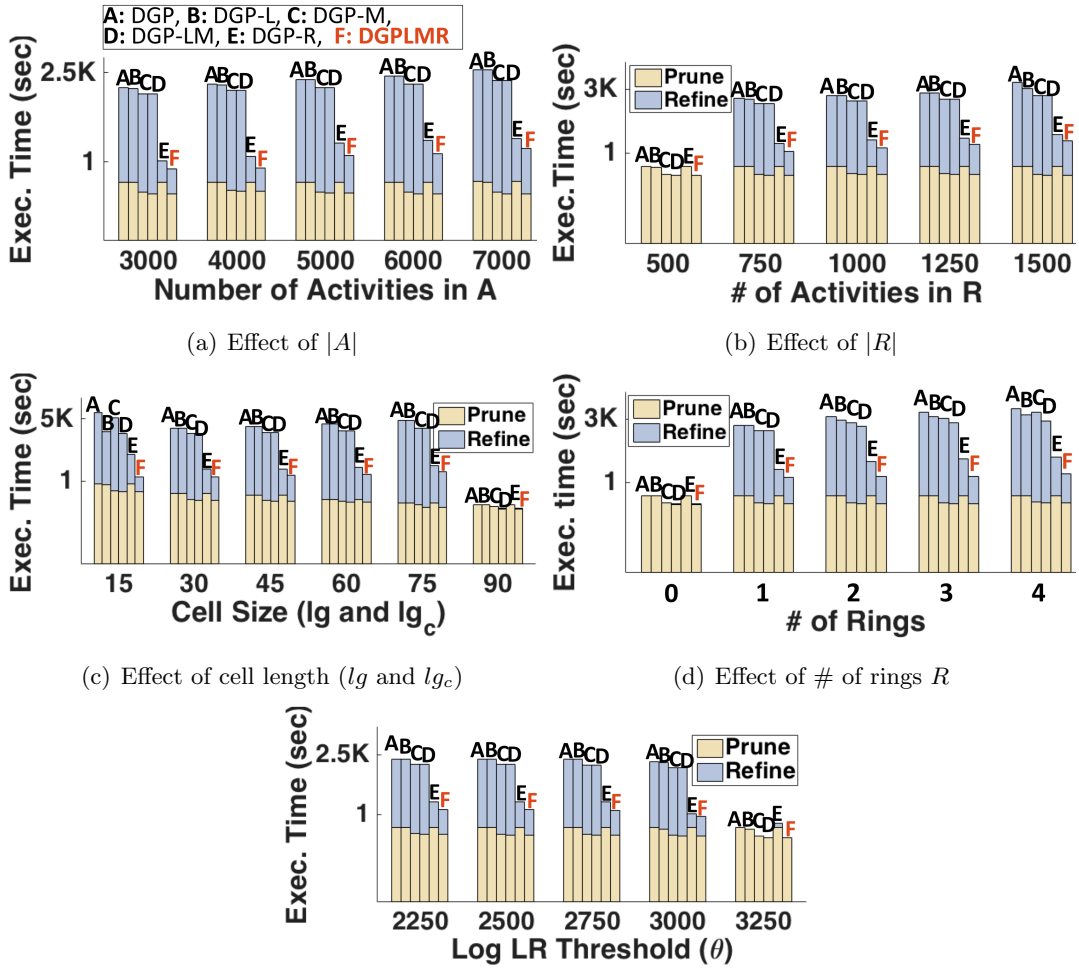
- Dual Grid Based Pruning Algorithm (DGP) [141]
- DGP with Local Maxima Elimination(DGP-L)
- DGP with Multi Cell Length (DGP-M)
- DGP with Local Maxima Elimination and Multi Cell Length (DGP-LM)
- DGP with Best Enclosing Ring Refine Phase (DGP-R)
- DGP with All Algorithmic Refinements (DGPLMR)

3.7.2 Comparative Analysis

Effect of the number of activities in the activity set ($|A|$): First, activity set sizes were varied from 3000 to 7000 activity points and the effect of the algorithmic refinements on DGPLMR execution times were observed. Figure 3.16(a) shows that each refinement improves execution time of DGPLMR and there is at least orders of magnitude difference between the execution times of DGP and DGPLMR.

Effect of the number of activities in the ring $|R|$: Activity counts inside the rings were varied from 500 to 1500. As can be seen in Figure 3.16(b), when the number of activities in the ring is small (i.e. 500), none of the algorithms can detect the ring,

because the pruning phase eliminates this ring using the θ threshold. Also it can be concluded that DGPLMR runs fastest for all ring activity counts tested ($|R|$).



(a) Effect of $|A|$

(b) Effect of $|R|$

(c) Effect of cell length (lg and lg_c)

(d) Effect of # of rings R

(e) Effect of θ

Figure 3.16: Comparative Analysis. Execution time for DGP (A), DGP with Local Maxima Elimination (B), DGP with Multi Cell Size Approach (C), DGP with both Local Maxima Elimination and Multi Cell Size Approach (D), DGP with Best Enclosing Ring Heuristic (E) and DGPLMR with all algorithmic refinements (F) (best in color).

Effect of the cell length: We compared the effect of cell length. For DGP, DGP-L and DGP-R, we varied cell length lg from 15 to 90, and for DGP-M, DGP-LM, DGPLMR

we varied lg_c from 15 to 90. It is worth noting that DGP-M, DGP-LM and DGPLMR use another cell length for the width of the ring lg_w which is selected the same as lg_c initially. In Figure 3.16(c), it can be seen that when the cell lengths are smaller (i.e. 15), all algorithms' execution times are higher due to the increased prune phase execution time. Between length 30 and 75, there is a slight increase in execution times. When the cell lengths are larger, prunedSet includes more points than the actual hotspot, causing a longer execution time for the refine phase. The sharp decrease after 75 occurs because the large cell size caused the prune phase to miss the rings and return an empty prunedSet. It can be concluded that a careful selection of cell length is important for the performance of the DGPLMR algorithm.

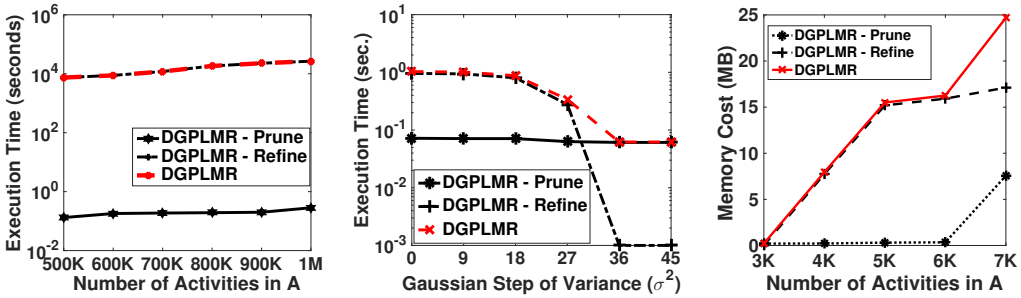
Effect of the number of rings: The number of rings was varied from 0 to 4 and the activities outside these rings were generated using complete spatial randomness (CSR). In Figure 3.16(d), when the number of rings is 0, all algorithms successfully end on their prune phase. Also the number of rings does not affect the execution times of the algorithms a lot. Finally, it can be seen that DGPLMR runs fastest followed by DGP-R.

Effect of the Log Likelihood Ratio Threshold θ: Log likelihood ratio threshold θ was varied from 2250 to 3250 and the effect on execution times was observed. Figure 3.16(e) shows the results. It can be seen that DGPLMR benefits from θ most. Also, when $\theta = 3250$, all algorithms missed the ring in the activity set. Again DGPLMR is the fastest performing algorithm but this experiment also shows that the careful selection of θ is important to detect rings.

In summary, the experiments show that DGPLMR with prune phase algorithmic refinements and the new refine phase outperforms DGP [141] by at least two orders of magnitude. Also it can concluded that most savings are achieved by the new refine phase algorithm.

3.7.3 Sensitivity Analysis

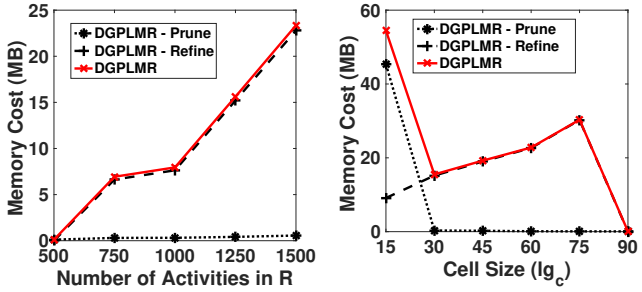
Effect of large activity sets ($|A|$): Activity sets with $500K$ to $1M$ points were generated and DGPLMR was run with $\theta = 50000$. Figure 3.17(a) shows that the bottleneck of DGPLMR is the refine phase algorithm. The prune phase is not affected by large $|A|$ and performs faster.



(a) Exec. time for large A (DGPLMR and Refine phase overlap)

(b) Exec. time for different Gaussian steps of variance

(c) Mem. Cost for varying A



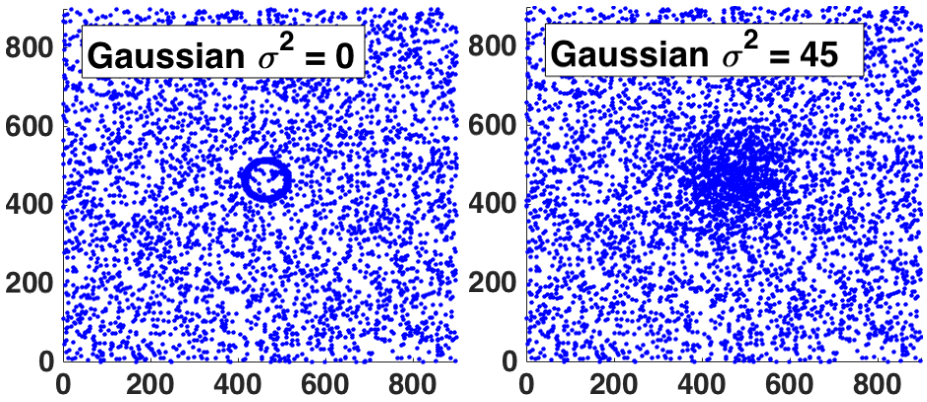
(d) Mem. Cost for varying number of activities in R

(e) Mem. Cost for varying lg_c

Figure 3.17: Sensitivity analysis of DGPLMR algorithm (best in color).

Effect of the variance of activity distribution of rings: To see the effect of the underlying point process that generates rings, we created rings using Gaussian distribution with different Gaussian steps of variance [182]. Initially, we created 4000 completely spatially random points in a study area with 900×900 square units. Then we created 1000 activity points in a R with $r_i = 45$, $r_o = 54$ as shown in Figure 3.18(a). Next, we diffused each of these 1000 points via a random walk with Gaussian steps of variance $\sigma^2 = 9$ to create different activity sets with $\sigma^2 = 9,\ 18,\ 27,\ 36$ and 45. Note that by the time $\sigma^2 = 45$, the amount of diffusion was sufficient to make those 1000 activity points form a circle instead of a ring as shown in Figure 3.18(b). Figure 3.17(b) shows that the increased σ^2 does not affect the prune phase. However, the sharp decrease on the refine phase execution time after $\sigma^2 = 36$ indicates that DGPLMR prune phase did not return any *prunedSet* to the refine phase since the points in the ring diffused

too much and their $Log\ LR_R$ did not exceed the $\theta = 2500$ threshold.



(a) Activity set with a ring generated with $\sigma^2 = 0$

(b) Activity set after points are scattered by a Gaussian $\sigma^2 = 45$

Figure 3.18: Two of the activity sets that were used for the experiment on the variance of activity distribution of rings (best in color).

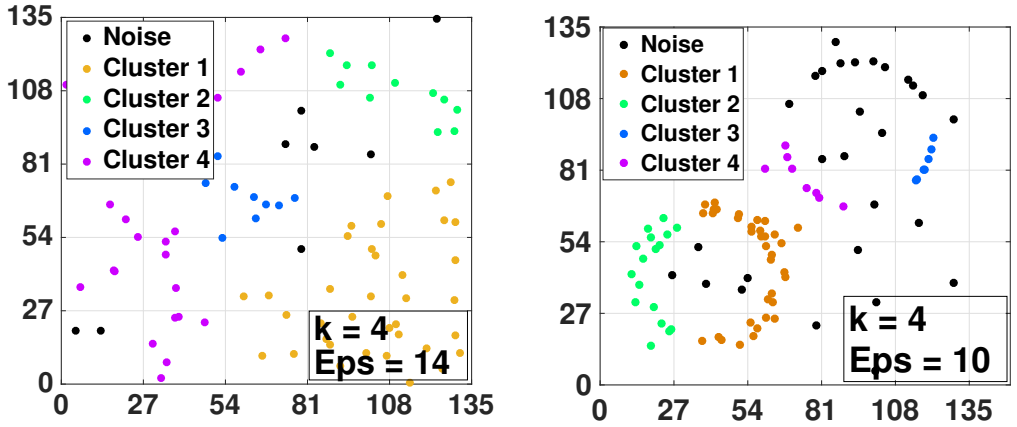
Memory Cost Analysis: We also analyzed the memory cost of DGPLMR under varying number of activities in A (Figure 3.17(c)), varying number of activities in R (Figure 3.17(d)) and varying cell size (Figure 3.17(e)). We can see from Figure 3.17(c) and Figure 3.17(d) that the DGPLMR prune phase does not affect memory when the cell size is constant. In Figure 3.17(c), the memory cost increase of the prune phase after $|A| = 6000$ indicates that increased number of activities in the study area caused the prune phase to do several iterations with multi cell size pruning, increasing the number of cells in memory. Also, in Figure 3.17(e), we can observe that smaller cell sizes cause the prune phase to consume more memory. In contrast, larger cell lengths cause the refine phase to use more memory to enumerate rings. Also the sharp decrease of refine phase memory cost on $lg_c = 90$ indicates that large cell sizes may cause missing rings.

In summary, sensitivity analysis shows that DGPLMR handles large activity sets in reasonable time and memory usage.

3.8 Discussion

Other Approaches: Traditional density based clustering techniques shouldn't be neglected although they do not take statistical significance into account. These techniques lack a metric (e.g. log likelihood ratio test) for ranking candidate clusters which cause adding a significance test non-trivial. Two representative density based clustering techniques are DBSCAN [9] and CLIQUE [183]. DBSCAN employs user specified parameters to reduce the effect of noise and to detect cluster density, whereas CLIQUE uses a grid based approach to find regions which are defined by their density. Although these are computationally inexpensive and can easily find arbitrarily shaped clusters in a space, since they lack a test for statistical significance, they sometimes produce false positive results. This limits their applicability in domains such as epidemiology and criminology where false positives can result in stigmatizing, economic loss, along with social/political challenges. To reinforce our point, we present two more examples that highlight the value of RHD over traditional clustering techniques i.e. DBSCAN. In Figure 3.19(a), DBSCAN generated 4 clusters for a complete spatial random dataset with 100 number of points since it lacks a statistical significance test whereas RHD with DGPLMR output was an empty set since it eliminates chance patterns using a statistical significance test. We also ran DBSCAN for the same input in Figure 3.5(a). Again, DBSCAN generated 4 clusters (3.19(b)) and these do not align with the ring-shaped hotspots detected by RHD (Figure 3.5(c)).

Post-processing of the Output: DGPLMR returns all rings that satisfy the thresholds in the Section 3.2. Just like popular statistically significant hotspot detection methods (e.g. SaTScan [10]), our output sometimes includes a large number of rings which may be overwhelming for users. In order to reduce the number of rings that are returned by the proposed approach, several strategies can be used. First, likelihood ratio and p-value thresholds may be chosen more selective. Second, rings can be ordered by their log likelihood ratio values to select top few rings. Thus, analysts can focus on the most interesting rings first. Third, the overlapping rings which may cause visual clutter may be eliminated from the output. This can be done either by an iterative manner as described in [10], or by a post-processing that may remove rings with the same points or overlapping areas.



(a) DBSCAN output for a random activity set (b) DBSCAN output for the activity set in Figure 3.1(a)

Figure 3.19: DBSCAN output for a random activity set and an activity set with rings (best in color).

3.9 Conclusions and Future Directions

This work explored the problem of ring-shaped hotspot detection (RHD) in relation to application domains such as criminology, biology, and epidemiology. A DGPLMR (**D**ual **G**rid Based **P**rune with **L**ocal maxima elimination, **M**ulti cell length and **R**efine with best enclosing ring) algorithm is proposed which discovers significant ring-shaped hotspots in a study area defined by an activity set. The proposed approach uses upper bound log likelihood ratio pruning as well as algorithmic refinements to enhance its scalability. Case study comparing the proposed method with a diffusion theory based approach (i.e., SaTScan) on a real dataset is presented. Experimental evaluation using synthetic datasets indicated that the proposed algorithmic refinements yield substantial computational savings. In future, we plan to add the effect of the underlying population in order to distinguish ring-shaped hotspots in rural areas from those in metropolitan areas. In addition, temporal dimension and underlying factors (i.e. geographic barriers, spatial networks) that generate irregular ring-shaped hotspots will be explored.

Chapter 4

Geographically Robust Hotspot Detection

4.1 Introduction

Given a set of geolocated points (e.g. locations of a disease/crime), a minimum radius, a minimum likelihood ratio and a significance threshold, Geographically Robust Hotspot Detection (GRHD) finds hotspot areas where the concentration of points inside is significantly higher than the number of points outside. This paper refers geographically robustness as the ability of being insensitive to minor gaps (e.g, contiguous hotspots separated by road segments, rivers, etc.) in the contiguity of the spatial pattern.

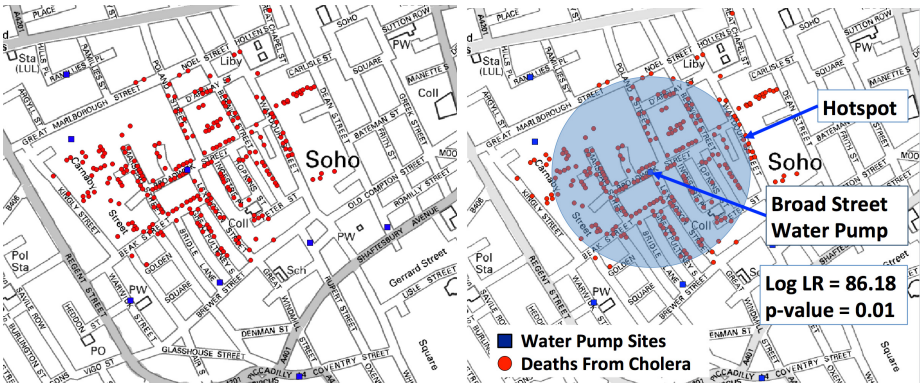
An example of a hotspot is shown in Figure 4.1. Example shows the London Cholera Outbreak in 1854 including 491 deaths aggregated on 250 house locations (in red) and 8 water pump locations (in blue). The detected hotspot (in blue) has a high test statistic value (i.e. log likelihood ratio) and a high statistical significance (i.e. p-value = 0.01). As can be seen in Figure 4.1(b), the detected hotspot points out the infected water pump, which may have caused the Cholera deaths.

4.1.1 Application Domain:

GRHD is important for application domains such as epidemiology, environmental criminology, ecology, medical imaging, biology, etc. where detection of circular hotspots may

reveal important information for domain experts. There are three important concepts related to the detection of hotspots in these domains: Elimination of chance hotspots (prevent false positives), ability to detect a hotspot if it exists (geographically robustness), modeling hotspots with respect to diffusion model.

Elimination of chance hotspots (i.e. false positive) is important since false positive hotspots may result in poor allocation of resources and inefficient management in critical situations (e.g. crime, epidemic). Moreover, chance hotspots may result in stigmatizing as such locations will not be visited by people. Thus in these domains (i.e., epidemiology, environmental criminology, etc.), in order to eliminate chance hotspots, statistical significance test is done.



(a) Input Cholera Cases Dataset (b) Output of GRHD

Figure 4.1: 1854 London Cholera Outbreak [8]. The blue hotspot indicates the location of the infected water pump.(best in color).

Another important concept in hotspot detection is the ability to detect a hotspot if it really exists (geographically robustness). In some application domains, the cost of failing to detect a hotspot although it really exists has important consequences (e.g. unnecessary deaths). Thus a hotspot detection technique should manage to detect all existing hotspots.

Spread of infectious diseases and crime is tend to be similar to the diffusion model in physics and chemistry. Diffusion means that molecules or heat will move away from their sources once they are discharged. Similarly, diffusion model provides a natural way to describe the "circular diffusion" of cases (i.e. diseases, crimes). For example,

most infectious diseases move from their source to physically nearest neighbors and these transmit the disease to their nearest neighbors and so on, causing hotspots of cases around the sources of diseases [184]. This also gives rise to circular footprints of hotspots in isotropic geographies.

Next, two example application domains will be introduced to illustrate these three important concepts.

Epidemiology is the study of distribution and determinants of disease spread across human populations and the applications to prevent and control the spread of a disease [185]. In epidemiology, infectious disease cases are known to follow diffusion models. For example, malaria transmission tends to be spatially heterogeneous around a radius [186]. The ability to detect (geographically robustness) these hotspots will help officials to take the required precautions to prevent the diffusion of a disease and elimination of chance hotspots (prevent false positives) will prevent waste of resources. For example, Ebola outbreak in 2014 raised concerns about the poor allocation of resources and the late reaction of the officials [187].

Environmental Criminology is the study of criminal patterns and how crimes are affected by the physical environment of the criminal [14]. An important theory in environmental criminology namely "Crime Pattern Theory" states that most of the serial crimes diffuse within a radius around an anchor point of a criminal [14]. The ability to detect (geographically robustness) these hotspots of crimes will help officials to decide where to deploy new units to prevent new crimes and will point out the "anchor point" to determine the criminal's location [18, 188]. Similarly, elimination of chance hotspots (prevent false positives) will help security officials to focus their efforts to a specific location and will prevent stigmatizing of neighborhoods. For example, a recent study shows that property values are affected by crime hotspots [189].

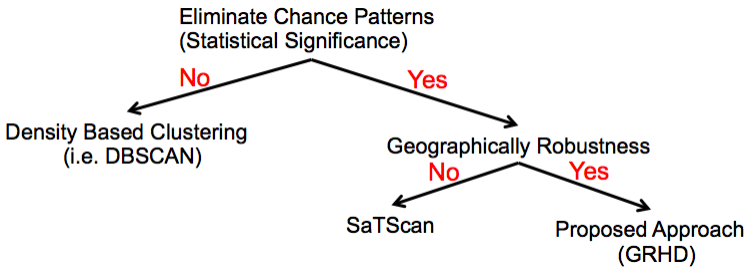


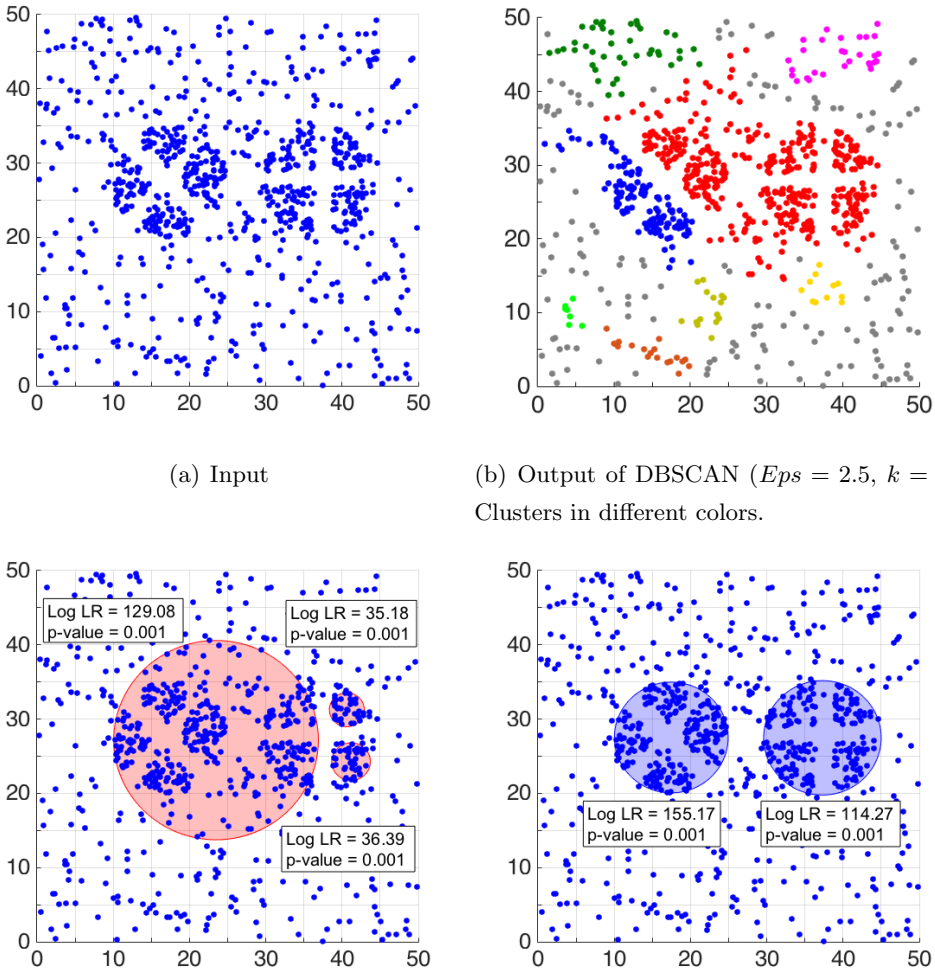
Figure 4.2: Related work of hotspot detection.

4.1.2 Challenges:

GRHD is challenging since it is hard to enumerate all possible candidate hotspots in a study area. Additionally, the location and the radius of the hotspots are not known beforehand which makes it difficult to select an enumeration technique. Once candidate hotspot enumeration is done, those are evaluated by their test statistic with a log likelihood ratio test. There are two challenges associated with the log likelihood ratio test: First, it requires a count of the points inside the enumerated candidate hotspot, causing the whole point set to be scanned for each candidate. Second, it is anti-monotonic, meaning a smaller candidate hotspot may have a higher log likelihood ratio than a bigger one or vice versa. Thus, monotonicity based filtering techniques are inapplicable. Furthermore, the test for statistical significance using randomization (Monte Carlo simulation) multiplies the cost.

4.1.3 Related Work:

Figure 4.2 shows the related work for hotspot detection. There are many techniques to find dense regions (i.e. hotspots) in a study area [9, 28, 183, 190]. For example, density based clustering techniques (i.e. DBSCAN [9], CLIQUE [183], etc.) are capable of finding arbitrarily shaped clusters and their ability to detect a hotspot is high. While these techniques are inexpensive, they tend to generate many false positive hotspots since they lack a statistical significance test. For example, in Figure 4.3 given an input point set with 800 points with two circular hotspots of 200 points each, DBSCAN produced 8 different clusters as shown in Figure 4.3(b).



(a) Input

(b) Output of DBSCAN ($Eps = 2.5$, $k = 6$). Clusters in different colors.

(c) Output of SaTScan

(d) Output of GRHD

Figure 4.3: Example output of Geographically Robust Hotspot Detection (GRHD) compared to DBSCAN [9] and SaTScan [10] (best in color).

SaTScan [10], is widely used for the detection and evaluation of circular hotspots of diseases in epidemiology. It uses spatial scan statistics, which is known as the most powerful statistical test, to eliminate chance patterns (reduce false positives) [178]. However, its reliance on point-centered hotspots is bad under some geographic conditions. For example, some geographic features such as road segments, rivers, etc. may cause gaps or discontiguity on hotspots (see Figure 4.3(a)). Also, some geographic features

(i.e. lakes, mountains, parks) close to the center of a hotspot may cause not to observe points and therefore cause a hotspot with a sparse center. Therefore, SaTScan risks failing to detect a hotspot (is not geographically robust) in those cases. Also, in some cases SaTScan may return very small hotspots that occurred by chance (suppose two or more points are exactly at the same location) since it lacks a minimum radius r_{min} threshold.

In contrast, proposed approach for GRHD does not rely on points to enumerate candidate hotspot centers and thus it is not affected by the sparseness around the center when detecting a hotspot. Also it is not affected by the gaps or discontiguity caused by the geographic features (road network, rivers, etc.). Therefore its ability to detect a hotspot (if it exists) is better (geographically robustness) in those cases. It also uses the spatial scan statistics to eliminate chance patterns (false positives). Moreover, it uses a minimum radius r_{min} threshold, which eliminates very small hotspots that occurred by chance.

4.1.4 Contributions:

This paper formally defines the problem of detecting geographically robust hotspots. To solve this problem, a novel approach namely cubic grid circle algorithm (CGC) is presented. CGC filter phase uses a cubic grid to filter points which will not contribute to a hotspot and refine phase enumerates hotspots using a smallest enclosing circle algorithm. A case study shows that CGC finds hotspots that are not discovered by related work (i.e. SaTScan). Computational analysis and experimental results show that CGC yields substantial computational savings compared to the related work.

4.1.5 Scope and Outline:

This paper focuses on geographically robust diffusion hotspots modeled as circles in a two dimensional isotropic space. The underlying population and other variables associated with the point set are not considered (these will be considered in future work). There may be multiple hotspots in the study area and those are assumed to be non-overlapping as described in the SaTScan user guide [10]. In addition, since this paper is focused on circular hotspots as defined in diffusion model, rectangular [191] and/or

snake-shaped hotspots as well as the predefined locations [192] are out of the scope.

Section 4.2 presents the basic concepts and problem statement of GRHD. Section 4.3 reviews the related work (i.e. SaTScan [10]) solution and describes the proposed Cubic Grid Circle (CGC) algorithm. Theoretical evaluation of the proposed approach is covered in Section 4.4. Section 4.5 presents a case study comparing CGC with SaTScan on a real crime data. The experimental evaluation is covered in Section 4.6. A discussion on the shape of significant hotspot detection is presented in Section 4.7. Conclusions and future work are covered in Section 4.8.

4.2 Basic Concepts and Problem Formulation

This section introduces basic concepts and defines the Geographically Robust Hotspot Detection (GRHD) problem.

4.2.1 Basic Concepts

Definition 14 ***A point set*** *P is a collection of geolocated points (i.e. crime, disease locations). A* ***point*** *$p \in P$ is associated with a pair of coordinates (x, y) representing its spatial location in the study area S.*

Definition 15 ***Study area*** *S is the minimum orthogonal bounding rectangle of P in the two dimensional Euclidean space. The area of S is denoted as $area_S$.*

Definition 16 *Given center coordinates (x, y),* ***Circle*** *C is the closed curve where any point on its perimeter is equidistant with a radius r to its center. Each C has three parameters: x, y coordinates of its center and its radius r. The area of C is denoted by $area_C$.*

Definition 17 ***Log Likelihood Ratio*** *($Log\ LR_C$) is the interest measure that is used as the test statistic for a candidate circle C [74, 177, 178]. The equation can be shown as;*

$$Log\ LR_C = Log\left(\left(\frac{c}{B}\right)^c \times \left(\frac{|P|-c}{|P|-B}\right)^{|P|-c} \times I()\right) \tag{4.1}$$

$$B = \frac{|P| \times area(C)}{area(S)} \quad and \; I() = \begin{cases} 1, \; if \; c > B \\ 0, otherwise, \end{cases}$$

B is the expected and c is the observed number of points in a particular area, $|P|$ is the cardinality of P and $I()$ is an indicator function. $I() = 1$ when the candidate hotspot has more points than expected ($c > B$) and $I() = 0$ otherwise [10].

For example, the area of the circle on the right in Figure 4.3(d) is $\pi * r^2 = 3.14 * 7.71^2 = 186.62$ and $area(S) = 50 \times 50 = 2500$. Thus $B = \frac{800 \times 186.62}{2500} = 59.72$. In this circle C, there are $c = 199$ points. Finally, $I = 1$ since $199 > 59.72$.

Using Equation 4.1, $Log\ LR_C = Log\left(\left(\frac{199}{59.72}\right)^{199} \times \left(\frac{800-199}{800-59.72}\right)^{800-199} \times 1\right) = 114.27$

Definition 18 ***Monte Carlo Simulation (MCS)*** *is a randomization test which is used to get the distribution of the test statistic ($Log\ LR_C$). MCS is done as follows; first, m random point sets ($P_{random_{1...m}}$) are created in the study area S. For each $P_{random_{1...m}}$, new circles are enumerated and the maximum $Log\ LR_C$ of each $P_{random_{1...m}}$ is stored in decreasing order in a list, namely $Log\ LR_C^{MCS}$.*

Hypothesis Test: In GRHD, the null hypothesis (H_0) states that the points are distributed randomly according to a homogeneous Poisson process over the study area S. The alternative hypothesis (H_1) states that the inside of a circle C has a higher number of points than outside [178]. Using the test statistic ($Log\ LR_C$) of a circle C and the distribution of the test statistic (acquired by MCS), the statistical significance of C is determined. The statistical significance (p-value) of a circle C is computed by finding the position (order) of its $Log\ LR_C$ in the distribution of the test statistic ($Log\ LR_C^{MCS}$) and dividing that position by $m + 1$. Given a desired significance level (α_p), if $p\text{-}value \leq \alpha_p$, then H_1 can not be rejected.

4.2.2 Problem Formulation

The Geographically Robust Hotspot Detection (GRHD) problem is formulated as follows:

Given:

1. A set of points P where each $p \in P$ has x and y coordinate in a two dimensional Euclidean space,

2. A minimum radius r_{min}

3. A log likelihood ratio threshold (θ),

4. A *p-value* threshold(α_p) and a number of Monte Carlo simulation trials (m)

Find: Circular Hotspots ($C(x, y, r)$) in the study area S with $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$.

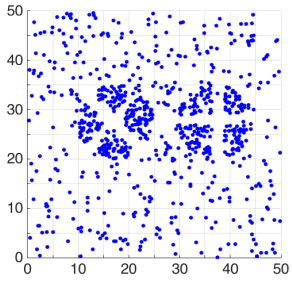
Objective: Computational efficiency and scalability

Constraints:

1. Correctness of the result set,

2. Detected circular hotspots do not overlap

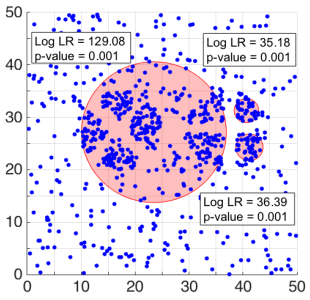
The minimum radius r_{min} input is domain specific and is intended to eliminate very small hotspots (several disease cases in a very small area -a house- may not be interesting). θ indicates the minimum desired $Log\ LR_C$ for a circle C. and α_p is the desired level of statistical significance for a circle C. Depending on the domain, a good practice is selecting the α_p as either 0.01 or 0.001. m indicates the number of Monte Carlo simulation trials and should be selected compatible to the desired level of statistical significance (α_p) [193]. The output of GRHD is non-overlapping circular hotspots with $r \geq r_{min}$ meeting the desired significance and log likelihood ratio levels. Non-overlapping constraint allows to get a single hotspot instead of multiple hotspots given a subset of P [10].



(a) Input

Step 1				Step 2			Step 3
Center x	Center y	r	area(C)	B	c	Log LR	p-value
23.48	27.16	13.42	566.1	181.15	389	129.08	0.001
23.52	27.74	13.42	565.9	181.09	388	128.01	0.001
23.34	27.05	13.28	553.76	177.2	382	126.75	0.001
23.34	27.84	13.52	574.5	183.84	390	126.3	0.001
41.18	24.27	2.54	20.21	6.47	38	36.4	0.001
41.05	24.05	2.42	18.37	5.88	36	35.69	0.001
40.63	31.26	2.29	16.47	5.27	34	35.18	0.001
40.78	31.13	2.24	15.72	5.03	32	32.7	0.001
44.14	43.94	0.99	3.09	0.99	7	7.71	0.76
43.47	43.86	1	3.13	1	6	5.76	0.89

(b) Execution Trace



(c) Output

Figure 4.4: Execution trace of SaTScan algorithm with points as centers (in color). Red rows correspond the red circles in Figure 4.4(c) (best in color).

Example: Suppose, given the point set in Figure 4.3(a), the aim is to find hotspots with a minimum radius of $r_{min} = 2.5$, minimum log likelihood ratio $\theta = 100$ and minimum level of significance $\alpha_p = 0.001$, even if a geographic barrier (river, road network, etc.) divides hotspots. Figure 4.3(d) shows the output of GRHD with two hotspots with log likelihood ratio 155.17 and 114.27 respectively. These hotspots have p-value= 0.001 indicating statistical significance at 99.9% confidence.

4.3 CGC Algorithm

In this paper, SaTScan serves as the baseline algorithm. First, SaTScan is reviewed in detail, an illustrative execution trace is provided and its limitations are summarized. Then a cubic grid circle algorithm (CGC) which is geographically robust (is not affected by gaps or discontiguities) is introduced.

4.3.1 Baseline Approach: SaTScan Algorithm

In order to review SaTScan in detail, it is implemented according to its user guide in [10]. The pseudocode of SaTScan, as shown in Algorithm 4, has three steps:

Step 1-Enumeration of Circles: The algorithm takes each $p \in P$ and makes it the center of a circle $C(x, y, r)$. Next, the radius r of $C(x, y, r)$ is determined by the distance of the rest of the points to its center. This step is done for all possible pairs of points in P (line 1-4).

Step 2-Log Likelihood Ratio Test: For each circle C, $Log\ LR_C$ is computed by Equation 4.1 using $area_C$, $area_S$, the count of the number of points inside (c), and the total number of points in the point set ($|P|$). Among the overlapping circles generated, only the ones which have the highest log likelihood ratio are stored as the $candidateCircles$ (line 5-8).

Step 3-Monte Carlo Simulation & Hypothesis Test: For the circles $C \in candidateCircles$, a *p-value* is computed by Monte Carlo simulation. First, m random datasets with Poisson distribution are generated. For each random dataset, new circles are enumerated and the maximum $Log\ LR_C$ of each random dataset is stored in $Log\ LR_C^{MCS}$ in decreasing order. To find the significance of a C, the position of the $Log\ LR_C$ associated with C is determined within the $Log\ LR_C^{MCS}$ list. This position

is divided by $(m+1)$ to determine the *p-value* (line 9-12). Finally, all non-overlapping circles with *p-value* $\leq \alpha_p$ are returned by the algorithm (line 13).

Algorithm 4 SaTScan Algorithm

Input:

1) A point set P with points $p(x, y)$,

2) A *p-value* threshold α_p, and A number of Monte Carlo Simulation trials (m)

Output:

Non-overlapping circular hotspots C with *p-value* $\leq \alpha_p$.

Algorithm:

Step 1: Enumeration of Circles

for each point $p_i \in P$ **do**

$C \leftarrow$ coordinates of p_i as center x, y

for each point $p_j \in \{P - p_i\}$ **do**

$C \leftarrow$ distance d_{p_i,p_j} as r

Step 2: Log Likelihood Ratio Test

for each C created in Step 1 **do**

Compute B, c, and $area_C$

$Log\ LR_C \leftarrow$ Log Likelihood Ratio using Equation 4.1

Add C and $Log\ LR$ to $candidateCircles$

Step 3: Monte Carlo simulation

for each Monte Carlo simulation $trial_1 ... trial_m$ **do**

Create P_{rand} and determine the $max(Log\ LR_C^{trial})$

Insert $max(Log\ LR_C^{trial})$ into the ordered $Log\ LR_C^{MCS}$ list

Determine the p-value of the circles

Return $candidateCircles$ with *p-value* $\leq \alpha_p$ as $significantCircles$

SaTScan Execution Trace: Figure 4.4 shows a sample execution trace of SaTScan. There are $|P| = 800$ points in the set P and S is $50 \times 50 = 2500$. The thresholds are set to $\theta = 100$ and $\alpha_p = 0.001$. Figure 4.4(a) shows the input dataset.

Step 1 generates all possible circles by assigning each $p \in P$ to be the center of a circle and then calculating the circle's radius. For illustration purposes, only 10 out of $800 \times (800 - 1) = 639200$ circles are shown in Figure 4.4(b).

Step 2 computes the $Log\ LR_C$ of each generated circle. First, the count (c) of the points inside a circle C is determined by computing the distance of every point to the circle center (a point whose distance is less than or equal to the radius r of a circle is determined to be inside that circle). For each enumerated circle, expected number of points B is computed using $area_C$, $|P|$, and $area_S$. Then, $Log\ LR_C$ of each circle is computed using B, c, and $|P|$. These values are shown in Figure 4.4(b). The circles in the white rows are overlapped with the circles with higher $Log\ LR_C$. Thus they are removed from the $candidateCircles$.

Step 3 determines the statistical significance of candidate circles using Monte Carlo simulation. The p-values of 10 circles are shown in Figure 4.4(b). Red rows in Figure 4.4(b) correspond to the output (i.e. red circles) in Figure 4.4(c).

As demonstrated, SaTScan relies on points as the centers which causes it to miss hotspots with sparse centers. In Figure 4.4(c), two hotspots are divided into multiple portions. Although these can be spotted visually, the output of SaTScan does not align with them and not satisfy $\theta = 100$ threshold (although shown in Figure 4.4(c)). In addition, SaTScan does not have a minimum radius. This may cause small hotspots (if two points have the same location, detected hotspot will have $area_C = 0$ causing $LR_C = \infty$) in the output that may occurred by chance. In addition, due to the costly circle enumeration and $Log\ LR_C$ computation, execution time of SaTScan becomes exorbitant for large datasets.

4.3.2 Proposed Approach

In this paper, a cubic grid circle algorithm (CGC) is proposed in order to address the following issues: (1) eliminate chance patterns, (2) detect non-contiguous (divided by rivers, road segments, etc.) or sparse center hotspots, (3) eliminate very small hotspots, (4) improve the scalability that is affected by the cardinality of P.

CGC consists of three phases. Filter phase enumerates hotspots in a parametric space, filters those which do not survive an upper bound on likelihood ratio ($Log\ \widehat{LR_{grid}}$) and return $filteredSets \in P$. Refine phase enumerates actual hotspots using $filteredSets$ and returns a hotspot with the highest $Log\ LR_C$ for each $filteredSet$. Finally, Monte Carlo simulation phase assesses the statistical significance of the enumerated hotspots.

It should be noted that although the problem formulation states a non-overlapping constraint and refine phase returns only one hotspot for each $filteredSet$, the refine phase can be tweaked to return all generated hotspots.

Basic Concepts:

Definition 19 ***A count grid*** *with cell length l_{cell} is a partitioning of the study area S into a 2-dimensional grid where each cell is a square with an area of $l_{cell} \times l_{cell}$. The number of count grid cells is denoted by $N \times N$, where $N = side\_length(S)/l_{cell}$. Each count grid cell ($cell_{count}$) is defined by its coordinate intervals $([x_{min}, x_{max}], [y_{min}, y_{max}])$ and the count (c_{cell}) of the points inside.*

Figure 4.3(a) shows a point set (P) with 800 points in $S = 50 \times 50$ units. Suppose $l_{cell} = 5$, then $N = 50/5 = 10$ and the total number of the count grid cells is $10 \times 10 = 100$. Figure 4.5(a) shows 8 $cell_{count}$ with their point counts c_{cell}. For example, $cell_{count} = ([16, 20], [16, 20]))$ has $c_{cell} = 14$ as shown in the fifth row.

x_{min}	x_{max}	y_{min}	y_{max}	c
1	5	1	5	6
6	10	6	10	5
11	15	11	15	5
16	20	6	10	5
16	20	16	20	14
16	20	26	30	37
31	35	21	25	30
46	50	46	50	7

(a) Count Grid

x_{min}	x_{max}	y_{min}	y_{max}	r_{min}	r_{max}	$\widehat{LR}_{Grid}$
16	20	26	30	6	10	760.43
36	40	26	30	6	10	665.93
21	25	26	30	11	15	635.79
21	25	21	25	11	15	614.55
...	...	...	...	...	...	...
46	50	1	5	11	15	12.42
46	50	46	50	11	15	12.42
6	10	1	5	21	25	10.26

(b) Cubic Circle Grid

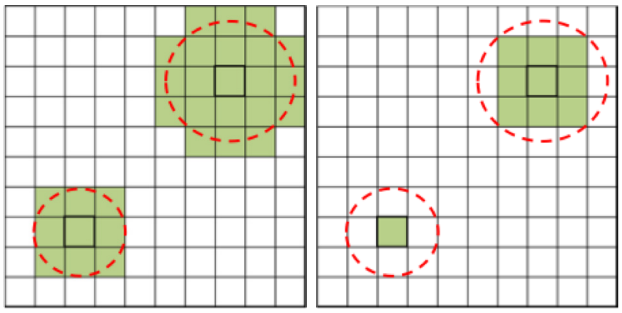
Figure 4.5: Count and Circle Grid cells for $l_{cell} = 5$ (best in color).

Definition 20 ***A cubic circle grid*** *is a three dimensional grid which represents sets of circles in parametric space that are defined with two dimensional center coordinate intervals $([x_{min}, x_{max}], [y_{min}, y_{max}])$ and a radius interval $[r_{min}, r_{max}]$. Given an $N \times N$ count grid, a cubic circle grid is a $N \times N \times N$ grid which parameterize the space into cells with $([x_{min}, x_{max}], [y_{min}, y_{max}], [r_{min}, r_{max}])$. Cubic circle grid cells are denoted by $cell_{circle}$. A $cell_{circle}$ represents a collection of circles with $C(x, y, r)$ where $(x_{min} \leq x \leq x_{max})$, $(y_{min} \leq y \leq y_{max})$ and $(r_{min} \leq r \leq r_{max})$.*

Figure 4.5(b) lists several cubic circle grid cells for the dataset in Figure 4.3(a) when $l_{cell} = 5$. For example, the top row shows a $cell_{circle} = ([16, 20], [26, 30], [6, 10])$. Count grid and cubic circle grid can be matched by the first two columns.

Definition 21 *Given a count grid, a* ***Minimum enclosing cell collection (MECC)*** *is a collection of* $cell_{count}$ *which encloses a set of circles with radius* $r \leq r_{MECC}$ *where* r_{MECC} *is the radius of the MECC. The green cells in Figure 4.6(a) represent an MECC for any actual circle with* r *less than that of* C *shown in dashed red lines.*

Definition 22 *Given a count grid, a* ***Maximum fit cell collection (MFCC)*** *is a collection of* $cell_{count}$ *which can fit completely inside a circle with radius* r *where* $r_{MFCC} = \lfloor \frac{r}{l_{cell}} \rfloor$ *and thus* $r_{MFCC} \leq r$. *The green cells in Figure 4.6(b) represent an MFCC for a* C *shown in dashed red lines.*



(a) MECC (b) MFCC

Figure 4.6: Illustration of minimum enclosing (MECC) and maximum fit cell collections (MFCC) shown in green which represent a circle (in red)

Definition 23 ***The grid upper bound likelihood ratio*** *(Log* $\widehat{LR_{grid}}$*) is an upper bound of the log likelihood ratio of the collection of circles which are defined by the cubic circle grid cells. Given a* $cell_{circle}$ *with* $([x_{min}, x_{max}], [y_{min}, y_{max}], [r_{min}, r_{max}])$*, Log* $\widehat{LR_{grid}}$ *equation is [141]:*

$$
\begin{aligned}
&Log\ \widehat{LR_{grid}} = Log\left(\widehat{LR_{int}} \times \widehat{LR_{ext}} \times \widehat{I()}\right),\ where \\
&\widehat{LR_{int}} = \left(\frac{U(c)}{L(B)}\right)^{U(c)},\ and \\
&\widehat{LR_{ext}} = \begin{cases} \left(\frac{|P|-L(c)}{|P|-U(B)}\right)^{(|P|-U(c))}, & if\ L(c) \geq U(B) \\ 1, otherwise \end{cases} \\
&\widehat{I()} = \begin{cases} 1,\ if\ U(c) > L(B) \\ 0, otherwise \end{cases}
\end{aligned}
\tag{4.2}
$$

$U(c)$ is an upperbound of c, $L(c)$ is a lowerbound of c, $U(B)$ is an upperbound of B and $L(B)$ is a lowerbound of B.
$U(c)$ = Number of points in MECC and $L(c)$ = Number of points in MFCC
$U(B) = \frac{area(MECC)\times|P|}{area(S)}$ *and* $L(B) = \frac{area(MFCC)\times|P|}{area(S)}$

Note that the grid upper bound log likelihood ratio can be applied when $MFCC$ has at least one count grid cell and $MECC$ has less than $N \times N$ count grid cells.

$\widehat{LR_{int}}$ and $\widehat{LR_{ext}}$ are the upper bounds of the two parts of the multiplication in the Equation 3.1, representing the interior and the exterior of the circle, respectively. In order to make the grid upper bound log likelihood ratio greater than the actual log likelihood ratio of the hotspots represented by $cell_{circle}$, $\widehat{LR_{int}}$ and $\widehat{LR_{ext}}$ are defined. $\widehat{LR_{int}}$ is always greater than $(\frac{c}{B})^c$ in Equation 3.1. $\widehat{LR_{ext}}$ is conditioned on whether $L(c) \geq U(B)$ or not. The indicator function $\widehat{I()}$ is similar to the indicator function in Equation 4.1 and is set to $\widehat{I()} = 1$ when the $U(c)$ is higher than the $L(B)$ under the hypothesis H_0 [141]. Finally, it is worth mentioning that the cell length l_{cell} makes the upperbound tight/loose and $\lim_{l_{cell}\to 0}(Log\ \widehat{LR_{grid}} = Log\ LR_C)$.

For example, in the second row of the table in Figure 4.5(b), $Log\ \widehat{LR_{grid}}$ is computed by the following; $U(B) = 72$, $L(B) = 8$, $L(c) = 15$ and $U(c) = 205$. $Log\ \widehat{LR_{grid}} = Log\left(\left(\frac{205}{8}\right)^{205} \times \left(\frac{800-15}{800-72}\right)^{800-205} \times 1\right) = 665.93$.

Cubic Grid Circle Algorithm (CGC): Algorithm 5 shows the three phases of CGC. Next, these phases will be explained in detail.

Algorithm 5 Cubic Grid Circle Algorithm

Input:

1) An point set P with points $p(x, y)$,

2) A minimum circle radius r_{min} 3) A log likelihood ratio threshold (θ),

4) A *p-value* threshold α_p and 5) A number of trials in Monte Carlo Simulation (m)

Output:

Circles C with $r \geq r_{min}$, $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$

Algorithm:

Filter Phase: $filteredSet \leftarrow$ *all* $p \in MECC$ for a $cell_{circle}$ with $Log\ \widehat{LR_{grid}} \geq \theta$

Refine Phase: $candidateCircles \leftarrow C$ with $Log\ LR_C \geq \theta$.

Monte Carlo Sim. Phase: $resultSet \leftarrow candidateCircles$ with $p\text{-}value \leq \alpha_p$

Return $resultSet$

Filter Phase: The pseudocode for the filter phase is given in algorithm 6. First, S is discretized into count grid and cubic circle grid using $l_{cell} = \frac{r_{min}}{2}$ (lines 1-2). The cell length l_{cell} is selected by the given input r_{min}, because in order to detect all the hotspots with $r \geq r_{min}$ at least one cell should fit completely inside the hotspot. Next, the number of expected points inside a single cell is determined by $B_{cell} \leftarrow |P|/(N \times N)$ (line 3). B_{cell} is used when computing the $U(B)$ and $L(B)$, since MECC and MFCC are constituted of cells. For every $cell_{circle}$, $Log\ \widehat{LR_{grid}}$ are computed (lines 5-7). To prevent the overlapping $filteredSet$s, once $Log\ \widehat{LR_{grid}}$s are computed, the $cell_{circle}$ with the highest $Log\ \widehat{LR_{grid}}$ is stored in $cell^{top}_{circle}$ if $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ (line 8) and points, which are associated with $cell^{top}_{circle}$, are stored as a $filteredSet$ (line 9-10) and are removed from P. This process is repeated until none of $Log\ \widehat{LR_{cell}} \geq \theta$ or $P = \emptyset$ (lines 4-11). Finally, $filteredSet$s and $cell_{count}$s are sent to the refine phase.

Algorithm 6 Filter Phase Algorithm of CGC

Input:

1) A log likelihood ratio threshold (θ)

2) $l_{cell} = \frac{r_{min}}{2}$ for the cell size of count grid and circle grid

Output:

All $filteredSet$s for each non-overlapping $cell_{circle}$ with $Log\ \widehat{LR_{grid}} \geq \theta$

Algorithm:

Create count grid with $N \times N$ cells using l_{cell}

Create cubic circle grid with $N \times N \times N$ cells

Compute the expectation for a single $cell_{count}$ by $B_{cell} \leftarrow |P|/(N \times N)$

while $P \neq \emptyset$ || $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **do**

  for each $cell_{count}$ in count grid **do**

    for each $r \leftarrow 1$ *to* $N/2$ **do**

      $cell_{count} \leftarrow Log\ \widehat{LR_{grid}}$

  $cell_{count}^{top} \leftarrow cell_{count}$ with the highest $Log\ \widehat{LR_{grid}}$

  if $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **then**

    $filteredSet_{1...k} \leftarrow$ *all* $p \in MECC$ of the $cell_{count}^{top}$

    $P \leftarrow [P - filteredSet]$

Return $cell_{count}^{1...k}$ and $filteredSet_{1...k}$

Lemma 10 *Filter phase of the CGC Algorithm can detect a circular hotspot C with $r \geq r_{min}$, if $l_{cell} \leq (r_{min}/2)$ and thus a count grid cell is completely inside C.*

Proof 10 *In the Equation 2, $L(B)$ and $L(c)$ are defined by the cells in MFCC. If at least one $cell_{count}$ is inside a hotspot, then $Log\ \widehat{LR_{grid}} > Log\ LR_C$. The proof lies in $U(B)$, $U(c)$, $L(B)$ and $L(c)$. Suppose a $cell_{count}$ is completely inside a hotspot. Then $U(B) \geq B$, since the area of MECC will be larger than $area_C$. Also, $L(B) \leq B$, since MFCC will be inside the circle and thus its area will be smaller than that of $area_{circle}$. In the worst case, suppose $U(c) = c$ and $L(c) = c$, then $Log\ \widehat{LR_{grid}}$ will still be $\geq Log\ LR_C$ since $L(B) \leq B$ and $U(B) \geq B$. Therefore, if $l_{cell} \leq (r_{min}/2)$, then $Log\ \widehat{LR_{grid}} \geq Log\ LR_C$.*

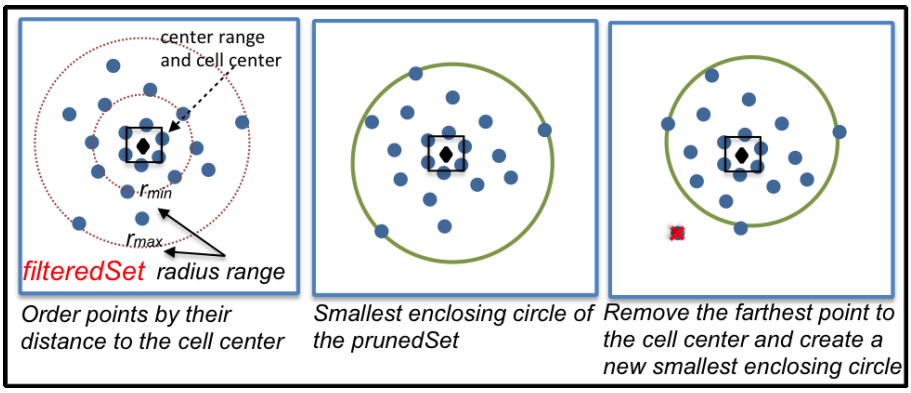


Figure 4.7: Illustration of the refine phase in CGC algorithm

Refine Phase: The pseudocode of the refine phase in algorithm 7 starts by ordering the points in the $filteredSet$ by their distance to the center of the $cell_{count}$ (line 2). Next, the minimum enclosing circle (C) of the $filteredSet$ is determined [194](line 4). If the center of C is inside the center interval of $cell_{count}$, the $Log\ LR_C$ is computed using the count of the points inside (which is equal to the cardinality of $filteredSet$) and the area of C (line 5-7). If the $Log\ LR_C$ is greater than the one computed previously as $Log\ LR_C^{previous}$, it is saved (line 8-9). Next, the point farthest from the center of the $cell_{count}$ is removed from the $filteredSet$ (line 10) and the algorithm repeats the process of finding a new minimum enclosing circle for the new $[filteredSet - p_{farthest}]$ until $|filteredSet| = \emptyset$ or the radius of the generated C is $r \leq r_{min}$ (line 3-10). Finally, the C with the highest $Log\ LR_C$ is saved in $candidateCircles$ (line 11). This process is done for all $filteredSet$s returned by the filter phase (line 1-12).

An execution trace of the refine phase can be seen in Figure 4.7. Given the $filteredSet$ and $cell_{circle}$, the points are ordered by their distance and their smallest enclosing circle is determined (second box). Once the $Log\ LR_C$ of this circle (shown in green) is computed, the point farthest from the center of $cell_{count}$ is removed from the set (shown in red) and a new smallest enclosing circle is created using the rest of the points. This process is repeated until the $filteredSet = \emptyset$.

Algorithm 7 Refine Algorithm of CGC

Input:

1) A log likelihood ratio threshold (θ)

2) $filteredSet_{1...k}$ and $cell^{1...k}_{count}$ returned by the filter phase

Output:

A circular hotspot C for each $filteredSet$ if $Log\ LR_C \geq \theta$

Algorithm:

for each $filteredSet_i \in filteredSet_{1...k}$ **do**

order $p \in filteredSet_i$ by distance to the center of $cell^i_{count}$

while $filteredSet_i \neq \emptyset$ or $r_i \geq r_{min}$ **do**

$C_i \leftarrow SEC(filteredSet_i, tmpSet)$ where $tmpSet = \emptyset$

if $center^i_C$ is inside the center interval of $cell^i_{count}$ and $r_i \geq r_{min}$ **then**

compute B and c

$Log\ LR^i_C \leftarrow$ Log Likelihood Ratio

if $Log\ LR^i_C \geq Log\ LR^{previous}_C$ **then**

$Log\ LR^{previous}_C = Log\ LR^i_C$ and $C_{previous} = C_i$

$filteredSet_i = filteredSet_i - p_{farthest}$

$candidateCircles \leftarrow C_{previous}$ and $Log\ LR_{previous}$

Return $candidateCircles$

procedure SEC($filteredSet, tmpSet$) [194]

if $filteredSet == \emptyset$ || $|tmpSet| == 3$ **then**

$C \leftarrow computeCircle(tmpSet)$

else

select a random $p \in filteredSet$

$C \leftarrow SEC(filteredSet - p, tmpSet)$

if p is not in C **then**

$C \leftarrow SEC(filteredSet - p, tmpSet \cup p)$

end procedure

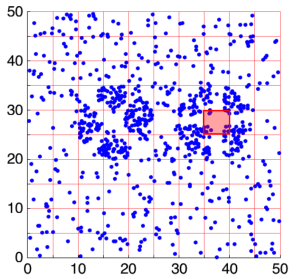
Monte Carlo Simulation Phase: During Monte Carlo simulation, the filter phase and refine phase of CGC algorithm is run for each individual random datasets created

for the Monte Carlo simulation and the highest $Log\ LR_C$ are stored and the circles' p-values are determined using the ordered list of these highest $Log\ LR_C$.

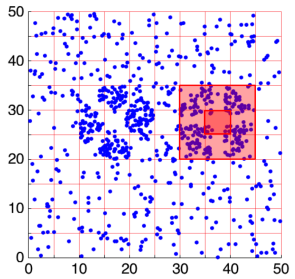
CGC Execution Trace: In Figure 4.8, the execution trace of CGC is illustrated using the point set in Figure 4.3(a) with $|P| = 800$ points in an area $S = 50 \times 50 = 2500$. The log likelihood ratio threshold $\theta = 100$ and the *p-value* threshold $\alpha_p = 0.001$. In order to make the execution trace be easy to follow, the cell length is selected $l_{cell} = 5$ for this example.

In the filter phase, the study area is partitioned into a count grid and a cubic circle grid. For example, the $cell_{count}$ $([36, 40], [36, 40])$ has $c_{cell} = 8$ (shown in red in Figure 4.8(a). Suppose a $cell_{circle}$ is created for coordinates $([36, 40], [36, 40], [6, 10])$. The MECC for this cell is shown in Figure 4.8(b) and the MFCC will be the red cell in Figure 4.8(a). Using these, $U(B) = 72$, $L(B) = 8$, $L(c) = 15$ and $U(c) = 205$. and $Log\ \widehat{LR_{grid}} = Log\left(\left(\frac{205}{8}\right)^{205} \times \left(\frac{800-15}{800-72}\right)^{800-205} \times 1\right) = 665.93$. Once all the $\widehat{LR_{grid}}$ are computed, the $cell_{circle}$ with the highest $\widehat{LR_{grid}}$ and the associated points are sent to the refine phase and the filter algorithm repeats for the rest of the points in P.

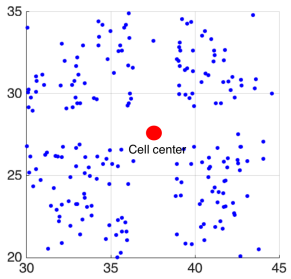
In the refine phase, as shown in Figure 4.8(c), $filteredSet$ is ordered distance wise to the center of the $cell_{circle}$ which was returned by the filter phase. Next, the minimum enclosing circle of the $filteredSet$ is determined and the $Log\ LR_C$ is computed as shown in Figure 4.8(d). Then, the farthest point to the center of $cell_{circle}$ is removed from the $filteredSet$ and a new minimum enclosing circle and its $Log\ LR_C$ is determined. This process is repeated until $filteredSet$ is empty or the minimum enclosing circle has a radius $r \leq r_{min}$. Now a list of $Log\ LR_C$ and circles enumerated using the $filteredSet$ are acquired as shown in Figure 4.8(f). Finally, the circle C with the highest $Log\ LR_C$ is returned as a $candidateCircle$ for Monte Carlo simulation. Monte Carlo simulation then determines the p-value of each candidate circle. This process is similar to the filter and refine phases except that random datasets are used as input. The final output of CGC can be seen in Figure 4.8(g).



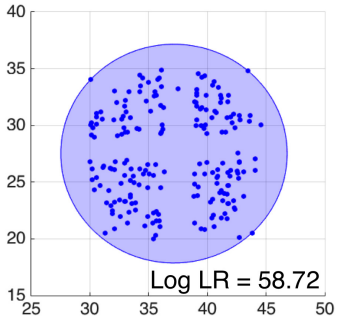
(a) Count grid cells are shown with red lines



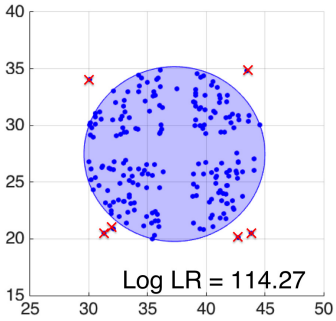
(b) A sample $cell_{circle}$ shown as red cells.



(c) Points in $filteredSet$ are ordered distance-wise to the center of $cell_{circle}$



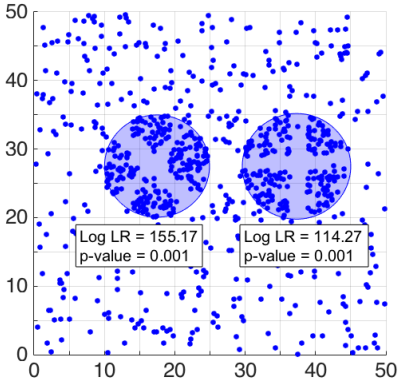
(d) Minimum enclosing circle of the $filteredSet$



(e) Minimum enclosing circle for a new $filteredSet$

Center x	Center y	r	area(C)	B	c	Log LR	p-value
37.16	27.51	9.64	292.21	93.51	205	58.72	0.01
37.54	27.52	9.39	277.17	88.69	204	64.51	0.01
37.54	26.45	8.62	233.47	74.71	203	86.71	0.01
37.22	26.53	8.45	224.26	71.76	202	91.22	0.01
37.62	26.75	8.37	220.11	70.44	201	92.65	0.01
37.19	27.01	7.97	199.55	63.86	200	105.64	0.01
37.3	27.48	7.71	186.62	59.72	199	114.26	0.01
37.32	27.46	7.69	186	59.52	198	113.35	0.01
37.34	27.44	7.68	185.27	59.29	197	112.52	0.01
37.34	27.43	7.68	185.24	59.28	196	111.14	0.01
37.34	27.43	7.68	185.24	59.28	195	109.75	0.01
...	...	...	...	...	...	...	...

(f) Minimum enclosing circles and their $Log\ LR_C$ for the $filteredSet$



(g) Output of CGC

Figure 4.8: Execution trace of the CGC algorithm (best in color).

4.4 Theoretical Evaluation

4.4.1 Correctness of the CGC Algorithm

Lemma 11 *CGC is functionally correct. Functional correctness means that CGC returns only the circular hotspots with $r \geq r_{min}$, Log $LR_C \geq \theta$ and p-value $\leq \alpha_p$.*

Proof 11 *Circles generated by CGC are evaluated against the $r \geq r_{min}$ in Algorithm 7 Line 5 and against the Log $LR_C \geq \theta$ in Algorithm 7 Line 8 and only if they satisfy the thresholds they are saved in $candidateCircles$. In the Algorithm 5 in Monte Carlo simulation phase each $candidateCircle$ is evaluated against the p-value threshold in Line 3 and only the ones which satisfy α_p are returned as the result set. Since CGC returns the circles with $r \geq r_{min}$, Log $LR_C \geq \theta$ and p-value $\leq \alpha_p$, it is functionally correct.*

Lemma 12 *Given a finite point set P, CGC terminates in finite time.*

Proof 12 *In algorithm 6, the iteration in Line 4 is bounded by the Log $\widehat{LR_{grid}}$ and $|P|$. Since in every iteration the associated points with cells which exceed the θ threshold are removed from P in Line 11, the $\lim_{iterations \to \infty} P \to \emptyset$. In the case where none of the cells exceed θ threshold loop stops again in Line 4. Thus algorithm 6 will terminate in a finite time. In the refine phase, on every iteration of the loop in Line 3, the farthest points are removed from the $filteredSet$ in Line 10. Thus the loop will terminate when the $filteredSet = \emptyset$. Since filter and refine phase terminate in finite time, CGC terminates in finite time.*

Theorem 2 *CGC is a correct approach to detect statistically significant circular hotspots. An algorithm is correct if it is functionally correct and it terminates in finite time.*

Proof 13 *Using Lemma 11, CGC is a functionally correct approach. Also using Lemma 12, CGC will terminate in finite time. Thus, CGC is a correct approach to detect statistically significant circular hotspots.*

4.4.2 Computational Analysis of the CGC Algorithm

The complexity of SaTScan is $O(m \times |P|^3)$ ($|P|^2$ to enumerate circles, $|P|$ to count points for each circle), where $|P|$ is the cardinality of the point set and m is the number

of Monte Carlo simulation trials.

In the filter phase of CGC, circles are enumerated using the count grid cells. Since all count grid cells (N^2) are traversed for all possible radii, the total cost of the filter phase is $O(N^3)$ which is equal to the size of the cubic circle grid. Note that, grids are created using l_{cell} which is determined by the minimum radius r_{min} defined in the problem statement. Therefore CGC filter algorithm execution time depends on the careful selection of these inputs.

In the refine phase of CGC, circles are enumerated for each $filteredSet$ using a smallest enclosing circle algorithm of a linear cost $O(|filteredSet|$, then points are removed one by one from the $filteredSet$ and new circles are enumerated until $filteredSet = \emptyset$ or $r \leq r_{min}$. The total cost of removal and recompute is $O(|filteredSet|log\ |filteredSet|)$. In the worst case, $|filteredSet| = |P|$ and the refine phase cost is $|P|log\ |P|$ and in the best case $filteredSet = \emptyset$ and the refine phase cost is $O(1)$.

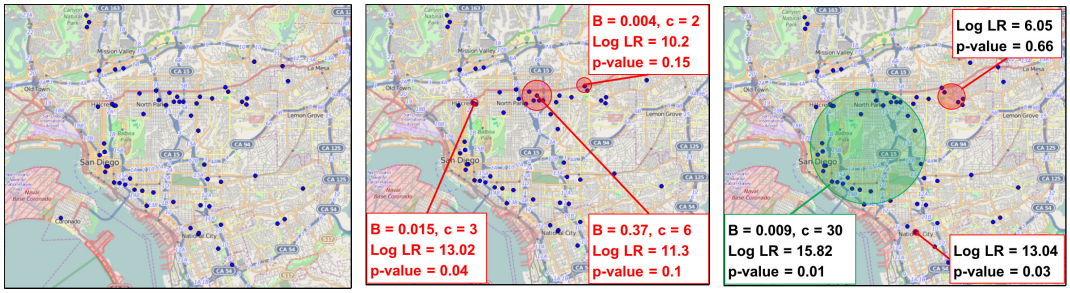
Thus, the worst case cost of CGC is $O(m \times (N^3 + |P|log\ |P|))$, if no filtering occurs and all points are returned to the refine phase. In the best case, the cost will be $\Omega(m \times N^3)$ as the filter phase will not return any $filteredSet$ and the algorithm will terminate. Note that since the datasets in the Monte Carlo simulation are created randomly, depending on the θ, these do not survive filter phase and the algorithm performance gets closer to the best case scenario.

4.5 Case Study

The proposed approach is evaluated qualitatively by comparing the CGC algorithm output with SaTScan using continuous Poisson process [10] on a crime dataset shown in Figure 4.9. The input point set in Figure 4.9(a) includes 64 unarmed robbery cases in San Diego between March 2013 - 2014 [11]. The inputs are selected as $r_{min} = 0.009$ degrees, $\theta = 5$ and $\alpha_p = 0.01$. Maps were prepared using QGIS' OLP [195].

The two algorithms generated quite different results. It appears that SaTScan's reliance on point centered circles caused it to miss a significant hotspot and its output did not satisfy the thresholds (Figure 4.9(b)). Also the output of the SaTScan includes small hotspots consists of two/three points which couldn't be filtered out. CGC algorithm handles this issue by using r_{min} threshold.

GRHD with the CGC algorithm discovered a hotspot (Figure 4.9(c)- shown in green) which satisfies the input thresholds. Although, domain experts may interpret these better, it can be stated that these crimes occurred along a road and this road surrounds a residential area where the crimes are sparse. This type of crime pattern can be seen in environmental criminology [14]. It should be noted that these crimes' sources are anonymized (for privacy issues), meaning that we can't compare the results with ground truth labels.



(a) Input crime dataset (b) Output of SaTScan (c) Sample Output of GRHD

Figure 4.9: Figure 4.9(a) shows 64 unarmed robbery crimes committed in San Diego in 2013 (blue dots) [11]. Figure 4.9(b) and Figure 4.9(c) compare the output of SaTScan and GRHD with the CGC algorithm(red/green circles - best in color).

4.6 Experimental Evaluation

4.6.1 Experimental Design

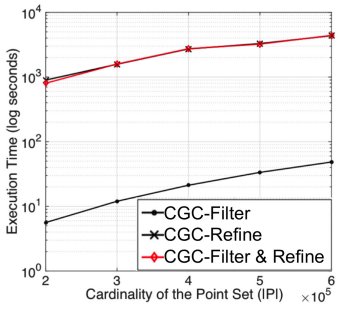
The goal of the experiments was twofold: to evaluate the performance of the CGC algorithm under different parameters and to compare its performance with SaTScan. To achieve these goals, the following questions are asked: (1) How is the scalability and the result quality of the proposed algorithm compared to its rivals (e.g. SaTScan)? (2) How effective is the filter step in reducing the cost of CGC algorithm?

Experimental Design: Experiments are performed on synthetic datasets which is created with varying number of points (default 5000) in a 1000×1000 study area. In these datasets 20% of the points were generated to form a hotspot and the rest of the points were created using complete spatial randomness (CSR). Inputs were; log

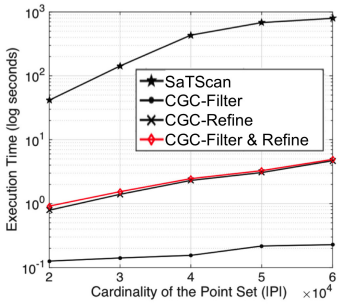
likelihood ratio threshold $\theta = 1000$, p-value threshold $\alpha_p = 0.01$ (99% confidence level) and minimum circle radius $r_{min} = 100$. Performance of both algorithms was measured in terms of CPU time. All experiments were performed on a MacBook Pro with a Intel Core i7 2.2 GHz CPU and 4GB memory. To compare CGC and SaTScan algorithms, both algorithms were implemented on Java platform and were executed 10 times for each experiment. When implementing SaTScan, continuous Poisson process is selected as defined in SaTScan user guide [10].

4.6.2 Experimental Results

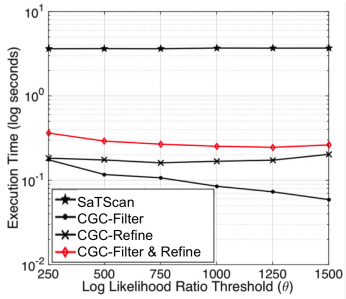
Effect of the Number of Points in P: In this experiment, two sets of synthetic points sets were created. The first set of point sets included points with cardinality ranging from 20K to 60K which used to compare SaTScan with CGC. With the second set of point sets, the scalability of CGC is tested with its Filter and Refine phases and does not include SaTScan. For the point sets ranging from 20K, 30K, 40K, 50K, 60K, θ is selected 10^4. For the point sets ranging from 200K, 300K, 400K, 500K, 600K, θ is selected 10^6. Figure 4.10(a) shows that there is at least two orders of magnitude difference between SaTScan and CGC execution times. Also in Figure 4.10(a), it can be seen that CGC filter phase performs faster and most of the execution time is spent on the refine phase. Overall CGC algorithm performs faster than SaTScan and the savings increase when the point set size increases.



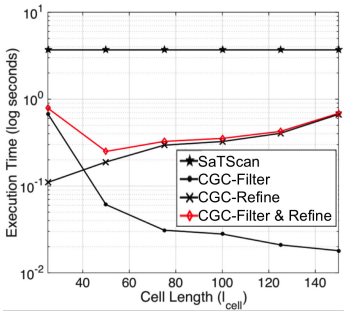
(a) Large Dataset size $|P|$



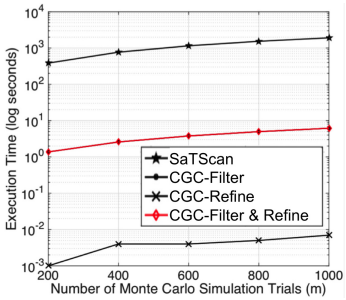
(b) Small Dataset size $|P|$



(c) Log Likelihood Ratio Threshold θ



(d) Cell Length for Filter Phase l_{cell}



(e) Number of Monte Carlo simulation trials m

Figure 4.10: Scalability of CGC with increasing (a), (b) number of points, (c) log likelihood ratio threshold, (d) cell length and (e) number of Monte Carlo simulation trials.

Effect of the Log Likelihood Ratio Threshold (θ): In this experiment, the log likelihood ratio threshold (θ) is varied by $250, 500, 750, 1000, 1250$ and 1500. In Figure 4.10(c), there is no change on the execution times of SaTScan, since SaTScan does not use any filter depending on θ. However CGC filter phase benefits from the θ and savings increase with θ .

Effect of the Cell Length (l_{cell}): In order to observe the effect of the cell length l_{cell}, cell length is changed by $25, 50, 75, 100, 125, 150$. In Figure 4.10(d), since SaTScan does not use any filtering method based on grid generation, cell length did not affect SaTScan. For CGC algorithm, when the cell length is small, filter phase takes more time and refine phase takes less time since the number of points in the $filteredSet$ is close to the actual points in the hotspot. On the other hand, when the cell length is

large, filter phase takes less time but this time refine phase takes more time since the $filteredSet$ includes more points than the actual hotspot. It can be concluded that a careful selection of cell length is important for the performance of the CGC algorithm.

Effect of the Number of Monte Carlo simulation trials (m): In this experiment, $m = 200, 400, 600, 800, 1000$ random synthetic datasets with 5000 points in a 1000×1000 study area are generated. Other inputs were kept the same per experimental design. Figure 4.10(e) shows the execution times for the Monte Carlo simulation trials. As can be seen CGC performed at least two orders of magnitude faster than SaTScan and in many trials CGC algorithm did not need to run the refine phase since most of the random datasets did not have any circular hotspot which exceed the specified $\theta = 100$ and therefore filter phase did not generate any $filteredSet$. Therefore it can be concluded that the grid circle upper bound log likelihood ratio filtering in the proposed CGC algorithm improves its scalability. In summary, the experiments show that CGC is more scalable than SaTScan for large point sets.

4.7 Discussion

It is worth mentioning that there are other techniques for the detection of statistically significant hotspots which use predefined grids (e.g. overlap-multires partitioning [175], predefined locations -counties/zip codes and aggregated number of points- [192] approximation approaches [174]). These techniques are fast when the cardinality of the input point set is higher than the number of grids. However, they are designed for rectangular/square hotspot detection. Moreover, these approaches use grids as the shape of the hotspot and their output is a "single" hotspot with the highest test statistic in the study area. Therefore, these techniques were not considered in this paper. However, it should be noted that CGC filter approach is not limited to circles and any shape that can be defined by parameters can be evaluated with this approach. For example, proposed CGC filter phase uses a cubic circle grid to enumerate circles using three parameters that define a circle, namely center coordinates and radius. Similarly, CGC approach may be generalized to be used with other shape-specific (rectangle, square, circle, ellipse, etc.) state-of-the-art methods to filter the candidate hotspots that do not contribute to an actual hotspot.

4.8 Conclusions and Future Work

This work explored the Geographically Robust Hotspot Detection (GRHD) problem, which is important for societal applications e.g. epidemiology, environmental criminology, etc. GRHD is challenging due to the difficulty of enumerating all possible hotspots and the high computational cost of the statistical significance test. This paper proposed a CGC algorithm which discovers statistically significant hotspots which couldn't be discovered previously by SaTScan. A case study demonstrated CGC's superior performance over SaTScan on a real crime dataset. Experiments also showed that the proposed algorithm (CGC) is highly scalable.

In the future, we envision to improve geographical robustness by adding enhancements to detect imperfect hotspots (i.e., half/quarter circles) where the points occur along the coast lines, jurisdiction boundaries, roads etc. which may cause these hotspots to be missed. In addition, we plan to add the effect of unhomogeneous point distributions caused by the characteristics of the geographic location (population, etc.). Also we plan compare GRHD with other state-of-the-art methods [174, 191, 192, 196, 197] that are designed to detect rectangular, elliptical and irregular shaped hotspots. Finally, we plan to add expectation based Poisson statistics to improve the result quality and scalability of our proposed approach.

Chapter 5

Mining Network Hotspots with Holes

5.1 Introduction

Given a spatial network and a collection of activities (i.e. crime locations), the problem of Mining Network Hotspots with Holes (MNHH) finds hotspots with doughnut shaped spatial footprint on a spatial network (i.e. road network), where the concentration of activities is unusually high (i.e. statistically significant).

The problem of Mining Network Hotspots with Holes (MNHH) has important societal applications in criminology, where identifying crime hotspots may improve police response [198]. In environmental criminology, domain experts create geographic profiles of criminals using the locations of crimes and try to find where a serial criminal frequently commutes, thereby focusing the efforts of police forces in the field [14]. Our notion of Network Hotspots with Holes originates from two key concepts in criminology, namely inner buffer zone (e.g. comfort zone) and distance decay [14]. Inner buffer zone is an area around a criminal's frequently visited locations, where crimes are less likely due to the risks caused by reduced anonymity. Distance decay relates to a least effort principle, where crimes occur relatively close to criminal's frequently visited locations, since traveling long distances requires time and money. The opposing effects of inner buffer zone and distance decay create an activity zone with a doughnut shaped spatial footprint around a path that a criminal usually travels. Fig.5.1 illustrates these

concepts where the green squares represent activities (i.e. crime), the blue line shows a path between home and work (blue squares), the black road segments represent the inner buffer zone where the activities are less likely and the red road segments create the activity zone (i.e. outer buffer) that we define as network hotspot with hole (*NHH*) in this paper.

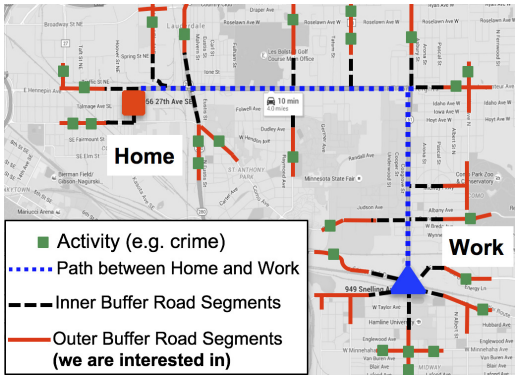
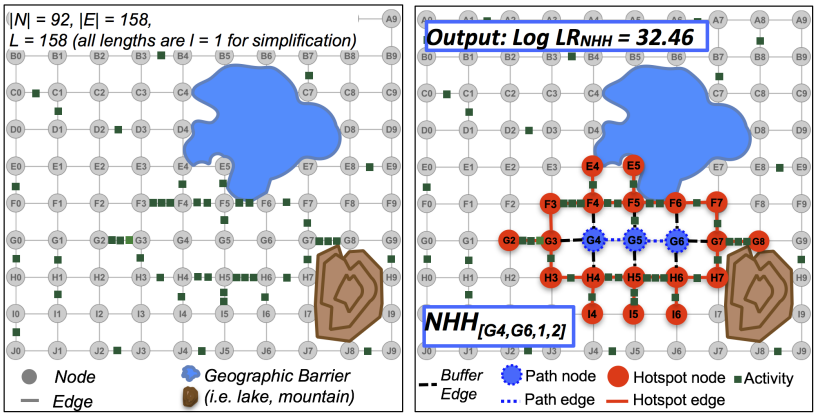


Figure 5.1: A path between home and work, an inner buffer zone where the activities are sparse and an activity zone (i.e. outer buffer) that we are interested in.

Informally, the problem of Mining Network Hotspots with Holes (MNHH) can be defined as follows: given a spatial network (e.g. road network), an activity set associated with road segments (e.g. street robberies), a log likelihood ratio threshold (θ), a *p-value* threshold (α_p), a maximum outer buffer distance ($\widehat{t_{max}}$) and a unit distance (ω), find network hotspots with holes where the concentration of activities is significantly higher than outside (*p-value* $\leq \alpha_p$).

Challenges: MNHH is challenging due to the potentially large number of candidate network hotspots with holes ($O(N^4)$) in a given dataset of millions of road network nodes (N). For large road networks (e.g. 10^8 road segments in the U.S.), this causes exorbitant computation times as well as a prohibitively large enumeration space. Moreover, the interest measure, "log likelihood ratio (*Log LR*)", does not have a monotonicity property, meaning that there is no order between the *Log LR* of a network hotspot with hole (*NHH*) and another *NHH* it may contain. Thus, interest measure cannot be used for computational speed-up. In addition, statistical significance test multiplies the cost.

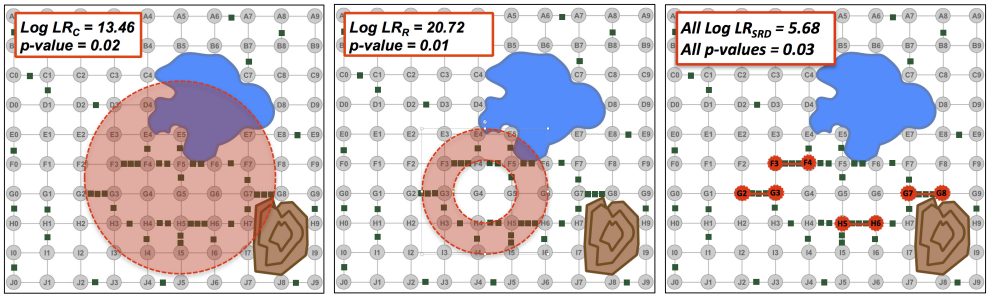
Related Work and their Limitations: Statistically significant hotspot detection approaches can be classified into two categories depending on the study area: Euclidean space based and Network based approaches. Euclidean approaches include spatial scan statistics and are widely used for the detection and evaluation of circular [10, 199], elliptical [200, 201], rectangular [202] and ring-shaped hotspots [141]. These techniques are useful for understanding the distribution of disease [203], or detecting a disease outbreak or even identifying the location of a criminal (e.g. through ring-shaped hotspot detection). However, criminal activities and other human activities diffuse along road networks [204] and therefore Euclidean distances do not reflect actual travel distances causing biased results. For example in Fig.5.2(a), the traveling distance from $E5$ node to $C7$ node will not be the same as Euclidean distance due to the lake in between. In addition, people's activities are mostly dependent on their routine commutes (i.e. home-work-recreation) instead of a single place. In Fig.5.3(a), SaTScan [10] outputs a circular hotspot with a large space without activities with a low log likelihood ratio. Similarly, in Fig.5.3(b), ring-shaped hotspot detection (RHD) outputs a hotspot with low log likelihood ratio due to using Euclidean distance as well as assuming a single center (i.e. crime base of a criminal). Thus, geometry-based techniques may not be appropriate for modeling hotspots on road networks. A more detailed comparison of the recent related work can be found in [203, 205].



(a) Input (b) Output

Figure 5.2: An example input and output of our proposed approach for Mining Network Hotspots with Holes. Edges represent streets and Nodes represent road intersections.

A second category of hotspot detection is network-based. These methods leverage the underlying spatial network, which improves the detection of activities that diffuse along the spatial network [206, 207, 208]. However, these often focus on detecting paths or road segments which have unusually high activities and require a hotspot to be a connected subgraph (e.g. shortest path), causing them to miss network hotspots with holes [12, 209, 210]. Fig.5.3(c) shows the output of a significant route discovery approach which enumerates shortest paths between nodes and returns those that have a significantly high number of activities [12]. The output fails to identify the significant region of interest and includes 4 hotspots with low log likelihood ratios and high *p-value*s, indicating lack of significance.



(a) (b) (c)

Figure 5.3: Output of the related work for the input in Fig.5.2(a). SaTScan (Fig.5.3(a)), Ring-Shaped Hotspot Detection (Fig.5.3(b)) and Significant Route Discovery (Fig.5.3(c))

In contrast to previous methods, our Mining Network Hotspots with Holes (MNHH) method can find statistically significant network hotspots with holes (e.g. Fig.5.2(b)) without requiring the output to be a connected subgraph.

Contributions: In this paper, we present the problem of Mining Network Hotspots with Holes (MNHH) on a spatial network. To the best of our knowledge, the proposed approach is the first to consider statistically significant hotspots with holes on a spatial network. Specifically, our contributions are as follows:

- We introduce the problem of Mining Network Hotspots with Holes (MNHH) on a spatial network and a Naïve Network Hotspot with Hole Generator (NaïveNHHG)

algorithm to solve MNHH.

- We propose Smart Network Hotspot with Hole Generator (SmartNHHG) algorithm which prevents redundant computations by dynamic programming.
- We present case studies comparing the proposed approach to geometry-based approaches (i.e., SaTScan, ring-shaped hotspot detection) on real crime datasets. Note that the output patterns should not be considered the same (e.g. circles and rings vs. network sub-graphs).
- Experimental results on real data show that SmartNHHG yields substantial computational savings over NaïveNHHG without sacrificing result quality.

Scope: This work focuses on finding hotspots with holes on road networks where each activity (i.e. crime event) is associated with a road segment (i.e. edge). This does not imply that the original activities must necessarily have occurred at edges. Each activity set is pre-processed to associate activities to the closest edge on the road segment. In addition, other properties of road networks (e.g. speed limit, traffic density) are not considered. In this work, the number of activities on the road network is fixed and does not change over time. Finally, this paper does not provide guidance on parameter (e.g. t_{min}, t_{max}) value selection. However, users may evaluate the spatial distribution of events using centrographic statistics [19] and select parameters accordingly.

Outline: This paper is organized as follows: Section 5.2 presents the basic concepts and problem statement for MNHH. Section 5.3 presents the Naïve and Smart Network Hotspot with Hole Generator (SmartNHHG) algorithms. Section 5.4 presents case studies which qualitatively evaluate the output of SmartNHHG on real crime datasets. Experimental evaluation is in Section 5.5. Section 5.6 presents a discussion. Section 5.7 concludes the paper and previews future work.

5.2 Basic Concepts and Problem Statement

5.2.1 Basic Concepts

Definition 24 *A **spatial network** $G = (N, E)$ is a set of nodes (N) and edges (E) where each node $n_v \in N$ is associated with coordinates (x, y) representing its location*

in an Euclidean space. E is a subset of the cross product $N \times N$ and an edge $e_i \in E$, which joins nodes n_u and n_v, is associated with a length $l_{u,v} \geq 0$.

In Fig.5.2(a) grey circles represent nodes (e.g. intersections), grey lines represent edges (e.g. streets) and there are two geographic barriers (e.g. lake, mountain). The length of the network is the sum of all edge lengths $L_{total} = \sum l_{e \in G}$.

Definition 25 *An **activity set** A is a collection of activities. An activity $a \in A$ is an object of interest associated with only one edge $e \in E$.*

For example in Fig.5.2(a), the edge between n_{G2} and n_{G3} has 3 activities.

Definition 26 *A **shortest path** $p_{u,v}$ is a sequence of nodes $[n_1, n_2, ..., n_i]$ such that $[e_1, e_2, ..., e_i] \in E$ and $n_i \in N$ are distinct and the sum of edge lengths is minimized. The length of a shortest path is $L_p = \sum l_{e \in p}$.*

For example, $p_{A0,B2} = [A0, A1, A2, B2]$ and $L_p = 3$ in Fig.5.2(a).

Definition 27 ***Distance** between a node n_i and a path $p_{u,v}$ is $d(n_i, p_{u,v}) = \min(L_p(n_i, n_j \in p_{u,v}))$.*

For example, $d(D3, p_{A0,B2}) = 3$ in Fig.5.2(a).

Definition 28 ***A Network Buffer** ($NB_{u,v,t}$) is a closed set of nodes $N_{NB} \subset N$ and edges $E_{NB} \subset E$ such that $d(n_i, p_{u,v}) \leq t$, $\forall n_i \in N_{NB}$ and $t = k\omega$ for some $k \in \mathbb{R}^+$ and a unit distance ω.*

For example , in Fig.5.2(b), $NB_{G4,G6,2}$ is the set of all blue/red nodes and all blue/red/black edges.

Definition 29 ***A Network Hotspot with Hole** ($NHH_{u,v,t_{min},t_{max}}$) is the closure [211] (a set and its limit points that is denoted by Cl) of the set difference of outer buffer $NB_{u,v,t_{max}}$ and inner buffer $NB_{u,v,t_{min}}$, where t_{min} is the inner and t_{max} is the outer buffer distance and the distance interval is closed i.e., inclusive of t_{min} and t_{max}.*

Thus, $NHH_{u,v,t_{min},t_{max}} = Cl(NB_{u,v,t_{max}} \setminus NB_{u,v,t_{min}})$. The sum of the length of the edges in NHH is denoted by $L_{NHH} = \sum l_{e \in NHH}$.

In Fig.5.2(b), black edges represent the $NB_{G4,G6,1}$ and red nodes and edges represent the $NHH_{G4,G6,1,2}$ around the path $p_{G4,G6}$ with $t_{min} = 1$ and $t_{max} = 2$.

Definition 30 ***Log Likelihood Ratio*** *(Log LR_{NHH}) is the test statistic for a candidate NHH. Since a NHH is on a road network, it uses L_{NHH}, instead of the hotspot area as used in [178]. The equation can be shown as:*

$$Log\ LR_{NHH} = Log\left(\left(\frac{c}{B}\right)^{c} \times \left(\frac{|A|-c}{|A|-B}\right)^{|A|-c} \times I()\right) \tag{5.1}$$

$$B = \frac{|A| \times L_{NHH}}{L_{total}} \quad and\ I() = \begin{cases} 1, & if\ c > B \\ 0, & otherwise, \end{cases}$$

B is the expected and c is the observed number of points for a NHH, $|A|$ is the cardinality of A and $I()$ is an indicator function. $I() = 1$ when a candidate NHH has more points than expected ($c > B$); otherwise $I() = 0$ [10].

For example, the sum of the lengths of the edges of $NHH_{G4,G6,1,2}$ in Fig.5.2(b) is $L_{NHH} = 19$ and the total length of the spatial network is $L_{total} = 158$. Thus $B = \frac{50 \times 19}{158} = 6.012$. In this *NHH*, there are $c = 30$ points. Thus, $I = 1$ since $30 > 6.012$.

Using Eq. 5.1, $Log\ LR_{NHH} = Log\left(\left(\frac{30}{6.012}\right)^{30} \times \left(\frac{50-30}{50-6.012}\right)^{50-30} \times 1\right) = 32.46$

Definition 31 ***A Hypothesis Test*** *determines whether a NHH occurred by chance or not. The null hypothesis H_0 states that the points are randomly distributed on a spatial network and the alternative hypothesis H_1 states that the candidate NHH has a significantly higher number of activities than outside. In order to determine the hypothesis test result, the significance level (*p-value*) of a NHH is computed by finding the order of the actual Log LR_{NHH} in the test statistic distribution (obtained by Monte Carlo simulations) and dividing that position by $m + 1$. If the* p-value *of a NHH is lower than the desired threshold (α_p), the H_1 cannot be rejected, and we say that the candidate NHH is a significant NHH. Note that these concepts are inherited from SaTScan [10].*

5.2.2 Problem Statement

Formally, Mining Network Hotspots with Holes (MNHH) problem is as follows:

Given:

1. A spatial network $G = (N, E)$ with activity count function $a(u, v) \geq 0$ and length function $l(u, v) > 0$ for each edge $e_i \in E$,

2. A log likelihood ratio threshold (θ) and a *p-value* threshold (α_p),

3. A number of Monte Carlo simulation trials (m),

4. A maximum outer buffer distance ($\widehat{t_{max}}$) and a unit distance (ω),

Find: Network hotspots with holes with $Log\ LR_{NHH} \geq \theta$ and $p\text{-}value \leq \alpha_p$.
Objective: Computational efficiency and correctness of the output.
Constraint: Nodes $n_j \in NHH$ may not be connected to each other.

Example: The graph in Fig.5.2(a) can be viewed as a road network, composed of streets (edges) and intersections (nodes). The aim is to find network hotspots with holes (*NHH*) that meet the given log likelihood ratio and significance levels (*p-value* threshold θ). In environmental criminology, finding such a hotspot may have two benefits: (1) it may focus the search for a criminal to the path at the center of *NHH*. (2) it may help determine the locations to deploy new police patrols to prevent crime. In Fig.5.2(b), $NHH_{G4,G6,1,2}$ is returned since $Log\ LR_{NHH} = 32.46$ and $p\text{-}value = 0.01$. Although the output includes more *NHH*s, since the other *NHH*s were overlapping, we show only the *NHH* with highest $Log\ LR_{NHH}$ to reduce the visual clutter.

5.3 Proposed Approach

In this section, we first describe a naïve version of our network hotspot with hole generator algorithm (NaïveNHHG). Then we present our SmartNHHG algorithm with refinements that include two novel dynamic programming approaches and a Monte Carlo simulation speed-up. The proposed algorithms present steps for candidate enumeration, candidate evaluation using *Log LR* and statistical significance test. It should be noted that in some communities these steps are practiced separately. However, in this work, we present algorithms that describe these processes together for the sake of self-containment.

5.3.1 Naïve NHH Generator Algorithm

Algorithm 8 presents the pseudocode for the NaïveNHHG approach. The algorithm begins by creating all pair shortest paths, P_{apsp}, in the spatial network (step 1). Next, each shortest path is used as a center to enumerate *NHH*s with different inner and outer

buffer distances (t_{min}, t_{max})(step 2). Finally, the statistical significance of each *NHH* is evaluated by m Monte Carlo simulations and the significant *NHH*s are returned (as the output).

NaïveNHHG Example: Table 5.1 shows a sample execution trace of NaïveNHHG. The spatial network has 92 nodes, 158 edges, and 50 activities (green squares on the edges). All edge lengths are set to 1 for illustration purposes. Inputs are set to log likelihood ratio threshold $\theta = 30$, *p-value* threshold $\alpha_p = 0.01$, maximum outer buffer distance $\widehat{t_{max}} = 5$ and unit distance $\omega = 1$.

Algorithm 8 NaïveNHHG Algorithm

Input:

1) A spatial network $G = (N, E)$ with activity count function $a(u, v) \geq 0$ and length function $l(u, v) > 0$ for each edge $e_i \in E$,

2) A log likelihood ratio threshold (θ) and a *p-value* threshold (α_p),

3) A maximum outer buffer distance $(\widehat{t_{max}})$ and a unit distance (ω)

Output:

Network hotspots with holes (*NHH*) with *p-value*$\leq \alpha_p$

Algorithm:

Step 1: Generate all pair shortest paths P_{apsp}

For each shortest path $p_{u,v} \in P_{apsp}$

Step 2: Enumerate candidate *NHH* with t_{min} and t_{max}

Step 3: Significant *NHH* $\leftarrow$ candidate *NHH* with *p-value*$\leq \alpha_p$ using m Monte Carlo simulations

Return Significant *NHH* with *Log* $LR_{NHH} \geq \theta$

In step 1 of Table 5.1, all pairs of shortest paths are computed as shown in the first column (4 out of 2.5×10^4 are shown). In step 2, the *NHH*s are enumerated by using the set difference of NB with t_{min} and t_{max}. Then, *Log* LR_{NHH} are computed for each *NHH* and *NHH*s with *Log* $LR_{NHH} \geq 30$ are stored as candidates. In step 3, the significance of candidate *NHH*s are determined and significant *NHH*s are returned (as the output) as shown in Fig.5.2(b). Although many *NHH*s were evaluated as significant, only the *NHH* on the top row of Table 5.1 is returned since $\theta = 30$. If a user is interested in all significant *NHH*, θ threshold can be set 0. Also, one may notice that many *NHH*s were

similar in the output. This issue is discussed in Section 5.6.

Table 5.1: An example execution trace of NaïveNHHG.

Step 1		**Step 2**					**Step 3**
Start - End	*Path*	t_{min}	t_{max}	c_{NHH}	*l*	Log LR_{NHH}	p-value$_{NHH}$
<G4,G6>	*[G4, G5, G6]*	*1*	*2*	*30*	*19*	*32.46*	*0.01*
<G4,H6>	*[G4, G5, G6, H6]*	*1*	*2*	25	21	20.39	0.01
<G4,F6>	*[G4, G5, G6, F6]*	1	2	27	19	*24.59*	0.01
<G4,G7>	*[G4, G5, G6, G7]*	1	2	*25*	*20*	*21.17*	*0.01*
…	…	…	…	…	…	…	…

Enumerating Candidate NHH: Algorithm 9 shows the steps of candidate *NHH* enumeration on NaïveNHHG. For each shortest path ($p_{u,v}$) in the set of all pair shortest paths (P_{apsp}) (line 1), candidate *NHH*s are enumerated as follows: First, inner and outer NB are defined by t_{min} and t_{max} (line 2-3). Note that these values are changed by unit distance ω on every iteration and even for a single $p_{u,v} \in P_{apsp}$, *NHH*s with different t_{min} and t_{max} are enumerated. Next, for each node n_j of $p_{u,v}$, single source shortest paths from that node to all other nodes in the spatial network are enumerated (line 5). If the length of any of these shortest paths is less than the t_{min}, it is saved in $NB_{u,v,t_{min}}$. Similarly, if the length of any of these shortest paths is less than t_{max}, it is saved in $NB_{u,v,t_{max}}$ (line 6-9). Finally, $NHH_{u,v,t_{min},t_{max}} = Cl(NB_{u,v,t_{max}} \setminus NB_{u,v,t_{min}})$ (line 10) and its *Log* LR_{NHH} is computed using $L_{NHH} = \sum l_{e \in NHH}$ and its activity count. If *Log* $LR_{NHH} \geq \theta$ threshold, then the *NHH* is saved as a candidate (line 11). This process is repeated for all paths in P_{apsp} and t_{min} and t_{max} until $\widehat{t_{max}}$.

Algorithm 9 Enumerating Candidate *NHH* - NaïveNHHG

```
for each p_{u,v} ∈ P_apsp do
  for each t_max = 2ω to t̂_max do
    for each t_min = ω to t_max do
      for each n_j ∈ p_{u,v} do
        for each Single Source Shortest Path p_{n_j,n} ∈ P_sssp do
          if t_min > L_{p_{n_j},n} then
            NB_{u,v,t_min} ← Edges and Nodes from p_{n_j,n}
          if t_max ≥ L_{p_{n_j},n} then
            NB_{u,v,t_max} ← Edges and Nodes from p_{n_j,n}
          NHH_{u,v,t_min,t_max} ← Cl(NB_{u,v,t_max} \ NB_{u,v,t_min})
          Candidate NHH ← NHH with Log LR_NHH ≥ θ
```

In NaïveNHHG, candidate *NHH*s are enumerated by varying t_{min} and t_{max} for all pairs of shortest paths. However, enumeration becomes exorbitant even for road networks of 10^2 nodes. To improve the scalability of NaïveNHHG, we analyzed NaïveNHHG and determined the redundant computations. Next, we propose refinements to reduce redundant computations but increase scalability.

5.3.2 Smart NHH Generator Algorithm

This section explains our smart approach for solving the MNHH problem. Our algorithm features three key ideas for achieving computational savings while maintaining result quality: Distance based dynamic programming, edge stitching and Monte Carlo simulation speed-up.

Distance Based Dynamic Programming (DP) Approach:

Algorithm 10 shows the steps of distance based DP approach, which avoids redundant calculation of *NHH* with different t_{min} and t_{max} by enumerating *NHH* with $t_{max} - t_{min} = \omega$ (line 2-9) and then using the set union of these to create *NHH*s with different inner and outer NB (line 10-12). A simplified example can be seen in Fig.5.4. In this example, in order to enumerate $NHH_{F4,F5,1,3}$ (on the right), the set union of $NHH_{F4,F5,1,2}$ (on

the left) and $NHH_{F4,F5,2,3}$ (in the middle) is used. Thus, instead of running a new enumeration process for $NHH_{F4,F5,1,3}$, the algorithm simply uses the previously computed $NHH_{F4,F5,1,2}$ and $NHH_{F4,F5,2,3}$.

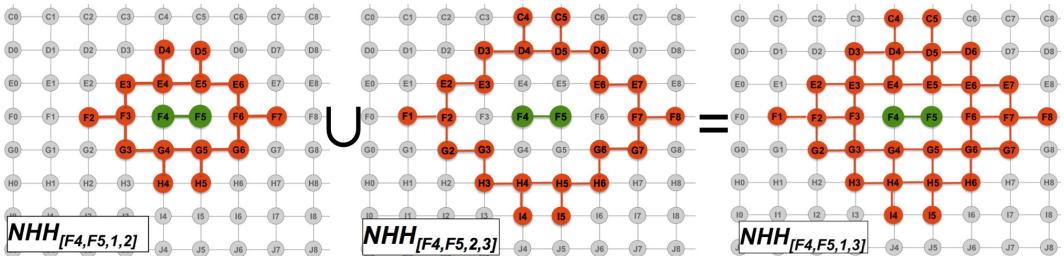


Figure 5.4: Distance based dynamic programming approach. To determine $NHH_{F4,F5,1,3}$, the set union of $NHH_{F4,F5,1,2}$ and $NHH_{F4,F5,2,3}$ is used (best in color).

Algorithm 10 Enumerating Candidate *NHH* - Distance Based DP Approach

for each $p_{u,v} \in P_{apsp}$ **do**
  for each $t_{max} = 2\omega$ to $\widehat{t_{max}}$ and $t_{min} = t_{max} - \omega$ **do**
    for each $n_j \in p_{u,v}$ **do**
      for each Single Source Shortest Path $p_{n_j,n} \in P_{sssp}$ **do**
        if $t_{min} > L_{p_{n_j},n}$ **then**
          $NB_{u,v,t_{min}} \leftarrow$ Edges and Nodes from $p_{n_j,n}$
        if $t_{max} \geq L_{p_{n_j},n}$ **then**
          $NB_{u,v,t_{max}} \leftarrow$ Edges and Nodes from $p_{n_j,n}$
        $NHH_{u,v,t_{min},t_{max}} \leftarrow Cl(NB_{u,v,t_{max}} \setminus NB_{u,v,t_{min}})$
  for each $t_{max} = 2\omega$ to $\widehat{t_{max}}$ **do**
    for each $t_{min} = \omega$ to t_{max} **do**
      $NHH_{tmin,tmax} = NHH_{tmin+\omega,tmax} \bigcup NHH_{tmin,tmax-\omega}$

Edge Stitching Approach:

Edge Stitching exploits a basic property of paths, i.e. every path consists of edges. Thus, *NHH*s around single edges can be enumerated, then these can be stitched to create *NHH*s around longer paths (avoid Line 1 of Algorithm 9). In Algorithm 11, first,

all *NHH*s around single edges are enumerated (Line 1-9). Next, these *NHH*s are stitched to create *NHH*s for longer paths (Line 10-14) as illustrated in Fig.5.5. In this example, in order to create $NHH_{G4,G6,1,2}$, $NHH_{G4,G5,1,2}$ and $NHH_{G5,G6,1,2}$ are stitched together. Once we create the set union of these *NHH*s, edges and nodes that belong to $NB_{G4,G6,1}$ are removed to determine $NHH_{G4,G6,1,2}$ (in Fig.5.2(b)).

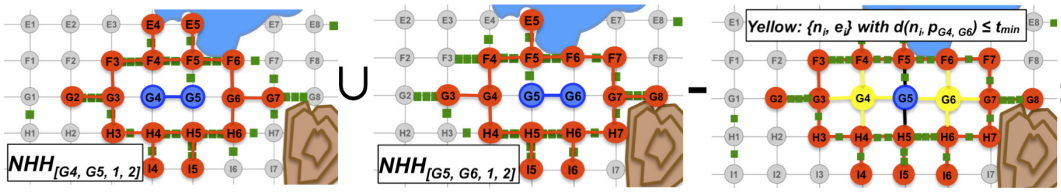


Figure 5.5: Edge stitching approach. To determine $NHH_{G4,G6,1,2}$ (Fig.5.2(b)); $NHH_{G4,G5,1,2}$ and $NHH_{G5,G6,1,2}$ are stitched together and then the nodes and edges of $NB_{G4,G6,1}$ are removed (best in color).

Algorithm 11 Enumerating Candidate *NHH* - Edge Stitching Approach

for each $e_i \in E$ **do**
  for each $t_{max} = 2\omega$ to $\widehat{t_{max}}$ **do**
    for each $t_{min} = \omega$ to t_{max} **do**
      for each Single Source Shortest Path from n_u and n_v $p_{n_j,n} \in P_{sssp}$ **do**
        if $t_{min} > L_{p_{n_j},n}$ **then**
          $NB_{u,v,t_{min}} \leftarrow$ Edges and Nodes from $p_{n_j,n}$
        if $t_{max} \geq L_{p_{n_j},n}$ **then**
          $NB_{u,v,t_{max}} \leftarrow$ Edges and Nodes from $p_{n_j,n}$
        $NHH_{u,v,t_{min},t_{max}} \leftarrow Cl(NB_{u,v,t_{max}} \setminus NB_{u,v,t_{min}})$
for each $p_{u,y} \in P_{apsp}$ **do**
  for each $t_{max} = 2\omega$ to $\widehat{t_{max}}$ **do**
    for each $t_{min} = \omega$ to t_{max} **do**
      $NHH_{u,y,t_{min},t_{max}} = Cl(NHH_{u,v,t_{min},t_{max}} \bigcup NHH_{v,y,t_{min},t_{max}} \setminus NB_{u,y,t_{min}})$
      Candidate $NHH \leftarrow NHH_{u,y,t_{min},t_{max}}$ with $Log\ LR_{NHH} \geq \theta$

Monte Carlo Simulation Speed-Up:

The following three refinements are used to speed-up Monte Carlo simulations. First refinement is to create all pair shortest paths once, and use them for each simulation trial. NaïveNHHG runs for m times for Monte Carlo simulations. However, the spatial network does not change between iterations. Thus, we prevent redundant shortest path calculations in Monte Carlo Simulations. Second, if any NHH^{random} has $Log\ LR_{NHH}^{random} \geq Log\ LR_{NHH}^{actual}$ then that iteration terminates since there is no reason to keep looking at all NHHs in that random dataset if a NHH^{random} beats the maximum of $Log\ LR_{NHH}$ from the actual dataset. Third, Monte Carlo simulation is terminated if the $p\text{-}value \geq \alpha_p$ because the α_p threshold won't be met at the end.

It should be noted that these and similar refinements are often used in related work to speed-up the Monte Carlo simulation process [10, 12, 212]. Therefore, details of the execution trace of those speed-up approaches are omitted from this paper.

5.4 Case study

We conducted two case studies to evaluate SmartNHHG qualitatively comparing its output with the output of SaTScan [10] and a ring-shaped hotspot detection [141] method using two real crime datasets (Fig.5.6(a) and Fig.5.7(a)). For both of the case studies, we matched activities to edges as counts. The road network was obtained from the US Census Bureau Tiger/Line Shapefile [213]. The map visualizations were prepared using QGIS and Open Layers Plugin (www.qgis.org).

The first crime dataset in Fig.5.6(a) consists of 64 theft committed between 2013 and 2014 in South Side Neighborhood of Chicago, Illinois [214]. We set $\omega = 0.04$ km, $\widehat{t_{max}} = 0.6$ km, $\theta = 20$ and $\alpha_p = 0.01$.

The second crime dataset in Fig.5.7(a) consists of 128 burglary crimes committed between 2013 and 2014 in Caballo Hills Neighborhood of Oakland, California [215]. We set $\omega = 0.5$ km, $\widehat{t_{max}} = 2.6$ km, $\theta = 20$ and $\alpha_p = 0.03$.

For the first crime dataset in Fig.5.6(a), SaTScan produced a small circular hotspot as shown in Fig.5.6(b). This is due to the fact that those activities occurred close to each other. For the second crime dataset in Fig.5.7(a), SaTScan's output was a large circular hotspot. Since SaTScan uses Euclidean distances to enumerate circles, none of

the outputs for the case studies reflected the effect of the road network.

In Fig.5.6(c) and Fig.5.7(c), ring-shaped hotspots returned by RHD indicate single center rings (i.e. a single crime source location) due to its enumeration method and space (i.e. Euclidean). Although the output of RHD in Fig.5.6(c) aligns with the activities in the study area due to the street morphology in Chicago, it produced very different results for the burglary crimes in Oakland, California.

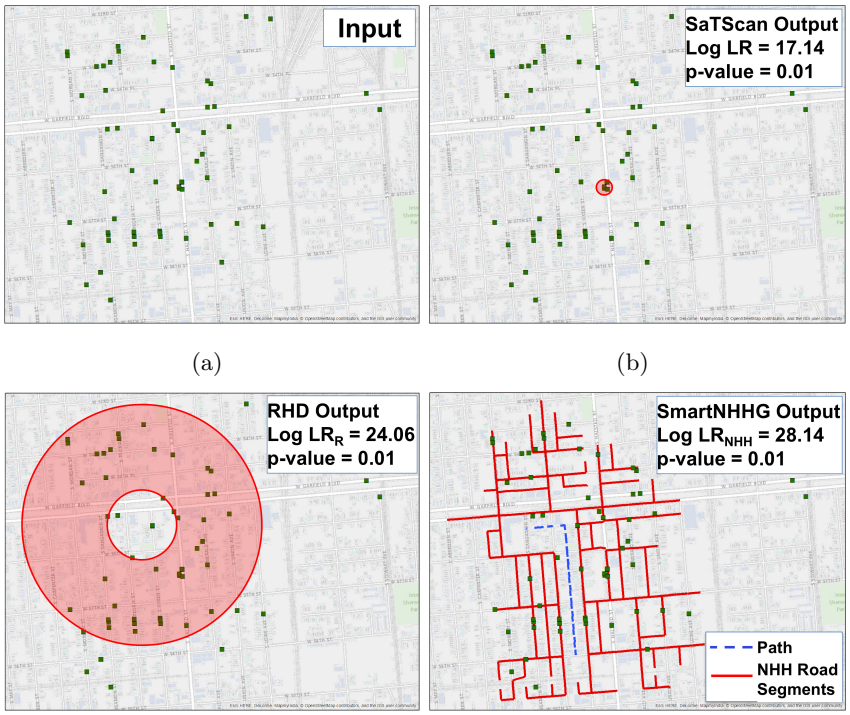


Figure 5.6: Case Study 1: Theft crimes in Chicago, Illinois. Fig.5.6(a) shows the input, Fig.5.6(b) shows the output of SaTScan, Fig.5.6(c) shows the output of Ring-Shaped Hotspot Detection and Fig.5.6(d) shows the output of SmartNHHG. *Log LR* values are not comparable due to the Euclidean and Network spaces. (best in color).

As noted earlier, criminals are known to commit crimes around the routes they often commute as described in environmental criminology [14]. Therefore, when we take a look at the output of SmartNHHG in both case studies (Fig.5.6(d), Fig.5.7(d) and Fig.5.7(e)), we see that the output aligns with such crime patterns. For example,

blue paths at the centers of the *NHH*s in Fig.5.7(d) and 5.7(e) are the only routes to reach those houses that burglary crimes occurred (i.e. activities) which may make sense in the context of environmental criminology. Finally, it should be noted that our tool should be considered as a decision support tool for the analysts and the results should be analyzed by them in the context of additional domain information to prevent potentially misleading results.

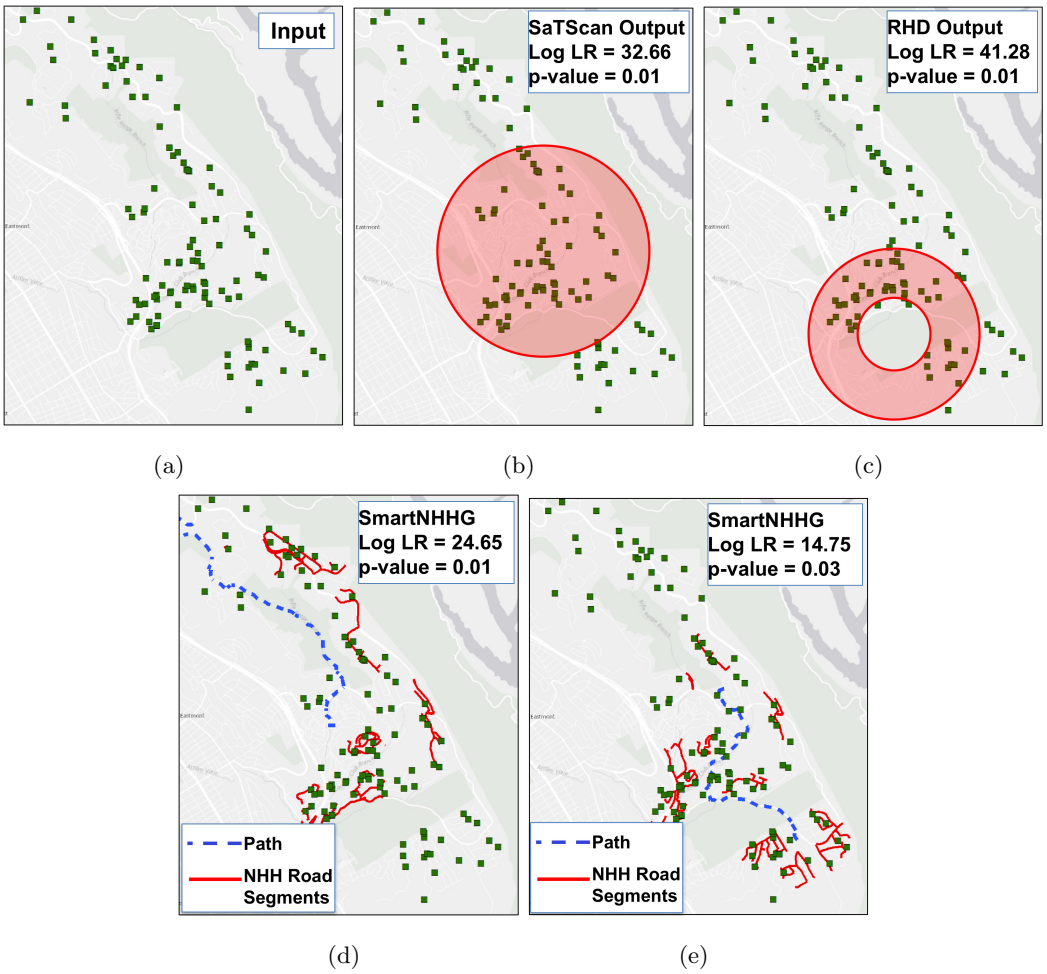
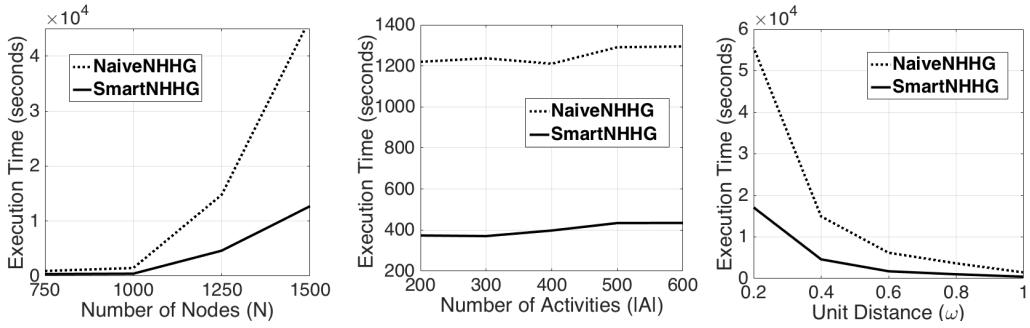


Figure 5.7: Case Study 2: Burglary crimes in Oakland, California. Fig.5.7(a) shows the input, Fig.5.7(b) shows the output of SaTScan, Fig.5.7(c) shows the output of Ring-Shaped Hotspot Detection and Fig.5.7(d) and 5.7(e) shows the output of SmartNHHG. *Log LR* values are not comparable due to the Euclidean and Network spaces.

5.5 Experimental Evaluation

We also conducted an experimental evaluation to observe the effect of the algorithmic refinements compared with the Naïve approach. The experiments were performed on real-world data obtained from the City of Chicago portal [214]. The dataset contained 676 theft crimes that were committed in Chicago, Illinois, between 2013 and 2014. The road network was obtained from the US Census Bureau Tiger/Line Shapefile [213]. For each edge on the road network, activities were matched and their counts on edges were aggregated. In the experiments, the number of Monte Carlo simulation trials was set to $m = 0$, since we did not perform any experiments on m due to the fact that our Monte Carlo simulation speed-up approaches were trivial and previously used in [12, 212].

Effect of the number of Nodes: We varied the number of nodes from 750 to 1500, causing the asymptotic increase on the all pair shortest paths, since total number of all pairs will be $\binom{750}{2}$ and $\binom{1500}{2}$ respectively. We set the log likelihood ratio threshold to $\theta = 20$. We also selected the maximum outer buffer distance ($\widehat{t_{max}} = 5$ km) and the unit distance to $\omega = 1$ km (note that these inputs will be selected by domain experts). SmartNHHG is faster than the NaïveNHHG. Also we can observe that the computational savings increase with increasing number of nodes thanks to SmartNHHG's edge stitching approach.



(a) (b) (c)

Figure 5.8: Scalability of SmartNHHG with increasing 5.8(a) number of nodes, 5.8(b) number of activities 5.8(c), and unit distance (ω).

Effect of the number of Activity Points: We also varied the number of activities in the activity set as shown in Fig.5.8(b). Since the algorithm uses only the activity count

on each edge, this experiment did not affect the execution times of either algorithm. However, SmartNHHG performs around three times faster than the NaïveNHHG. In the future, we plan to leverage activity counts on edges to improve the scalability of SmartNHHG.

Effect of unit distance ω: In this experiment, inputs are the number of nodes $|N| = 1000$ and the maximum outer buffer distance $\widehat{t_{max}} = 5$ km. The unit distance ω was varied by 0.2, 0.4, 0.6, 0.8 and 1 km. In Fig.5.8(c), SmartNHHG is faster and computational savings increase with smaller ω thanks to SmartNHHG's distance based dynamic programming approach.

In summary, experiments confirm that SmartNHHG performs faster than NaïveNHHG thanks to the proposed algorithmic refinements.

5.6 Discussion

Techniques without Significance Test: This paper focuses on hotspot detection techniques that use statistical significance to remove chance patterns but there are also techniques that do not test for statistical significance. These techniques (i.e. DBSCAN [9], K-Means [57], KMR [216], Clumping [206]) are state-of-the-art to detect clusters (i.e. a set of objects partitioned into a set of meaningful sub-classes) in a point process. However, since they do not test for statistical significance, they are not suitable for applications where false positive results may cause harm. For example, a neighborhood falsely identified as a crime hotspot may become stigmatized, causing residents' property values to drop. In addition, adding significance test to these approaches is often non-trivial since they lack a metric (e.g. log likelihood ratio test) for ranking candidate clusters. Thus, techniques without statistical significance test were not considered in our work.

Post-Processing of the Output: Our proposed approach returns all possible *NHH* given an activity set and a spatial network. However, during our experiments, we often observed that multiple overlapping hotspots were returned on the same subgraph of the spatial network. To reduce the visual clutter, we used two simple rules in our visualizations: *(1) For two p_k and p_l:* If $p_l \subset p_k$ and there are two significant NHH_k and NHH_l and $t^k_{min} = t^l_{min}$ and $t^k_{max} = t^l_{max}$, then only NHH_k will be returned. *(2) For*

a path p: If there are two significant NHH_i, NHH_j and $t^i_{min} \leq t^j_{min}$ and $t^i_{max} \geq t^i_{max}$ then only NHH_i will be returned.

5.7 Conclusion

This work explored the problem of mining network hotspots with holes in relation to important application domains such as crime analysis. We proposed a Smart Network Hotspot with Hole Generator algorithm that discovers multiple network hotspots with holes (*NHH*) on a spatial network. The proposed approach uses distance based dynamic programming and edge stitching approaches as well as Monte Carlo simulation speed-ups to enhance its performance. We presented two case studies using crime activity sets comparing our proposed approach with a ring-shaped hotspot detection method. Experimental evaluation using real data indicates that the proposed algorithmic refinements yield substantial computational savings without sacrificing result quality.

In future, we plan to explore refinements including sub-edge level *NHH* enumeration, active node filtering and dynamic segmentation. We also plan to explore "emerging" *NHH*s from spatiotemporal activity sets (i.e. time tags for the activities). Additionally, factors (e.g. demographics, activity relationships [217], urbanization [218, 219]) that generate *NHH* will be explored.

Chapter 6

Conclusions and Future Work

6.1 Key Results

Despite the significant growth in geospatial data science applications, relatively little attention has been paid to formalizing the scientific foundation of geospatial data science. The lack of scientifically consistent approaches has made it difficult to develop reliable and trustworthy geospatial models and tools. The specific properties of geospatial data, its volume, variety and velocity and the implicit but complex spatial relationships limit the applicability of traditional data science methods. This thesis defines geospatial data science as a scientific process of extracting valuable information from raw geospatial data with reasonable effort. My research explores this emerging field from a transdisciplinary perspective across the three closely related scientific disciplines of statistics, mathematics and computer science. I have also presented some challenges that are related to geospatial data science and illustrated how these can be addressed via examples and proposed tools. More specifically, the work addresses the computational scalability issues by proposing fast algorithms that use either a prune and refine strategy which prevents the unnecessary enumeration of uninteresting patterns or using dynamic programming approaches. The issues that arise from chance patterns are addressed by using a statistical approach, e.g. spatial scan statistics. Also, the mathematical completeness issue is addressed with a dual grid approach that enumerates pattern families via their mathematical parameters. A summary of the contributions related to their scientific disciplines is shown in Table 6.1.

Table 6.1: Thesis contributions and their related geospatial data science focus.

Science	Discipline	Focus	Thesis Contributions
Geospatial Data Science and Geospatial Data-driven Science	Mathematics	- Correctness - Completeness	- Parameter Grids ensures all pattern families are represented (completeness) (Ch. 3-4) - Pattern families are evaluated against certain thresholds (correctness) (Ch. 3-5) - Other patterns can be represented by their geometric parameters. (Ch. 3-4)
	Statistics	- Reduce chance patterns	- Reduces chance patterns by evaluating each candidate against a significance threshold (Ch. 3-5).
	Computer Science	- Computational Scalability	- A Prune and Refine framework to reduce the computational cost (Ch. 3-4) - Dynamic programming approaches to improve scalability (Ch. 5)
	Application Domain	- Domain Specific Theories	- Incorporates domain specific theories (e.g. diffusion theory, routine activity theory, crime pattern theory) (Ch. 3-5)

6.2 Short Term Future Directions

In the short term, I plan to investigate several issues related to those individual fields of geospatial data science as follows: (a) Parallelization of the proposed approaches for greater computational scalability. (b) Richer enumeration space for better mathematical completeness. (c) Using other statistical measures to provide tighter statistical significance and eliminating more chance patterns (d) Using more domain related information to identify new patterns.

Parallelization of the Proposed Approaches: As geospatial dataset sizes increase and the number of observations grows ever larger, parallelization may need to be considered for greater scalability. However, there are challenges that need to be addressed to pinpoint the algorithm bottlenecks and overcome the issues that arise from the synchronization across multiple iterations of the algorithm. I plan to deploy our algorithm either on Apache Spark, which uses an in-memory approach, or the Hadoop Map-Reduce framework. One approach may be to parallelize the prune phase parametric grid cell computations in a mapping phase and merge the results in the reduce phase. Note that the previous experiments on the proposed approaches showed that

finer resolutions in grid cell sizes makes the prune phase output close to the output after the refine phase. Thus, using a parallelization approach with extremely fine cell sizes on the prune phase may make it possible to eliminate the need for a refine phase.

Richer Enumeration Space: The proposed approaches described in this thesis use a generic approach for prune phases but refine phases define rings and circles as having a specific number of points on their perimeter. Thus, approaches with a richer refine phase may be needed to give a more complete set of patterns. An argument can be made that using a prune phase with finer resolutions as defined above may solve the problem by approximating the rings and circles in parameter space to the actual patterns in the study area. Also, while my research approach uses concentric rings and perfect circles, I may consider approaches that use non-concentric rings as well as distorted circles to achieve a richer enumeration space.

In addition to creating a richer enumeration space for ring shaped and geographically robust hotspot detection (in Euclidean space), we can also enrich the enumeration space for finding network hotspots with holes. In the current work, shortest paths are used at the center of a hotspot pattern. However, in a real world scenario, shortest paths may not reflect actual driver behavior because real-world drivers consider not only the length of their trip but also other factors (e.g., rush hour, road construction, etc.) that may affect their trip duration. Thus, it may be beneficial to explore simple-path based network hotspots with holes. Current approaches for enumerating all simple paths in a graph fall roughly into three categories: (i) approaches that use the power of adjacency matrices, (ii) approaches that use depth/breadth first search (DFS/BFS) with backtracking and (iii) approaches that use cut vertices (i.e. articulation points) and bi-connected components. Adjacency-matrix based methods have high cost of matrix multiplications and equal edge weight assumptions, DFS/BFS with backtracking methods consider only a single node pair on each iteration, and the methods that use cut vertices assume there are enough cut vertices to reduce the enumeration cost. However, cut vertices are rare in transportation networks. My approach would be based on a divide and conquer paradigm leveraging graph-partitioning into hierarchical fragments, then computing all simple paths inside fragments and between fragments, and then using the output of these sub-problems to enumerate all possible simple paths between any pairs of locations. The total cost of computing simple path-based network hotspots

with holes depends not only on the cost of all simple paths for a node pair, but also on the cost of testing for statistical significance. Thus, to further reduce the total computational cost of the divide and conquer approach, I hope to investigate ways to prune out node pairs that are not going to contribute to the final output as well as reduce the cost of statistical significance testing.

Other Statistical Measures: In this thesis, the proposed approaches use spatial scan statistics with a log likelihood ratio to eliminate the chance patterns. Spatial scan statistics, as is, uses a Monte Carlo simulation framework using randomization. Often, this approach causes the algorithms to run thousands of times to identify the test statistic distribution with random datasets. Promising future areas of investigation include ways to terminate Monte Carlo simulation once chance patterns are eliminated; Bayesian spatial scan statistics to evaluate the posterior probabilities of candidate hotspot regions; and methods that use other statistical measures to identify the locations with high number of activities. For example, Getis Ord Gi* as well as Local Moran's I statistical measures can be used to filter out unnecessary candidates from a dataset which can later be evaluated for a specific type of pattern.

Domain Specific Pattern Detection: My dissertation research investigated hotspot patterns using theories from environmental criminology (e.g. routine activity theory and crime pattern theory) and epidemiology (diffusion theory). Those theories were the basis of our work for defining the shapes of the patterns. In fact, there are many theories as well as patterns to be investigated for specific application domains. Table 6.2 lists examples of evasion related patterns that can be found in various real-world settings. The first row (in red) refers to the patterns already studied in this thesis but the other evasive patterns in the table have yet to be studied. For example, one could identify the neighborhoods that people intentionally avoid (e.g. for having heavy traffic or being "sketchy"), as well as evasive individuals (e.g. criminals who avoid checkpoint locations, taxi drivers who avoid specific risky neighborhoods, etc.) by comparing the GPS trajectories with the corresponding shortest paths. Similarly, by analyzing GPS trajectory datasets, irregular behaviors could be detected such as visits to unknown locations during abnormal times.

Table 6.2: Patterns of evasion that could be studied in different domains.

Application Domain: Identifying Patterns of Evasion		
Pattern Family	**Activity**	**What is Being Evaded**
Ring-Shaped Hotspots (Doughnut Patterns)	Crime Locations of a crime source (e.g. serial criminal)	- Neighbors - Police Stations - Locations that recognize the criminal
Evasive Trajectories	- Self-driving cars	- Collision with others (e.g. cars, pedestrians)
	- Pac-Man Video Game	- Monsters
	- Insurgents	- Security checkpoints - Surveillance cameras - Guardians (e.g. security enforcements)
	- Drivers	- Congestion - Unsafe Areas (crime-ridden) - Construction Sites on the Road
	- Planes	- Adversary Radar Sites - Bad Weather - Missiles
Evasive Visits	- Visit outside regular hours (e.g. visit office at night) - Visit unknown locations.	- Detection by others

6.3 Long Term Future Directions

In this thesis I did not discuss the effect of the size of the study area on the output patterns. Current analytic tools are often affected by the scale, causing them to miss patterns when the scale of the study area changes. For example, a hotspot detection method may miss spatial patterns at a county level but not at a state level even when the input activities/events are the same. Such effects can not only be seen in hotspot detection methods but also seen in many other geospatial pattern tools such as co-location and outlier detection. Since mathematical completeness is one of the goals of geospatial data science, the scale should not affect the result quality. Thus, there is a need for a deeper investigation of statistical measures as well as computational methods to reduce the effect of scale.

Finally, current tools and approaches are often data-driven and empirical, but this does not mean that they respect the laws of physics. In other words, identifying patterns using only the data in hand may result in patterns which violate the laws of physics. To

compensate, the result sets may need to be reviewed by analysts for such considerations. Incorporating laws of physics will likely play an increasingly important role in data driven transdisciplinary geospatial data science research, as seen, for example, in the new "nexus" initiatives proposed recently to study food, energy, and water systems.

References

[1] Brent E Turvey. *Criminal profiling: An introduction to behavioral evidence analysis*. Access Online via Elsevier, 2011.

[2] Tarush Jain and Rohan Somni. Intel Distribution for Apache Hadoop. 4(3):530–532, 2013.

[3] Daniel A. Reed and Jack Dongarra. Exascale computing and big data. *Communications of the ACM*, 58(7):56–68, 2015.

[4] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. *Focal-test-based spatial decision tree learning: A summary of results*, pages 320–329. dec 2013.

[5] Orlando Hernandez, Sait Khurama, and Gretta C. Alexander. Structural Modeling of the Vichada Impact Structure from Interpreted Ground Gravity and Magnetic Anomalies . *Boletin de GeologÃa*, 33:15 – 26, 06 2011.

[6] New york bronx legionnaire's outbreak summer 2015. `http://www.nyc.gov/html/doh/html/diseases/cdlegi.shtml`. Accessed: 2015-12-10.

[7] Hotel that enlivened the bronx is now a "hot spot" for legionnaires'. `http://www.nytimes.com/2015/08/11/nyregion/death-toll-from-legionnaires-disease-outbreak-in-bronx-rises-to-12.html?_r=0`. Accessed: 2015-12-10.

[8] John Snow. *On the mode of communication of cholera*. John Churchill, 1855.

[9] Martin Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[10] M Kulldorff. Satscan user guide for version 9.0, 2011.

[11] Robbery (w/o weapon) cases in san diego in 2013. `http://www.sandiego.gov/police/services/ewatch.shtml`, Accessed: 2015-03.

[12] Dev Oliver et al. Significant route discovery: A summary of results. In *GIScience*, pages 284–300. Springer, 2014.

[13] Xun Tang, Emre Eftelioglu, Dev Oliver, and Shashi Shekhar. Significant linear hotspot discovery. *IEEE Transactions on Big Data*, 2017.

[14] Paul J Brantingham and Patricia L Brantingham. *Environmental criminology*. Sage Publications Beverly Hills, CA, 1981.

[15] Lawrence E Cohen and Marcus Felson. Social change and crime rate trends: A routine activity approach. *American sociological review*, pages 588–608.

[16] PJ Brantingham and PL Brantingham. Environment, routine and situation: Toward a pattern theory of crime. *Advances in criminological theory*, 5:259–294, 1993.

[17] D.K.Rossmo. *Geographic profiling*. CRC press, 1999.

[18] John Eck et al. Mapping crime: Understanding hotspots. 2005.

[19] Ned Levine. Crime mapping and the crimestat program. *Geographical Analysis*, 38(1):41–56, 2006.

[20] David J Icove and MH Estepp. Motive-based offender profiles of arson and fire-related crimes. *FBI law enforcement bulletin*, 56(4):17–23, 1987.

[21] Rigel software web site. http://www.rigelanalyst.net. Accessed: 5/1/2016.

[22] National Research Council. *Big Data: A Workshop Report*. The National Academies Press, Washington, DC, 2012.

[23] Fbi uniform crime reports data portal. https://www.fbi.gov/about-us/cjis/ucr. Accessed: 5/1/2016.

[24] William R. Newman and Lawrence M. Principe. Alchemy vs. Chemistry: The Etymological Origins of a Historiographic Mistake. *Early Science and Medicine*, 3(1):32–65, jan 1998, arXiv:1011.1669v3.

[25] Pierre Legendre, Mark R T Dale, Marie-josée Fortin, Philippe Casgrain, Source Ecology, and No Dec. Effects of Spatial Structures on the Results of Field Experiments EFFECTS OF SPATIAL STRUCTURES ON THE RESULTS. *Source: Ecology Ecology*, 85(12):3202–3214, 2011.

[26] Fulvio Mazzocchi. Could big data be the end of theory in science? *EMBO reports*, page e201541001, 2015.

[27] H Miller. Geographic Data Mining and Knowledge Discovery. *New York*, page 20, 2004.

[28] Shashi Shekhar, Michael R. Evans, James M. Kang, and Pradeep Mohan. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):193–214, 2011.

[29] Declan Butler. When Google got flu wrong. *Nature*, 494(7436):155–156, feb 2013.

[30] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science*, 343(6167):1203–1205, mar 2014.

[31] Organizers P Drineas, X Huo, and Executive Summary. *Workshop on Theoretical Foundations of Data Science (TFoDS)*, pages 1–20. apr 2016.

[32] Gary Marcus and Ernest Davis. Eight (No, Nine!) Problems With Big Data. *The New York Times*, pages 1–6, 2014.

[33] Ming-hsiang Tsou. *Big data : techniques and technologies in geoinformatics*, volume 5683. CRC Press, 2014, arXiv:1011.1669v3.

[34] Axel Graumann, Tamara Houston, David Levinson, Neal Lott, Sam McCown, Scott Stephens, and David Wuertz. Hurricane Katrina: A climatological perspective: Preliminary report. *Technical report no. 2005-01.*, 1, 2006.

[35] National hurricane center. http://www.nhc.noaa.gov/. Accessed: 3/1/2017.

[36] Lorraine Daston. *Objectivity.* the Mit Press, 2007.

[37] Dr. Chris Drummond. *Replicability is not reproducibility: Nor is it good science*, pages 1–4. Number 2005. jun 2009.

[38] R.˜D. Peng. Reproducible Research in Computational Science. *Science*, 334(6060):1226–, dec 2011, 0901.4552.

[39] H G Gauch. *Scientific method in practice.* Cambridge University Press, 2003.

[40] Nassim Nicholas Taleb. *The black swan : the impact of the highly improbable*, volume 2. Random house, 2007, arXiv:1011.1669v3.

[41] R Evans Michael, Oliver Dev, Zhou Xun, and Shekhar Shashi. Spatial Big Data. *Big Data*, pages 149–176, 2014.

[42] S Shekhar, Z Jiang, R Y Ali, E Eftelioglu, X Tang, V M V Gunturi, and X Zhou. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information*, 4(4):2306–2338, 2015.

[43] Osm planet gpx web site. http://planet.openstreetmap.org/gps/. Accessed: 03/01/2017.

[44] Eliseo Clementini and Paolino Di Felice. An algebraic model for spatial objects with indeterminate boundaries. *Geographic objects with indeterminate boundaries*, 2:155–169, 1996.

[45] A. G. Cohn and N. M. Gotts. The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. *Proceedings of the GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries*, 2:171–187, 1996.

[46] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, feb 2011, 1010.0302.

[47] Hanan Samet. Sorting Spatial Data . 2016.

[48] C R Ehlschlaeger and M F Goodchild. *Uncertainty in spatial data: Defining, visualizing, and managing data errors*, pages 1–8. 1994.

[49] Mei-Po Kwan, Irene Casas, and Ben Schmitz. Protection of Geoprivacy and Accuracy of Spatial Information: How Effective Are Geographical Masks? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(2):15–28, 2004.

[50] E W Dijkstra. A Note on Two Probles in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[51] Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *International Workshop on Experimental and Efficient Algorithms*, pages 319–333. Springer, 2008.

[52] Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7, 2000.

[53] Zhe Jiang, Shashi Shekhar, Pradeep Mohan, Joseph Knight, and Jennifer Corcoran. Learning spatial decision tree for geographical classification: a summary of results. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 390–393. ACM, 2012.

[54] Baris M Kazar, Shashi Shekhar, David J Lilja, D Shires, J Rogers, and M Celik. *A parallel formulation of the spatial auto-regression model*, volume 72, pages 58–73. Citeseer, 2005.

[55] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.

[56] Giorgio Giacinto, Fabio Roli, and Lorenzo Bruzzone. Combination of neural and statistical algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters*, 21(5):385–397, 2000.

[57] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[58] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[59] Søren Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica: Journal of the Econometric Society*, pages 1551–1580, 1991.

[60] Michael A Babyak. What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. *Psychosomatic medicine*, 66(3):411–421, 2004.

[61] F Melgani and L Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. and Remote Sens.*, 42(JANUARY 2004):1778–1790, 2004.

[62] Nsf workshop on geospatial data science. http://cybergis.illinois.edu/events/ geodatascience_workshop/home. Accessed: 3/1/2017.

[63] Harvey Russell Bernard. *Sampling Theory*, pages 169–185. Springer International Publishing, 2006.

[64] Hwanhee Hong, Bradley P Carlin, Tatyana A Shamliyan, Jean F Wyman, Rema Ramakrishnan, François Sainfort, and Robert L Kane. Comparing Bayesian and frequentist approaches for multiple outcome mixed treatment comparisons. *Medical decision making : an international journal of the Society for Medical Decision Making*, 33(5):702–14, jul 2013.

[65] Douglas Curran-Everett. Explorations in statistics: confidence intervals. *Advances in physiology education*, 33(2):87–90, jun 2009.

[66] Grace L Yang and L Le Cam. Asymptotics in statistics: some basic concepts. *Berlin, German: Springer*, 2000.

[67] Robert Jacobs. *Maximum Likelihood Estimation*, volume 2, pages 3–5. Springer New York, 2008.

[68] John R Rice. *Mathematical Statistics and Data Analysis*. Thomson/Brooks/Cole, 2006.

[69] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[70] Jean-Paul. Chiles and Wiley InterScience (Online service). *Geostatistics : modeling spatial uncertainty*. John Wiley & Sons, 2011.

[71] C. K. I. Williams. *Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond*, pages 599–621. Number 89. Springer Netherlands, 1998.

[72] A Stein and L C A Corsten. Universal Kriging and Cokriging as a Regression Procedure. *Source: Biometrics*, 47(2):575–587, jun 1991.

[73] Jesper Møller and Rasmus Waagepetersen. Modern Statistics for Spatial Point Processes. *Scandinavian Journal of Statistics*, 34(4):643–684, 2016.

[74] Quantum gis software web site. `http://www.qgis.org/en/site/`. Accessed: 2013-12-12.

[75] Philip M. Dixon. Ripley's K function. *Encyclopedia of Environmetrics*, 3(December):1796–1803, 2002.

[76] Arthur Getis and J. K. Ord. The Analysis of Spatial Association by Use of Distance Statistics. *Geographical Analysis*, 24(3):189–206, sep 1992.

[77] Andrew David. Cliff and J.k. Ord. *Spatial processes : models & applications* . Pion, 1981.

[78] Melanie M. Wall. A close look at the spatial structure implied by the CAR and SAR models. *Journal of Statistical Planning and Inference*, 121(2):311–324, 2004.

[79] Tang Xun, Eftelioglu Emre, Oliver Dev, and Shekhar Shashi. Significant Linear Hotspot Discovery. *IEEE Transactions on Big Data Special Issue on Urban Computing, Accepted*, pages 1–1, 2017, arXiv:1206.1208v2.

[80] Nery Sofiyanti, Dyah Iriani Fitmawati, and Andesba A. Roza. *Stenochlaena Riauensis (Blechnaceae), A new fern species from riau, Indonesia*, volume 22. The Guilford Press (ISBN: 160918176X), 2015, arXiv:1011.1669v3.

[81] R.W. Lillesand, T.M; Kiefer. *Remote Sensing and Image Interpretation*. John Wiley & Sons, dec 1987.

[82] Emre Eftelioglu, Zhe Jiang, Reem Ali, and Shashi Shekhar. Spatial computing perspective on food energy and water nexus. *Journal of Environmental Studies and Sciences*, 6(1):62–76, 2016.

[83] Radford M Neal. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

[84] K. R. Gabriel and J. Neumann. A Markov chain model for daily rainfall occurrence at Tel Aviv. *Quarterly Journal of the Royal Meteorological Society*, 88(375):90–95, jan 1962, arXiv:1011.1669v3.

[85] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*, volume 54. 1998, 1111.4503v1.

[86] I T Jolliffe. *Principal component analysis*. Wiley Online Library, 1986.

[87] Michael Kutner, Christopher Nactsheim, John Neter, and William Li. *Applied Linear Statistical Models*, volume 4. Irwin Chicago, 2004.

[88] V.V. Voyevodin. *The algebraic eigenvalue problem*, volume 6. Clarendon Press Oxford, 1966.

[89] G H Golub and C Reinsch. Singular value decomposition ans least squares solutions. *Numer. Math.*, 14(5):403–420, 1970.

[90] MichaelE. Wall, Andreas Rechtsteiner, and LuisM. Rocha. *Singular Value Decomposition and Principal Component Analysis*, pages 91–109. Springer, 2003, 0208101v4.

[91] George Scatchard. Thermodynamics. An advanced treatment for chemists and physicists. *Journal of Chemical Education*, 27(5):291, 1950.

[92] Matthias Ruth. *Information, entropy and progress: A new evolutionary paradigm*, volume 12. Springer Science & Business Media, 1995.

[93] M.A. Hall and Geoffrey Holmes. Benchmarking attribute selection techniques for data mining. *. . . and Data Engineering, IEEE Transactions . . .*, 15(6):1437–1447, 2003.

[94] J Ross Quinlan. Induction of Decision Trees quinlan.pdf. *Machine Learning*, 1(1):81–106, 1986.

[95] S R Safavian and D Landgrebe. Survey of Decision Tree Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[96] Alex McBratney, Brett Whelan, Tihomir Ancev, and Johan Bouma. Future directions of precision agriculture. *Precision Agriculture*, 6(1):7–23, 2005, 0402594v3.

[97] Naiqian Zhang, Maohua Wang, and Ning Wang. Precision agriculturea worldwide overview. *Computers and Electronics in Agriculture*, 36(2-3):113–132, 2002.

[98] Stephen P Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):126 p., 2011.

[99] Pablo Pedregal. *Introduction to Optimization*, volume 46, page 245. 2004, arXiv:1011.1669v3.

[100] Léon Bottou, N E C Labs America, and Princeton Nj. *Large-Scale Machine Learning with Stochastic Gradient Descent*, pages 177–187. Number x. Springer, 2010.

[101] Mason. *Boosting algorithms as gradient descent*, volume 3, pages 1–11. 2000, 0607324v2.

[102] Lurng-Kuo Liu and Ephraim Feig. A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Transactions on circuits and systems for Video Technology*, 6(4):419–422, 1996.

[103] Kenneth Miller and Bertram Ross. An Introduction to the Fractional Calculus and Fractional Differential Equations. 1993.

[104] K. R. K. R. Douglas-Mankin, R. R. Srinivasan, and J. G. J. G. Arnold. Soil and Water Assessment Tool (SWAT) Model: Current Developments and Applications. *Transactions of the ASABE*, 53(5):1423–1431, 2010.

[105] P W Gassman, M R Reyes, C H Green, and J G Arnold. the Soil and Water Assessment Tool: Historical Development, Applications, and Future Research Directions. *Transactions of the American Society of Agricultural and Biological Engineers*, 50(4):1211–1250, 2007.

[106] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. *Computer Networks*, 33(1):309–320, 2000.

[107] J Kleinberg, R Kumar, P Raghavan, S Rajagopalan, and A Tomkins. *The web as a graph: measurement, models and methods*, pages 1–18. Springer, 1999.

[108] Linton C Freeman. Centrality in Social Networks. *Social Networks*, 1(3):215–239, 1978.

[109] A Mislove, M Marcon, Krishna P Gummadi, P Druschel, and B Bhattacharjee. *Measurement and Analysis of Online Social Networks*, pages 29–42. ACM, 2007.

[110] Chi Li-Ping, Wang Ru, Su Hang, Xu Xin-Ping, Zhao Jin-Song, Li Wei, and Cai Xu. Structural Properties of US Flight Network. *Chinese Physics Letters*, 20(8):1393–1396, 2003.

[111] J. Brimberg, P. Hansen, K.-W. Lih, N. Mladenović, and M. Breton. An oil pipeline design problem. *Operations Research*, 51(2):228–239+340, 2003.

[112] Afra Zomorodian. Topological Data Analysis course. *Advances in applied and computational topology*, 70:1–39, 2012, 1411.0613.

[113] David A. Randell and Anthony G. Cohn. Exploiting lattices in a theory of space and time. *Computers and Mathematics with Applications*, 23(6-9):459–476, 1992.

[114] Anthony G Cohn, David A Randell, and Zhan Cui. {T}axonomies of logically defined qualitative spatial relations. *Int. J. Hum.-Comput. St.*, 43(5–6):831–846, 1995.

[115] Rachel Myrow. *UPS Takes Left Turn Out of Deliveries*. 2007.

[116] Ekkehard Köhler, Ekkehard Köhler, Katharina Langkau, Katharina Langkau, Martin Skutella, and Martin Skutella. {*Time-Expanded*} *Graphs for* {*Flow-Dependent*} *Transit Times*, pages 49–56. Number 03 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, sep 2002.

[117] Chapter Description and Chapter Objectives. *Introduction to Geographic Information Systems*. McGraw-Hill Higher Education Boston, 2005.

[118] Kenneth J Dueker Ric and Vrana. Dynamic Segmentation Revisited: A Milepoint Linear Data Model. *GIS-T '92 Conference*, 4(2):1–14, 1984.

[119] Paul Abrahams. President's letter. *Communications of the ACM*, 30(6):472–473, jun 1987.

[120] T Cormen, R Rivest, and C Leiserson. *Introduction to Algorithms*. MIT press, 1989.

[121] David S Johnson and C.H. Papadimitriou. *Computational Complexity*. John Wiley and Sons Ltd., 1985.

[122] Ramez Elmasri and Shamkant B Navathe. *Fundamentals of Database Systems*, volume 28. Pearson, 7 edition edition, jun 2003, arXiv:1011.1669v3.

[123] E F Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the Association for Computing Machinery*, 13(6):377–387, 1970, arXiv:1011.1669v3.

[124] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. *The Google file system*, volume 37, page 29. ACM, 2003, z0024.

[125] D Borthakur. The hadoop distributed file system: Architecture and design, Hadoop Project Website,. *Hadoop Project Website*, 11(2007):21, 2007.

[126] J Dean and S Ghemawat. Simplified data processing on large clusters. *Sixth Symp. Oper. Syst. Des. Implement.*, 51(1):107–113, 2004, 10.1.1.163.5292.

[127] Apache spark - lightning-fast cluster computing. https://spark.apache.org. Accessed: 3/1/2017.

[128] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[129] Heikki Mannila. *Mining Frequent Patterns without Candidate Generation A Frequent-Pattern Tree Approach.pdf*, volume 29, pages 53–87. ACM, 2004.

[130] Open geo spatial portal. http://www.opengeospatial.org/. Accessed: 3/1/2017.

[131] Y.-K.a Whang K.-Y.a Song I.L.-Y.c Lee J.-H.a b Lee. A Physical Database Design Method for Multidimensional File Organizations. *Information Sciences*, 102(1-4):31–65, 1997.

[132] Wolfgang Kainz and Wolfgang Kainz. *The Design and Analysis of Spatial Data Structures*, volume 5. Addison-Wesley Reading, MA, 1991.

[133] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984, ISBN 0-89791-128-8.

[134] Shashi Shekhar, Sanjay Chawla, Siva Ravada, Andrew Fetterer, Xuan Liu, and Chang Tien Lu. Spatial databases accomplishments and research needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):45–55, 1999.

[135] Michael R Evans, Dev Oliver, Kwangsoo Yang, Zhou Xun, and Shekhar Shashi. Enabling Spatial Big Data via CyberGIS: Challenges and Opportunities. *CyberGIS: Fostering a New Wave of Geospatial Innovation and Discovery*, d(Table 1):1–16, 2014.

[136] Eldawy Ali and Mohammad Mokbel. SpatialHadoop.

[137] Ablimit Aji, Fusheng Wang, Hoang Vo, Rubao Lee, Qiaoling Liu, Xiaodong Zhang, and Joel Saltz. Hadoop GIS. *Proceedings of the VLDB Endowment*, 6(11):1009–1020, aug 2013.

[138] Jia Yu, Wu Jinxuan, and Sarwat Mohamed. *GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data*, pages 4–7. Number 3. ACM, 2015.

[139] S K Prasad, S Shekhar, M McDermott, X Zhou, M Evans, and S Puri. *GPGPU-accelerated interesting interval discovery and other computations on GeoSpatial datasets - A summary of results*, page 7. BigSpatial '13. ACM, 2013.

[140] S. Shekar and S. Chawla. *Spatial Databases: A Tour*. Prentice Hall, 2003.

[141] Emre Eftelioglu, Shashi Shekhar, Dev Oliver, Xun Zhou, Michael R. Evans, Yiqun Xie, James M. Kang, Renee Laubscher, and Christopher Farah. Ring-Shaped Hotspot Detection: A Summary of Results. In *2014 IEEE International Conference on Data Mining*, pages 815–820, 2014.

[142] Emre Eftelioglu, Shashi Shekhar, James M Kang, and Christopher C Farah. Ring-shaped hotspot detection. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3367–3381, 2016.

[143] James P Rogers and James A Shine. *Sustained Emerging Spatio-Temporal Co-occurrence Pattern Mining : A Summary of Results*, pages 106–115. IEEE, 2006.

[144] Mete Celik, Shashi Shekhar, James P. Rogers, James A. Shine, and Jin Soung Yoo. *Mixed-drove spatio-temporal co-occurrence pattern mining: A summary of results*, pages 119–128. dec 2006.

[145] Yan Huang, Shashi Shekhar, and H. Xiong. Discovering colocation patterns from spatial data sets: a general approach. *Tkde*, 16(12):1472–1485, 2004.

[146] Pradeep Mohan, Shashi Shekhar, James A Shine, and James P Rogers. Cascading spatio-temporal pattern discovery: A summary of results. In *SDM*, pages 327–338. SIAM, 2010.

[147] P. Mohan, S. Shekhar, et al. Cascading spatio-temporal pattern discovery. *Knowledge and Data Engineering, IEEE Transactions on*, 24(11):1977–1992, 2012.

[148] Michael R Evans, Dev Oliver, Shashi Shekhar, and Francis Harvey. *Summarizing trajectories into k-primary corridors: a summary of results*, pages 454–457. SIGSPATIAL '12. ACM, 2012.

[149] Dev Oliver, Abdussalam Bannur, James M Kang, Shashi Shekhar, and Renee Bousselaire. A k-main routes approach to spatial network activity summarization: A summary of results. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 265–272. IEEE, 2010.

[150] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, and Shijian Li. *Real-time detection of anomalous taxi trajectories from GPS traces*, volume 104 LNICST of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 63–74. Springer Berlin Heidelberg, dec 2012.

[151] Zhouyu Fu, Weiming Hu, and Tieniu Tan. *Similarity Based Vehicle Trajectory Clustering and Anomaly Detection* , volume 2, page 8. IEEE, 2005.

[152] Zhenhui Li, Ming Ji, Jae-gil Lee, Lu-an Tang, Yintao Yu, Jiawei Han, and Roland Kays. *MoveMine : Mining Moving Object Databases*, pages 1203–1206. ACM, 2010.

[153] Wei Liu. *Discovering Spatio-Temporal Causal Interactions in Traffic Data Streams*, pages 1010–1018. ACM, 2011.

[154] Wanli Min, Laura Wynter, and Yasuo Amemiya. Road Traffic Prediction with Spatio-Temporal Correlations Road Traffic Prediction with Spatio-Temporal Correlations. *IBM Research Report*, 24275(4):RC24275 (W0706–018) June, 2007.

[155] Dimitris Sacharidis, Kostas Patroumpas, Manolis Terrovitis, Verena Kantere, Michalis Potamias, Kyriakos Mouratidis, and Timos Sellis. *On-line discovery of hot motion paths*, pages 392–403. ACM, 2008.

[156] Jung-im Won, Sang-wook Kim, Ji-haeng Baek, and Junghoon Lee. *Trajectory Clustering in Road Network Environment (09).pdf*, pages 299–305. IEEE, 2009.

[157] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. *Driving with knowledge from the physical world*, pages 316–324. Number 5. ACM, 2011.

[158] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. *iBAT*, page 99. UbiComp '11. ACM, 2011.

[159] Yu Zheng. *Compute with spatial trajectories*. Springer Science & Business Media, 2011.

[160] S Shekhar, C T Lu, and P Zhang. *Detecting Graph-Based Spatial Outliers: Algorithms and Applications*, pages 371–376. ACM, 2001.

[161] S Shekhar, C Lu, and P Zhang. A unified approach to detecting statial outliers. *GeoInformatica*, 7(2):139–166, 2003.

[162] Zhe Jiang, Shashi Shekhar, Pradeep Mohan, Joseph Knight, and Jennifer Corcoran. *Learning spatial decision tree for geographical classification: a summary of results*, pages 390–393. ACM, 2012.

[163] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. Focal-test-based spatial decision tree learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1547–1559, jun 2015.

[164] Xun Zhou et al. Spatiotemporal change footprint pattern discovery: an inter-disciplinary survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(1):1–23, 2014.

[165] Richard S Ostfeld, Gregory E Glass, and Felicia Keesing. Spatial epidemiology: an emerging (or re-emerging) discipline. *Trends in ecology & evolution*, 20(6):328–336, 2005.

[166] Steven Le Comber et al. Geographic profiling as a novel spatial tool for targeting infectious disease control. *International journal of health geographics*, 10(1):1–8, 2011.

[167] Nigel E Raine, D Kim Rossmo, and Steven C Le Comber. Geographic profiling applied to testing models of bumble-bee foraging. *Journal of the Royal Society Interface*, 6(32):307–319, 2009.

[168] Steven C Le Comber, Barry Nicholls, D Kim Rossmo, and Paul A Racey. Geographic profiling and animal foraging. *Journal of Theoretical Biology*, 240(2):233–240, 2006.

[169] Alan R Putnam and William B Duke. Allelopathy in agroecosystems. *Annual review of phytopathology*, 16(1):431–451, 1978.

[170] Wenche E Dramstad. Do bumblebees (hymenoptera: Apidae) really forage close to their nests? *Journal of Insect Behavior*, 9(2):163–182, 1996.

[171] Mark D Stevenson, D Kim Rossmo, Robert J Knell, and Steven C Le Comber. Geographic profiling as a novel spatial tool for targeting the control of invasive species. *Ecography*, 35(8):704–715, 2012.

[172] David MH Baker, James W Head, Caleb I Fassett, Seth J Kadish, Dave E Smith, Maria T Zuber, and Gregory A Neumann. The transition from complex crater to peak-ring basin on the moon: New observations from the lunar orbiter laser altimeter (lola) instrument. *Icarus*, 214(2):377–393, 2011.

[173] Bevan M French. Traces of catastrophe: A handbook of shock-metamorphic effects in terrestrial meteorite impact structures. 1998.

[174] Deepak Agarwal et al. Spatial scan statistics: approximations and performance study. In *Proceedings of the 12th ACM SIGKDD international conference on KDD*, pages 24–33. ACM, 2006.

[175] Daniel B Neill and Andrew W Moore. Rapid detection of significant spatial clusters. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 256–265. ACM, 2004.

[176] James Munkres. *Topology (2nd Edition)*. 2 edition, January 2000.

[177] Martin Kulldorf. *Spatial Scan Statistics: Models, Calculations and Applications*, pages 303–322. Springer, 1999.

[178] Martin Kulldorff. A spatial scan statistic. *Communications in Statistics-Theory and methods*, 26:1481–1496, 1997.

[179] Katarina Fritzon. An examination of the relationship between distance travelled and motivational aspects of firesetting behaviour. *Journal of Environmental Psychology*, 21(1):45–60, 2001.

[180] Kim Dralle and Mats Rudemo. Stem number estimation by kernel smoothing of aerial photos. *Canadian Journal of Forest Research*, 26(7):1228–1236, 1996.

[181] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. *Computational geometry*. Springer, 2000.

[182] Rudolf Gorenflo and Francesco Mainardi. Random walk models for space-fractional diffusion processes. *Fract. Calc. Appl. Analysis*, 1(2):167–191, 1998.

[183] Rakesh Agrawal et al. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[184] Gerald F Pyle. *Applied medical geography*. VH Winstond and Sons, 1979.

[185] Ray M Merrill. *Introduction to epidemiology*. Jones & Bartlett Publishers, 2013.

[186] Jacklin F Mosha et al. Epidemiology of subpatent plasmodium falciparum infection: implications for detection of hotspots with imperfect diagnostics. *Malar J*, 12(221):10–1186, 2013.

[187] Alexandra Sifferlin. The 5 Biggest Mistakes in the Ebola Outbreak. *Time Magazine, http://time.com/3426642/the-5-biggest-mistakes-in-the-ebola-outbreak-so-far/*, Sept 25 2014.

[188] Leslie W Kennedy and David R Forde. Routine activities and crime: An analysis of victimization in canada*. *Criminology*, 28(1):137–152, 1990.

[189] Lyndsay N. Boggess, Robert T Greenbaum, and George E Tita. Does crime drive housing sales? evidence from los angeles. *Journal of Crime and Justice*, 36(3):299–318, 2013.

[190] Dong-Wan Choi et al. A scalable algorithm for maximizing range sum in spatial databases. *Proceedings of the VLDB Endowment*, 5(11):1088–1099, 2012.

[191] Daniel B Neill. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting*, 25(3):498–517, 2009.

[192] Daniel B Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):337–360, 2012.

[193] Kurt Binder. *Introduction: Theory and technical aspects of Monte Carlo simulations*. Springer, 1986.

[194] Emo Welzl. *Smallest enclosing disks (balls and ellipsoids)*. Springer, 1991.

[195] Open street map web site. `http://www.openstreetmap.org/`.

[196] Daniel Neill. An empirical comparison of spatial scan statistics for outbreak detection. *International Journal of Health Geographics*, 8(1):20, 2009.

[197] Daniel B Neill. *Detection of spatial and spatio-temporal clusters*. PhD thesis, University of South Carolina, 2006.

[198] J Fitterer, TA Nelson, and F Nathoo. Predictive crime mapping. *Police Practice and Research*, 16(2):121–135, 2015.

[199] Emre Eftelioglu, Xun Tang, and Shashi Shekhar. Geographically robust hotspot detection: A summary of results. In *ICDM International Workshop on Spatial and Spatiotemporal Data Mining (SSTDM)*, 2015.

[200] Kulldorff et al. An elliptic spatial scan statistic. *Statistics in medicine*, 25(22):3929–3943, 2006.

[201] Xun Tang et al. Elliptical hotspot detection: A summary of results. In *ACM SIGSPATIAL Workshops*, 2015.

[202] Daniel B Neill and Andrew W Moore. A fast multi-resolution method for detection of significant spatial disease clusters. In *Advances in Neural Information Processing Systems*, page None, 2003.

[203] Tony H Grubesic, Ran Wei, and Alan T Murray. Spatial clustering overview and comparison: Accuracy, sensitivity, and computational expense. *Annals of the Association of American Geographers*, 104(6):1134–1156, 2014.

[204] Daniel JK Beavon, Patricia L Brantingham, and Paul J Brantingham. The influence of street networks on the patterning of property offenses. *Crime prevention studies*, 2:115–148, 1994.

[205] Jane Law, Matthew Quick, and Ping Chan. Bayesian spatio-temporal modeling for analysing local patterns of crime over time at the small-area level. *Journal of quantitative criminology*, 30(1):57–78, 2014.

[206] Atsuyuki Okabe, Kei-Ichi Okunuki, and Shino Shiode. The sanet toolbox: new methods for network spatial analysis. *Transactions in GIS*, 10(4):535–550, 2006.

[207] Atsuyuki Okabe and Kokichi Sugihara. *Spatial analysis along networks: statistical and computational methods*. John Wiley & Sons, 2012.

[208] Shino Shiode and Narushige Shiode. Network-based space-time search-window technique for hotspot detection of street-level crime incidents. *International Journal of Geographical Information Science*, 27(5):866–882, 2013.

[209] Lei Shi and Vandana P Janeja. Anomalous window discovery for linear intersecting paths. *Knowledge and Data Engineering, IEEE Transactions on*, 23(12):1857–1871, 2011.

[210] Marcelo Azevedo Costa, Renato Martins Assunção, and Martin Kulldorff. Constrained spanning tree algorithms for irregularly-shaped spatial clustering. *Computational Statistics & Data Analysis*, 56(6):1771–1783, 2012.

[211] Kazimierz Kuratowski. *Topology*, volume 1. Elsevier, 2014.

[212] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[213] Us census bureau tiger/line shapefiles, Accessed: Dec 9, 2015.

[214] City of chicago data portal. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2. Accessed: 12/1/2014.

[215] City of oakland data portal. https://data.oaklandnet.com. Accessed: 1/5/2016.

[216] Dev Oliver et al. A k-main routes approach to spatial network activity summarization. *Knowledge and Data Engineering, IEEE Transactions on*, 26, 2014.

[217] Diansheng Guo. Local entropy map: A nonparametric approach to detecting spatially varying multivariate relationships. *International Journal of Geographical Information Science*, 24(9):1367–1389, 2010.

[218] Mary K Wolfe and Jeremy Mennis. Does vegetation encourage or suppress urban crime? evidence from philadelphia, pa. *Landscape and Urban Planning*, 2012.

[219] Alex Hirschfield et al. How places influence crime: The impact of surrounding areas on neighbourhood burglary rates in a british city. *Urban Studies*, 2013.