Modeling Spatial and Spatio-temporal Co-occurrence Patterns

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

Mete Celik

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Shashi Shekhar, Advisors

July 2008
Acknowledgements

I wish to thank several people who, in some way, have helped me during the development of this thesis.

First, I would like to express my gratitude to my supervisor, Prof. Shashi Shekhar for his continued encouragement and invaluable suggestions during this study. It has been a great pleasure to do research with him.

I would like to thank members of my advisory committee Professors Jaideep Srivastava, Arindam Banerjee, and Sudipto Banerjee for their guidance and constructive and useful comments. This research also benefited from discussions with Prof. Daniel Boley.

I sincerely would like to thank to US Army Corps of Engineers, Topographic Engineering Center Researchers James P. Rogers and Dr. James A. Shine with whom I have collaborated during my PhD research. Their conceptual and technical insights into my thesis work have been invaluable.

I am very grateful to the all members of Spatial Databases and Data Mining group members at the Department of Computer Science for their discussions, critiques, and contributions. There are three people I need to mention especially: Betsy George who helped me unselfishly whenever I needed, and Jin Soung Yoo and Baris Kazar who provided assistance during the early stages of this research. I apologize to all I have forgotten, among whom are the anonymous referees of my papers. I assure them that their help was valued as much as that of the people mentioned above.

During the course of this work, at the University of Minnesota, I was supported in part by the Turkish Council of Higher Education and with grants from the US Army Corps of Engineers, Topographic Engineering Center. I am grateful for their support.

Many people on the staff of the Computer Science Department assisted and encouraged me in various ways during my course of studies. I am grateful to Georganne Tolaas, Bonnie Klein, and Jennifer Barrett. I also would like to thank to Kim Koffolt for her comments to improve the readability of my papers and this dissertation.

Lastly, and most importantly, I wish to thank my parents, Fatma Celik and Metin Celik.
for everything they gave me. I am very grateful for my wife Filiz Dadaser Celik, for her love, patience, and support during the PhD period. One of the best experiences that we lived through in this period was the birth of our children Cagri Fatih and Elif Huma.
Abstract

As the volume of spatial and spatio-temporal data continues to increase significantly due to both the growth of database archives and the increasing number and resolution of spatio-temporal sensors, automated and semi-automated pattern analysis becomes more essential. Spatial and spatio-temporal (ST) data analyses have emerged in recent decades to develop understanding of the spatial and spatio-temporal characteristics and patterns. However, in the last decade, the growth in variety and volume of observational data, notably spatial and spatio-temporal data, has out-paced the capabilities of analytical tools and techniques.

Major limitations of existing classical data mining models and techniques include the following. First, these do not adequately model richer temporal semantics of data observations (e.g. co-occurrence patterns of moving objects, emerging and vanishing patterns, multi-scale cascade patterns, periodic patterns). Second, these do not take into account time dimension of the data observations. Third, these do not provide sufficient interest measures and computationally efficient algorithms to discover spatial and spatio-temporal co-occurrence patterns. These limitations represent critical barriers in several application domains that require to analyze huge datasets.

In this dissertation, I proposed addressed these limitations by i) providing a framework to model the rich semantics of the ST patterns of data observations by developing a taxonomy of spatial and ST co-occurrence patterns, ii) designing new techniques that are taking into account the time dimension of the data, and iii) developing new monotonic composite interest measures and scalable algorithms. The proposed approaches reduced the manual effort by reducing the plausible set of hypotheses. Major focus would be on developing scalable algorithms to mine spatial and ST co-occurrence patterns.
Contents

List of Figures vii

List of Tables ix

1 Introduction 1

1.1 Challenges .......................................................... 2
1.2 Innovation ............................................................ 3
1.3 Research Methodology .............................................. 4
1.4 Contributions ....................................................... 5

2 Zonal Co-location Pattern Discovery with Dynamic Parameters 10

2.1 Introduction .......................................................... 11
2.1.1 Contributions ..................................................... 12
2.1.2 Outline ............................................................. 12
2.2 Related Work ........................................................ 12
2.3 Basic Concepts and Problem Statement .......................... 14
2.3.1 Basic Concepts ..................................................... 15
2.3.2 Modeling Zonal Co-location Patterns .......................... 16
2.3.3 Problem definition ............................................... 17
2.4 Mining Zonal Co-location Patterns ............................... 18
2.4.1 Naïve Approach .................................................... 20
2.4.2 clQuad-Tree Bulk Load Algorithm ............................ 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.3</td>
<td>Zoloc-Miner Algorithm</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>Experimental Evaluation</td>
<td>25</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Experimental Setup</td>
<td>25</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Experiment Results</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Conclusions and Future Work</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>Mixed-Drove Spatio-Temporal Co-occurrence Pattern Mining</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Contributions</td>
<td>33</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Scope and Outline</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Related Work</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Basic Concepts and Problem Definition</td>
<td>36</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Spatial Prevalence Measure</td>
<td>36</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Modeling MDCOPs</td>
<td>37</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Problem statement</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>Mining MDCOPs</td>
<td>40</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Naïve approach</td>
<td>41</td>
</tr>
<tr>
<td>3.4.2</td>
<td>MDCOP-Miner</td>
<td>42</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Modified MDCOP-Miner (FastMDCOP-Miner)</td>
<td>46</td>
</tr>
<tr>
<td>3.5</td>
<td>Analysis of the MDCOP-Miner</td>
<td>48</td>
</tr>
<tr>
<td>3.5.1</td>
<td>The Mixed-Drove Prevalence Index Measure is Monotonic</td>
<td>49</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Correctness and Completeness</td>
<td>49</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Algebraic Cost Model</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Experimental Evaluation</td>
<td>54</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Datasets</td>
<td>55</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Experiment Results for Real Datasets</td>
<td>56</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Experiment Results for Synthetic Datasets</td>
<td>60</td>
</tr>
<tr>
<td>3.7</td>
<td>Conclusions and Future Work</td>
<td>66</td>
</tr>
</tbody>
</table>
4 Sustained Emerging Spatio-Temporal Co-occurrence Pattern Mining: A Summary of Results 68

4.1 Introduction ...................................................... 69
  4.1.1 An Application Domain Example ...................... 69

4.2 Related Work .................................................... 72
  4.2.1 Scope and Outline .......................................... 73

4.3 Basic concepts and problem statement ....................... 73
  4.3.1 Spatial prevalence measure ............................... 73
  4.3.2 Modeling SECOPs ........................................ 75
  4.3.3 Problem statement ....................................... 76

4.4 Mining SECOPs .................................................. 77
  4.4.1 Naïve Approach .......................................... 77
  4.4.2 SECOP-Miner ............................................ 78

4.5 Analysis of the SECOP mining algorithm ..................... 82
  4.5.1 Sustained emergence prevalence measure is monotonic .. 82
  4.5.2 Correctness and completeness ............................ 84

4.6 Experimental evaluation ....................................... 85
  4.6.1 Effect of number of time slots ............................ 87
  4.6.2 Effect of number of object-types ........................ 87
  4.6.3 Effect of time prevalence index threshold ............... 88
  4.6.4 Effect of spatial prevalence index threshold ............. 88

4.7 Conclusions and future work ................................ 89

5 Conclusions and Future Work .................................. 91
  5.1 Future Work .................................................. 92

Bibliography ......................................................... 94
List of Figures

1.1 Phases of the ST co-occurrence pattern mining ........................................ 4

2.1 Example dataset ...................................................................................... 14

2.2 An example dataset ................................................................................ 15

2.3 Example dataset for execution trace ....................................................... 22

2.4 Experimental setup and design ............................................................... 26

2.5 Effect of Number of Zones .................................................................. 27

2.6 Effect of Size of Zone ........................................................................... 27

2.7 Effect of Zonal Overlap ....................................................................... 28

3.1 An example Hail Mary play in American football in 4 time slots ......... 32

3.2 An example spatio-temporal dataset ..................................................... 38

3.3 Comparison of Naïve Approach, MDCOP-Miner, and FastMDCOP-Miner .......................................................... 41

3.4 Execution trace of the MDCOP-Miner algorithm .............................. 46

3.5 Execution trace of the FastMDCOP-Miner algorithm ....................... 48

3.6 Experimental setup and design ............................................................ 55

3.7 Effect of number of time slots in MDCOP mining algorithms using real dataset .............................................................................. 57

3.8 Effect of number of object-types in MDCOP mining algorithms using real dataset .............................................................................. 58
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>Effect of the time prevalence index threshold in MDCOP mining algorithms using real dataset</td>
<td>59</td>
</tr>
<tr>
<td>3.10</td>
<td>Effect of the spatial prevalence index threshold in MDCOP mining algorithms using real dataset</td>
<td>61</td>
</tr>
<tr>
<td>3.11</td>
<td>Effect of number of time slots in MDCOP mining algorithms using synthetic dataset</td>
<td>62</td>
</tr>
<tr>
<td>3.12</td>
<td>Effect of number of object-types in MDCOP mining algorithms using synthetic dataset</td>
<td>63</td>
</tr>
<tr>
<td>3.13</td>
<td>Effect of the time prevalence index threshold in MDCOP mining algorithms using synthetic dataset</td>
<td>64</td>
</tr>
<tr>
<td>3.14</td>
<td>Effect of the spatial prevalence index threshold in MDCOP mining algorithms using synthetic dataset</td>
<td>65</td>
</tr>
<tr>
<td>3.15</td>
<td>Effect of noise and average number of instances in MDCOP mining algorithms using synthetic dataset</td>
<td>66</td>
</tr>
<tr>
<td>4.1</td>
<td>Examples of EIDs. Adapted from [17]. Black circles represent newly emerging diseases and white circles represent re-emerging diseases</td>
<td>70</td>
</tr>
<tr>
<td>4.2</td>
<td>Weekly dengue fever cases in 2005 in Singapore [1]</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>An example spatio-temporal dataset</td>
<td>74</td>
</tr>
<tr>
<td>4.4</td>
<td>An execution trace of mining sustained emerging spatio-temporal co-occurrence patterns (SECOPs)</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Experimental setup and design</td>
<td>85</td>
</tr>
<tr>
<td>4.6</td>
<td>One instance of a SECOP</td>
<td>86</td>
</tr>
<tr>
<td>4.7</td>
<td>Statistical summary of participation index of SECOP</td>
<td>86</td>
</tr>
<tr>
<td>4.8</td>
<td>Effect of number of time slots</td>
<td>87</td>
</tr>
<tr>
<td>4.9</td>
<td>Effect of number of object-types</td>
<td>88</td>
</tr>
<tr>
<td>4.10</td>
<td>Effect of time prevalence index threshold</td>
<td>89</td>
</tr>
<tr>
<td>4.11</td>
<td>Effect of spatial prevalence index threshold</td>
<td>89</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Global and Zonal Co-locations</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Normal and buffer points of zones</td>
<td>22</td>
</tr>
<tr>
<td>2.3</td>
<td>Zonal co-location patterns</td>
<td>23</td>
</tr>
<tr>
<td>2.4</td>
<td>Co-location patterns of the root</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Synthetic Dataset Generation Parameters</td>
<td>56</td>
</tr>
</tbody>
</table>
Introduction

Traditional data mining methods (e.g. association rules, classifications, and clustering) often assume that data observations have the same properties regardless of their spatial location. This violates Tobler’s First Law of Geography: everything is related to everything else, but nearby objects in space are more related than distant objects [45, 43]. The same principle also applies to objects near or far in time from one another (as shown in time series modeling). As a result, the values of attributes of neighboring spatial and spatio-temporal (ST) data objects tend to affect each other.

The complexity of spatial and spatio-temporal data and intrinsic spatial relationships limit the usefulness of classical techniques for extracting spatial and spatio-temporal patterns. Specifically, classical techniques do not work well when the data is characterized by i) extended spatial objects (e.g. points, lines, and polygons), ii) implicit spatial relationships among the variables, iii) observations that are not independent, and iv) spatial and temporal autocorrelation. The fact that spatial and spatio-temporal data are not transactional further limits the application of classical techniques. New models, objective functions, and patterns more suited for ST databases and their unique properties are needed for knowledge discovery from these databases.

The focus of this dissertation is to create and explore new models of ST co-occurrence patterns. Formally, given a collection of Boolean (binary) ST features and their instances, these new models identify the subsets of features frequently located together in space and
1.1 Challenges

Adapting classical data mining methods to mine ST patterns is far from trivial. For example, co-occurrence patterns often look similar to association patterns [26], which identify subsets of item-types that co-occur frequently in a given collection of transactions, each specifying a subset of item-types. However, instances of ST features are embedded in continuous space and time, and share a variety of ST relationships. Conceptual modeling of these patterns is challenging due to the absence of pre-defined transactions in many datasets and the unique interest measures. Using association rule mining for ST data requires a transaction database which is not natural [45, 28], because the transaction boundaries may split co-occurrence pattern instances across distinct transactions, for example, those defined by cells of a rectangular grid.

There is no efficient technique to mine spatial patterns with dynamic parameters (i.e., repeated specification of zone and interest measure values according to user preferences) from huge spatial datasets. Previous methods in spatial co-location pattern mining suffer due to the lack of an indexing structure and the subsequent need for repeated specification of the parameter values according to the user preferences [43]. This dissertation explores efficient methods and designs indexing structures to support repetitive specifications of the parameters according to user preferences.

No standard taxonomy for spatial and ST co-occurrence patterns exists in the current literature. The challenge in creating this taxonomy is to find categories of interesting and nontrivial patterns. This dissertation explores and defines new spatial and ST co-occurrence patterns using concepts such as periodicity, moving objects, emergence and disappearance. Current taxonomy of spatial data deals with the patterns/objects, which are fixed at a time. These patterns/objects are associated with geometry and position in space. OGIS [3] de-
defines a taxonomy for spatial data and provides a framework for exchanging spatial data. In contrast to this, there is no accepted framework for taxonomy of ST data, which deals with the patterns in space and time and captures relationships between a pattern and space-time.

Another problem and challenge is the incorporation of temporal data. Current co-occurrence patterns have been defined for spatial data features, but not for ST data types. Also, these defined spatial data feature patterns assume stationarity in space and not for nonstationary features such as are frequently found in ST data.

The most challenging technical barrier is to create and formalize new interest measures to mine interesting and non-trivial ST co-occurrence patterns. Current scalar interest measures are not sufficient to mine interesting and non-trivial ST co-occurrence patterns.

There are also no existing methods to mine ST co-occurrence patterns out of massive ST datasets in a computationally efficient manner. In terms of data size, ST datasets will be larger than the classical datasets because of addition of the time dimension. To handle these massive datasets the challenge is to explore efficient and scalable methods.

1.2 Innovation

This dissertation creates new models for spatial and ST datasets with a new taxonomy of ST co-occurrence patterns and interest measures. In contrast to current scalar interest measures, new composite interest measures are designed to characterize interesting and useful ST co-occurrence patterns. These new interest measures such as temporal, spatial, and spatio-temporal probabilities of co-occurrence patterns are used to characterize where and when a co-occurrence is prevalent. Also, while current interest measures are scalars, there is a need for new, non-scalar types of patterns and composite interest measures that are functions of time, location, or other parameters. Further, scalar interest measure for a given feature subset may be periodic or have a trend (e.g. increase or decrease over time) defining periodic, emerging, and vanishing co-occurrence patterns. Composite interest measures, e.g. spatial map of locations of co-occurrence instances, may show complete spatial randomness [14], hot-spots or regularity. In addition, the hot-spots may be emerging
(or vanishing) over time. These give rise to emerging or vanishing co-occurrence patterns. Feature sets may include static or moving objects leading to additional classes.

In recent years, many studies have focused on finding spatial co-location [28] patterns, which are subsets of features whose instances are frequently located together in geographic space. However, traditional co-location patterns are based purely on geographic proximity and do not account for temporal relationships. For example, they cannot differentiate between emerging ST co-occurrence patterns and vanishing ones.

1.3 Research Methodology

The strategy of this research can be depicted as shown in Figure 1.1, with the boxes showing the process phases, the arrows representing flow direction, and the loop outlining the iterative part of the ST co-occurrence pattern mining process. The phases are described as follows:

![Figure 1.1. Phases of the ST co-occurrence pattern mining](image)

The phases of the ST co-occurrence pattern mining process:

1. Create and Define ST Pattern Taxonomy: In this stage, based on the application domain, data mining problems are outlined, initial ST data are collected, characteristics of this data are examined, data quality is verified, and initial hypotheses are explored.

2. Create and Define Conceptual Model of ST Co-occurrence Patterns: A conceptual model of ST co-occurrence patterns are defined and initial spatial and ST raw data are prepared as an input for the mining method. The model helps define concepts and expand the taxonomy for spatial and ST co-occurrence patterns.
3. Create and Define Composite Interest Measures: These measures quantify new composite definitions such as temporal, spatial, and ST probabilities of ST data. These measures capture the characteristics of spatial ST datasets.

4. Create and Design Computationally Efficient Methods: Existing data mining methods are examined and new computationally efficient methods for mining ST co-occurrence patterns are created and tested.

5. Mine Patterns: Run the new methods to mine the ST co-occurrence patterns, defined by ST taxonomy, from the ST datasets.

6. Validate Patterns: Accuracy and completeness of the newly discovered spatial and ST co-occurrence patterns are checked, via user evaluation.

1.4 Contributions

This dissertation focuses on discovering interesting and non-trivial spatial ST co-occurrence patterns by creating new methods. In this dissertation, I define a new taxonomy of the spatial and spatio-temporal co-occurrence patterns and then propose new models and define new patterns for some of the categories of this taxonomy.

A taxonomy of ST co-occurrence patterns provides a classification of the patterns listed as below. It is natural to ask if there are other interesting and useful classes of ST co-occurrence patterns. One approach to address this issue is to examine current application domains and application domain scientists and create a consensus classification scheme. Another approach is to use taxonomies for ST data types and their relationships, and study their implications for classes of ST data [18]; taxonomies of spatial data types may even be extended for this purpose. Object and field are two common models of spatial data [51]. An object model is ideal for representing discrete identifiable entities such as lakes, road networks, and cities. This model may be generalized to ST datasets by categorizing objects into stationary and mobile objects [18] as well as subclasses such as rigid and deforming. A field model is defined by a spatial framework (SF) and a set of field functions mapping the
SF to attribute value domains. This model may be generalized to ST datasets by defining a ST framework (STF) and field mapping the STF to attribute domains. STF fields may be categorized as largely static (e.g., elevation) or dynamic (e.g., temperature) for a given time scale [18].

I use a combination of these approaches by leveraging the work of application domain scientists in military terrain, ecology, climatology, and Earth science to create a new taxonomy for ST data [18,35]. It is also important to create a taxonomy of the common usages of ST patterns by domain scientists. Common activities include evaluation, explanation, and prediction, where scientists observe where and when the co-occurrence patterns are valid.

New types of spatial and ST co-occurrence patterns can be listed in five major categories:

1. Spatial co-location patterns: A subset of object-types that are frequently located together in space for a given set of feature-types, their instances, and a neighbor relation $R$ [28]. The aim of the spatial co-location pattern mining is to discover patterns from spatial datasets and it does not take into account the time information of the dataset. Spatial co-location patterns mainly can be divided into two categories such as global co-location pattern [28] and zonal co-location patterns.

2. Co-occurrence patterns of moving objects: In this kind of ST co-occurrence pattern, at least one of the co-occurring objects can be a moving object. On the other hand, all co-located objects can be moving objects.

3. Emerging or vanishing patterns: ST co-occurrence patterns whose interest measures are getting stronger or weaker with time (i.e., road locations or municipal areas where the incidence of attacks increase or decrease over a period of time).

4. Periodic co-occurrence patterns: If a new feature or a new instance of existing features is introduced to the space or extracted from the space, new co-occurrence patterns can appear or existing ones may disappear, or a new co-occurrence pattern may
reflect periodicity. Example pattern can be incidents that occur on Fridays but not on other weekdays.

5. Multi-scale ST cascade patterns: ST patterns that are prevalent in a particular neighborhood such as a block or a city may not be prevalent at a higher scale such as a state or a country. The same applies to ST patterns over time. Patterns that are prevalent on a particular day may not be prevalent over a week. Scale is an inherent property of ST data, and hence it presents a challenge to discover patterns at multiple levels of scale.

In this dissertation, I developed ST models and defined new patterns in the categories of the spatial co-locations, co-occurrence patterns of moving objects, and emerging or vanishing patterns of the taxonomy. I also defined new monotonic interest measures and designed new computationally efficient algorithms to mine these patterns.

In this dissertation, each chapter is written in the style of a separate paper, and each has been published in conferences and/or journals. An overview of the contents of these four chapters is as follows.

Chapter 2 focuses on mining Zonal Co-location Patterns. Co-location pattern discovery is the discovery of subsets of object-types that are frequently together in space [43]. I defined the problem of zonal co-location pattern discovery where zonal co-location patterns represent subsets of object-types that are frequently located in a subset of space (i.e., zone). Discovering zonal co-location patterns is an important problem in areas such as ecology, public health, and homeland defense. For example, in ecology, there may be several zonal co-location patterns, e.g., symbiotic relationships, predator-prey interactions. The association between the African crocodile and Egyptian Plover bird in which the bird will eat pieces of meat between a crocodile’s teeth is an example of symbiotic species. The interaction between a wolf and its diet of a variety of animals (e.g., elk, caribou, moose, rodent, bison, etc.) is an example of a predator-prey pattern. This predator-prey relationship may differ significantly from one place to another based on the availability of the prey. However, discovering these patterns with dynamic parameters (i.e., repeated specification of
zone and interest measure values according to user preferences) is computationally complex due to the repetitive mining process. Also, the set of candidate patterns is exponential in the number of feature types, and spatial datasets are huge. Previous methods in spatial co-location pattern mining suffer due to the lack of an indexing structure and the subsequent need for repeated specification of the parameter values according to the user preferences [43]. Furthermore, most methods assume that patterns are uniformly distributed over the space. This assumption violates the spatial heterogeneity law of Geography: “results of analysis may vary from one place to another” [19]. I proposed an indexing structure that stores co-location instances to discover zonal co-location patterns efficiently for repeated specifications of the parameter values, e.g., interest measure thresholds and zones, and experimentally evaluated the proposed algorithms. A summary of the preliminary work has been published in the *Seventh IEEE International Conference on Data Mining (ICDM 2007)* [10].

Chapter 3 focuses on mining Mixed-Drove Spatio-temporal Co-occurrence Patterns. I defined mixed-drove spatio-temporal patterns (MDCOPs) which represent subsets of different object-types whose instances are located close in geographic space for a significant fraction of time. Discovering MDCOPs is an important problem with many applications such as identifying tactics in battlefields, games (e.g., interactions between players of opponent teams), and ecology (e.g., predator-prey interactions). However, mining MDCOPs is computationally very expensive because the interest measures are computationally complex, datasets are larger due to the archival history, and the set of candidate patterns is exponential in the number of object-types. Previous studies either focus on discovery of uniform groups of moving objects (e.g., a sheep flock, a bird flock) without taking into account types of the objects in the discovery process, or they do not define the semantics for moving objects. I proposed a new composite interest measure and novel, computationally efficient algorithms for mining MDCOPs. The proposed methods have been evaluated analytically and experimentally. A summary of preliminary work has been published in the *Sixth IEEE International Conference on Data Mining (ICDM 2006)* [13], and the ex-
tended version of this work with new algorithm and analysis has been published in the *IEEE Transactions on Knowledge and Data Engineering (TKDE)* [12].

Chapter 4 focuses on mining **Sustained Emerging Spatio-temporal Co-occurrence Patterns**. I analyzed the trends of mixed groups of moving objects. I defined sustained emerging spatio-temporal co-occurrence patterns (SECOPs) that represent subsets of object-types that are increasingly located together in space and time. Discovering SECOPs is important for applications such as predicting emerging infectious disease outbreaks, and predicting defensive and offensive intent from troop movement patterns, etc. I proposed a monotonic interest measure for mining SECOPs and a novel SECOP mining algorithm. This work has been published in the *18th IEEE International Conference on Tools with Artificial Intelligence* as an invited paper [11].

Chapter 5 provides a summary of this study with future directions.
Zonal Co-location Pattern Discovery with Dynamic Parameters

Abstract

Zonal co-location patterns represent subsets of feature-types that are frequently located in a subset of space (i.e., zone). Discovering zonal spatial co-location patterns is an important problem with many applications in areas such as ecology, public health, and homeland defense. However, discovering these patterns with dynamic parameters (i.e., repeated specification of zone and interest measure values according to user preferences) is computationally complex due to the repetitive mining process. Also, the set of candidate patterns is exponential in the number of feature types, and spatial datasets are huge. Previous studies have focused on discovering global spatial co-location patterns with a fixed interest measure threshold. In this paper, we propose an indexing structure for co-location patterns and propose algorithms (Zoloc-Miner) to discover zonal co-location patterns efficiently for dynamic parameters. Extensive experimental evaluation shows our proposed approaches are scalable, efficient, and outperform naïve alternatives.
2.1 Introduction

Spatial data mining and spatial analysis techniques are playing a major role in spatial database systems to discover interesting but implicit patterns in spatial datasets of ever increasing size and complexity. Extracting useful and interesting patterns from massive spatial datasets is important for many application domains, including ecology (e.g., discovering the interactions between species and the interactions between vegetation types), public health (e.g., determining sources of pandemic diseases), and homeland defense (e.g., looking for significant or unusual “events”). Most data mining algorithms assume that patterns are uniformly distributed over the space. This assumption violates the spatial heterogeneity law of Geography: “results of analysis vary from one place to another” [20].

Given a collection of boolean spatial feature-types, their instances over a common spatial framework, a neighbor relation, and a subset of a spatial framework (i.e., zone), a zonal co-location pattern mining algorithm aims to discover correct and complete sets of interesting and non-trivial spatial co-location patterns while minimizing the computation cost. For example, in ecology, there may be several zonal co-location patterns, e.g., symbiotic relationship, predator-prey interactions. The association between crocodiles and birds where a bird will eat pieces of meat between a crocodile’s teeth is an example of symbiotic species. The African crocodile and Egyptian Plover in specific regions of Africa form such an association, i.e., a zonal co-location pattern. The interaction between a wolf and its diet of a variety of animals (e.g., elk, caribou, moose, rodent, bison, etc.) is an example of a predator-prey pattern. This predator-prey relationship may differ significantly from one place to another based on the availability of the prey.

However, mining zonal co-location patterns with dynamic parameters (i.e., repeated specification of zone and interest measure values according to user preferences) is challenging for several reasons. First, the repetitive mining process is computationally very expensive due to the dynamic set of parameters. Second, discovering patterns to support dynamic parameters is challenging due to lack of co-location indexing structures. Third, the set of candidate patterns increases exponentially with the number of feature-types. Fi-
nally, since spatial datasets are huge, computationally efficient and scalable algorithms are necessary.

2.1.1 Contributions

This chapter makes following contributions:

- It defines zonal co-location patterns and gives a formal definition of the zonal co-location pattern mining problem.
- It proposes a novel and computationally efficient zonal co-location pattern mining algorithm (Zoloc-Miner).
- It proposes an index structure for storing co-location patterns to handle dynamic parameters (i.e., changing user parameter specifications).
- It experimentally evaluates the proposed algorithms.

2.1.2 Outline

The rest of the paper is organized as follows. Section 2.3 presents basic concepts and defines the problem of mining zonal co-location patterns. Section 2.4 discusses the challenges of zonal co-location pattern discovery and presents proposed algorithm. The experimental evaluation is given in Section 2.5. I conclude and name directions for future work in Section 2.6.

2.2 Related Work

Previous research on spatial co-location pattern mining has focused on discovering global co-location patterns based on a fixed interest measure. Morimoto [37] used a support measure to discover frequently neighboring class sets. Huang et al. developed join-based [28] and Yoo et al. developed partial-join [53] and join-less [54] co-location algorithms using a fixed interest measure (i.e., spatial prevalence measure). Zhang et al. developed fast
2.2 Related Work

co-location mining algorithms using multi-way spatial joins [57]. Huang et al. proposed a projection based co-location pattern mining paradigm by extending the FP-tree structure [29] based on an interest measure parameter.

However, these approaches mainly focused on the discovery of global co-location patterns in a spatial dataset, due to the lack of user control on specifying zones of interest. The results of previous approaches may not represent or capture the characteristics of different zones inside the spatial dataset. Even if co-location patterns of the zone of interest are captured, their interest measure (i.e., spatial prevalence) values will be affected by the instances of the feature-types that are found in the spatial dataset as a whole, but which may not be found in the zone of interest because the spatial prevalence of the patterns may differ for different zones. Recently Ding et. al. defined problem of regional association rule mining problem and proposed methods discovering regional associations by identifying zones automatically [15]. In this approach, the main focus was identifying interesting subregions, e.g., zones, in spatial datasets for which regional association rules are then generated.

Overall, previous approaches do not provide adequate support for efficient pattern mining for changing user parameter specifications. These approaches might need to re-compute the patterns for each set of parameters. In contrast, I define the problem of zonal co-location pattern mining, propose an index structure based on Quad-trees [41] to support dynamic parameters, and develop computationally efficient methods to mine these patterns.

To illustrate the difference between global and zonal co-location patterns, I will use spatial dataset given in Figure 2.1. The dataset gives feature-types and their instances over the space. Each feature-type is represented by a distinct shape. The dataset has four zones (A, B, C, and D). These zones may represent topographic boundaries (e.g., four different districts in a city, etc.) or may represent the regions divided by an indexing approach (e.g., Quad-tree, R-tree, etc.) [41]. Each dashed circle shows the co-located instances that form a clique. Previous approaches will discover global patterns \{*, o\} and \{+, x\} for spatial prevalence index threshold 0.5 and neighbor distance 5, because they focus on the whole dataset (Table 2.1). However, the strength or existence of the global co-locations may vary
spatially for each zone. For example, in zone A, pattern \{+, x\} is frequently co-located with a stronger spatial prevalence measure. In zone B, patterns \{*, o\}, \{o, □\}, and \{o, *\} are frequently co-located. Patterns \{o, □\}, and \{o, *\} are not global co-locations, i.e., specific to zone B. In zone C and D, patterns \{+, x\} and \{*, o\} are frequently co-located zonal patterns respectively. Also, the prevalence measures of patterns \{+, x\} and \{*, o\} are increased (Table 2.1).

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Global co-locations</th>
<th>Zonal co-locations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Zone A</td>
<td>Zone B</td>
</tr>
<tr>
<td>+, x</td>
<td>0.786</td>
<td>0.615</td>
</tr>
<tr>
<td>* , o</td>
<td>0.667</td>
<td>0.667</td>
</tr>
</tbody>
</table>

2.3 Basic Concepts and Problem Statement

In this section, first I explain the modeling of co-location patterns in space and then, I explain how I model zonal co-location patterns with dynamic parameters.
2.3.1 Basic Concepts

Spatial co-location mining algorithms are used to discover subsets of feature-types that are frequently located together in space for a given set of feature-types, their instances, and a neighbor relation $R$ [28]. If instances form a clique for a given neighbor relation, they are co-located. For example, in Figure 2.2, $\{D.4, C.3\}$ is an instance of co-location pattern $\{D,C\}$, if the distance between the objects is no more than a given neighborhood distance threshold. In Figure 2.2, the solid lines show the distance between the objects that satisfies the neighborhood distance threshold.

![Figure 2.2. An example dataset](image)

A spatial prevalence measure, e.g., participation index (PI), is used to determine the strength of the co-location pattern, that is, whether the index is greater than or equal to a given threshold [28]. Such a co-location is called **prevalent**. PI is defined as the minimum of the **participation ratios** (the fraction of the number of instances of feature-types forming co-location instances to the total number of instances). For example, in Figure 2.2, $\{D, C\}$ is a co-location and its instances are $\{D.4, C.3\}$, $\{D.5, C.3\}$, and $\{D.6, C.4\}$. In the dataset, feature-type D has 6 instances and three of them (D.4, D.5, and D.6) are contributing to the co-location $\{D, C\}$, so the participation ratio of D is $3/6$. The participation ratio of
C is 2/4 since 2 out of 4 instances, i.e. C.3 and C.4, are contributing to the co-location pattern \{D, C\}. The participation index of the co-location pattern \{D, C\} is 0.5, which is the minimum of the participation ratios of object-types D and C. For more information, please refer to [28].

It has been shown that the participation index is anti-monotone in size of co-locations [28, 44]. In other words, \( \text{participation}_j(P_j) \leq \text{participation}_i(P_i) \) if \( P_i \) is a subset of \( P_j \). In addition, [28, 44] show that the participation index has a spatial statistical interpretation as an upper bound on the cross-K function [14].

### 2.3.2 Modeling Zonal Co-location Patterns

This section presents several relevant definitions to proposed approaches.

**Definition 2.3.1** Given a spatial framework, a set of spatial feature-types, their instances, and a neighbor relation \( R \), a **global co-location pattern** is a subset of feature-types whose instances are frequently neighbors in space.

**Definition 2.3.2** Given a spatial framework, a **zone** is a subset of the spatial framework.

Zones can be pre-defined, e.g., cities in a state or states in a country, etc. For example, in Figure 2.1(a), the dataset has four pre-defined zones, i.e., zones A, B, C, and D. Also, zones to be mined can be identified by the user.

**Definition 2.3.3** Given a zone, a set of spatial feature-types, their instances, and a neighbor relation \( R \), a **zonal co-location pattern** is a subset of feature-types whose instances are frequently neighbors in the given zone.

**Definition 2.3.4** Given a zone and a neighbor relation \( R \), a **buffer** is the surrounding area of the zone \( R \) distance.

For example, in Figure 2.2(a), the dotted line surrounding zone 3 shows the buffer of zone 3.
Definition 2.3.5 Given neighboring co-location instances and a zone with its buffer, a cross-neighbor is a co-location instance where at least one of the co-location features is in the buffer zone.

For example, in Figure 2.1, the co-location instance \{o, *\} is a cross-neighbor since feature o is in zone C and feature * is in the buffer of zone C.

Lemma 2.3.1 Given two zones S and T, if zone P is the union of zones S and T, i.e. \( P = S \cup T \), the co-location patterns of zone P are the union of the co-locations of zone S, zone T, the buffer of zone S, and buffer of zone T that are intersecting with zone P.

Proof: The co-location patterns and their instances of zones S and T will be generated from the spatial points where all the points in zones S and T respectively. The co-location patterns of the buffer of zone S will include co-location instances where at least one of the spatial points are in the buffer. The co-location patterns of the buffer of zone T will include co-location instances where at least one of the spatial points of instances are in the buffer. Buffers will capture the cross-neighbor co-located patterns. \(\square\)

For example in Figure 2.1, suppose zone P is the bottom half of the given dataset, that is, the union of zones C and D. In addition to the co-location pattern instances of zones C and D, zone P will include buffer instance \{o, *\}, which is a cross-neighbor.

2.3.3 Problem definition

Formally, given a set of spatial feature-types over a common spatial framework SF, a neighbor relation R, a minimum spatial prevalence threshold \( \theta \), and a subset of space (i.e. zone) Z, the aim of zonal co-location miner is to find correct and complete set of spatial co-locations satisfying spatial prevalence threshold within zone Z while minimizing the computation cost.

Given:

- A set of spatial feature types over a common spatial framework SF.
• A neighbor relation $R$ over locations.

• A minimum spatial prevalence threshold $\theta$.

• A subset of space, i.e., zone, $Z$.

**Find:** A set of spatial co-locations satisfying an interest measure threshold (i.e., participation index $\geq \theta$) and within zone $Z$.

**Objective:** Minimize computation cost.

**Constraints:** To find a correct and complete set of zonal spatial co-locations.

**Example:** In the dataset given in Figure 2.2(a) Boolean feature-types are A, B, C, and D. Each instance of the feature-types has unique identifiers, e.g., D.4. The straight lines between instances show the neighboring instances. The neighbor relation $R$ may be defined by a distance. For example, in Figure 2.2(a), D.4 is a neighbor of C.3. The dataset has 4 zones, i.e., 1, 2, 3, and 4. Co-location pattern $\{D, C\}$ is a zonal co-location pattern in zone 3 for a given spatial prevalence threshold 0.5.

### 2.4 Mining Zonal Co-location Patterns

The basic idea of zonal co-location pattern mining is to index space while storing relevant co-location patterns. It is possible to use any of the co-location mining approaches developed in the literature [28, 53, 54, 57]. However, known spatial indexing structures are not directly applicable for co-location pattern mining. In this study, I propose an index structure to mine zonal co-location patterns with dynamic parameters efficiently. Some of the major challenges of designing an index structure for zonal co-location patterns are:

(i) **Discovering cross-neighboring co-locations:** It is possible that co-located patterns are split and stored in different nodes of the index structure. This makes it hard to discover the co-location patterns of parent nodes, and may lead to separate patterns and is addressed
in the proposed index structure (Section 2.4.2) by adding a buffer (Definition 2.3.4) for each indexed zone. Index not only stores co-location instances of zone of interest but also cross-neighboring instances (Definition 2.3.5).

(ii) **Computational complexity:** As the number of points increases, the computational complexity of the co-location algorithm increases due to the generation of all possible candidates and is addressed by running the algorithm in the leaf nodes and their buffers of the tree (Section 2.4.2).

(iii) **Overlapping user-defined mining zones:** If a user-defined zone fits one of the nodes of the tree structure, the algorithm will output patterns of this zone, since patterns are stored in the tree structure. If a user-defined zone overlaps more than one node of the tree, the challenge is how to discover co-located patterns across multiple or partial nodes and is addressed in Section 2.4.3 based on Lemma 2.3.1.

A discussion of a naïve approach to mine zonal co-location patterns is given in Section 2.4.1. Then, I propose a novel zonal co-location pattern mining algorithm (Zoloc-Miner). The naïve approach initially indexes the spatial framework using the classical Quad-tree and performs the general co-location method on the zone of interest. Due to the repeated specifications of zones of interest and interest measure values according to user preferences, an excessive amount of computation time is needed to re-calculate the co-locations for each zone of interest and interest measure values. In contrast, I propose to discover co-locations of each indexed space within a new index structure called a clQuad-tree (Section 2.4.2). Then, an algorithm can be used to discover the co-locations of each leaf of the tree (Section 2.4.3). The mining process will start from the bottom to the root of the index. This leads to a decrease in the computational cost as shown in experimental evaluation in Section 2.5. This is because, number of feature-types and instances of a child will be less than its parent and so the spatial join operations at the child level will be cheaper than those at the parent level to discover co-location patterns. After the co-location patterns of the child nodes of a parent are discovered, I only need to check the buffers of the child nodes to know if any co-location pattern or instance has been missed. The co-location
patterns of the parent level will be the union of all child patterns and their buffer patterns.

2.4.1 Naïve Approach

The naïve approach to discover zonal co-location patterns has two phases. The first phase aims to index the space using a spatial indexing structure. Given a zone, the index structure will retrieve the area to be mined. The second phase aims to retrieve the zone of interest from the index structure. Then a co-location mining algorithm [28, 53, 54, 57] can be used to discover the prevalent patterns in that zone using a prevalence threshold. For dynamic parameters, i.e., repeated specification of zone and PI threshold values, the second phase of the naïve approach will be repeated several times. This leads to unnecessary computational costs.

2.4.2 clQuad-Tree Bulk Load Algorithm

This section presents the bulk load algorithm for the proposed index structure clQuad-tree (Algorithm 1) that maintains an index of spatial co-located patterns. Then, in Section 2.4.3, I present the zonal co-location pattern mining algorithm utilizing the clQuad-Tree. The clQuad-Tree is built using a neighborhood distance threshold $R$ that is provided by the user and the set of spatial points or objects. Each spatial point consists of its $x$ and $y$-coordinates and feature-type. In general, the clQuad-tree is built in three phases: (1) the initial quads are created within the clQuad-Tree, (2) points belonging to the quads’ buffer regions are assigned, and (3) the co-locations are generated and stored for each quad in the clQuad-Tree.

**Phase 1-Create Initial clQuad-tree**: In this phase, all possible quads are created for the next two subsequent steps. Each point $p$ in the set $Points$ is inserted into the clQuad-Tree (Lines 2-4 of Algorithm 1). The insertion process is the same as in the classical Quad-Tree [41].

**Phase 2-Create Buffer Regions**: In this phase, the aim is to ensure that potential co-located patterns, i.e., cross-neighbors, are not split because of the quad boundaries. Thus, a buffer is added to each quad for a given neighbor distance $R$ by investigating each point
Algorithm 1 Bulk Load of clQuad-tree

1. Function BULKCLQT (Neighbor $R$, set of points $\text{points}$)
   2. for each object $o \in \text{points}$ {
   3. insert $o$ into clQuad-Tree
   4. }
   5. for each object $o \in \text{points}$ {
   6. if a quad $q$ does not contain $o$ then {
   7. Increase $q$ size by $R$
   8. if $q$ contains object $o$ then {
   9. Insert $o$ into quad $q$'s BufferRegion
   10. }
   11. }
   12. }
   13. for each quad $o \in \text{clQuad-tree}$ {
   14. Generate candidate patterns using quad $q$ object-types
   15. Discover pattern instances using $R$ within all objects BufferRegions
   16. Store patterns as object instances and pattern labels for $q$
   17. }

Object $o$ in the dataset (Line 5 of Algorithm 1). If a quad in the clQuad-Tree does not contain this object, then the quad region is increased by a factor of the distance $R$ only once (Lines 6-7 of Algorithm 1). After the quad region has been expanded, if this object $o$ is now contained in region, then this object is added to this quad (Lines 8-9 of Algorithm 1). Otherwise object is not added to the quad. This process will continue until all points have been added to their respective quad buffer regions (Lines 5-12 of Algorithm 1).

Phase 3-Generate Co-locations: In this phase, the co-located patterns and their instances of each zone are generated (Lines 13-17 of Algorithm 1). First, candidate patterns are generated (Line 14 of Algorithm 1) for each quad and buffer. Second, the instances of the candidates are discovered for a given distance $R$ (Line 15 of Algorithm 1). Finally, the patterns and their instances are stored as a simple table (Line 16 of Algorithm 1). Once the patterns and instances are generated for the leaf nodes and buffers of the tree, the patterns and instances of the parent nodes are found by taking the union of the patterns and instances of leaf nodes and their buffers (Lemma 2.3.1).

Execution Trace: The execution trace of the proposed algorithm is discussed using the dataset in Figure 2.3. The first phase of clQuad-tree indexes the dataset to capture not only
the dataset, but the co-locations as well. This dataset contains four object-types A, B, C, and D and their instances over the space. A, B, C, and D has 8, 6, 4, and 6 instances respectively.

Figure 2.3(a) gives the quad tree representation of the dataset. The lines between features shows the co-located instances that are satisfying distance $R$. The second phase identifies the spatial points that fall into the buffers of each leaf node of the clQuad-tree. The normal and buffer points of zones 1, 2, 3, and 4 are given in Table 2.2. In the third phase,
patterns of the indexed zones are discovered. The pattern of a parent node is the union of its child nodes and their buffers (Lemma 2.3.1). To find the patterns of the root node of the clQuad-tree, the patterns of zones 1, 2, 3, and 4 and their buffers should be found (Table 2.3). Zone 1 and its buffer has no pattern. Table 2.4 gives the patterns of the root node which is the unions of the patterns and instances of the child nodes of the root that are given in Table 2.3.

<table>
<thead>
<tr>
<th>Co-location</th>
<th>Zone 2</th>
<th>Zone 3</th>
<th>Zone 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>Buffer</td>
<td>Normal</td>
</tr>
<tr>
<td>Patterns</td>
<td>A, C</td>
<td>A, D</td>
<td>B, C</td>
</tr>
<tr>
<td></td>
<td>C, D</td>
<td>A, D</td>
<td>A, B</td>
</tr>
<tr>
<td>Instances</td>
<td>&lt;A.3,C.2&gt;</td>
<td>&lt;A.4,D.3&gt;</td>
<td>&lt;B.3,C.2&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;C.3,D.4&gt;</td>
<td>&lt;A.7,D.6&gt;</td>
<td>&lt;A.5,B.4&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;A.3,D.5&gt;</td>
<td>&lt;A.6,B.4&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;A.4,D.6&gt;</td>
<td>&lt;A.7,B.4&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Co-location</th>
<th>Root of Quad Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patterns</td>
<td>A, B</td>
</tr>
<tr>
<td></td>
<td>A, C</td>
</tr>
<tr>
<td></td>
<td>A, D</td>
</tr>
<tr>
<td></td>
<td>B, C</td>
</tr>
<tr>
<td></td>
<td>B, D</td>
</tr>
<tr>
<td></td>
<td>C, D</td>
</tr>
<tr>
<td>Instances</td>
<td>&lt;A.5,B.4&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;A.3,C.2&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;A.4,D.3&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;B.3,C.2&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;B.2,D.3&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;C.3,D.4&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;A.7,D.6&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;A.7,D.6&gt;</td>
</tr>
</tbody>
</table>

### 2.4.3 Zoloc-Miner Algorithm

This section presents the zonal co-location pattern mining algorithm (Algorithm 2) utilizing the clQuad-tree. The Zoloc-Miner is performed in two phases: (1) a set of potential candidates are retrieved for a given zone, and (2) patterns are discovered that satisfy the prevalence threshold $\theta$.

**Phase 1-Zonal Co-location Pattern Retrieval**: In phase 1, co-location instances for a given zone are retrieved from the tree. Each quad $q$ that intersects with the zone $Z$ is inspected (Line 3 of Algorithm 2). Each quad consists of a set of patterns and their instances. If the instances of a pattern intersect the zone, then the pattern is added to the $cand$ set (Line 6 of Algorithm 2). This process continues for all quads intersecting with the
Algorithm 2 Zonal Pattern Mining using clQuad-tree

(1) Function SEARCHCLQT(MiningZone r, clThreshold θ)

{Phase I: Zonal Co-location Pattern Retrieval}

(2) $cand ← 0$

(3) for each quad $q$ intersecting $Z$

(4) for each pattern $p \in q$

(5) if object instances $oi$ for $p$ intersect $r$ then

(6) $cand ← cand ∪ oi$

(7) }

(8) }

{Phase II: Find Prevalent Co-location Patterns}

(10) $ans ← 0$

(11) for each pattern $p \in cand$

(12) Calculate participation index $PI$ for $p$

(13) if $PI > \theta$ then

(14) $ans ← ans ∪ p$

(15) }

(16) }

(17) return $ans$

zone (Lines 3-9 of Algorithm 2).

Phase 2-Find Prevalent Patterns: In phase 2, prevalent patterns that satisfy prevalence threshold $\theta$ are determined. Initially, for each pattern $p$ in the candidate $cand$ set, the PI is calculated (Lines 11-12 of Algorithm 2). If the PI of a pattern is no less than the PI threshold $\theta$, then the pattern is added to the $ans$ set. Finally, the algorithm outputs the set of prevalent zonal co-location patterns $ans$.

Execution Trace: There may be two zones of interest to discover the zonal co-location patterns. If the user-defined zone fits one of the leaf nodes of the tree, Zoloc-Miner will retrieve relevant pattern instances and spatial points of this leaves. Then, the PIs of patterns will be calculated and the ones that do not satisfy the PI threshold $\theta$ will be pruned. For example, assume that zone 2 (Figure 2.3(a)) is the mined area with the PI threshold 0.5. In zone 2, there are 4 candidate patterns and in that zone: A has 3 instances (i.e., A.2, A.3, and A.4), B has 2 instances (i.e., B.2, B.3), C has 1 instance (i.e., C.2), and D has 2 instances (i.e., D.2, D.3). Patterns $\{A,C\}$ and $\{A,D\}$ will be pruned since their PIs are 1/3 and 1/3 respectively. Patterns $\{B,C\}$ and $\{B,D\}$ are discovered since they satisfy the PI threshold of 0.5. If the given zone overlaps more than one node of the clQuad-tree, the points and pattern instances that are intersecting with the zone of interest will be retrieved. Then,
non-prevalent patterns will be pruned. For example, if the zone with the bolded outline in Figure 2.3(b) is the mined area having the PI threshold of 0.5, then there are 3 candidates and zone A has 3 instances (i.e., A.4, A.5, and A.7), B has 2 instances (i.e., B.2, B.4), C has 1 instance (i.e., C.1), and D has 2 instances (i.e., D.3, D.5). Patterns \{A,B\}, \{A,D\} and \{B,D\} are discovered since they satisfy the PI threshold of 0.5.

2.5 Experimental Evaluation

In this section, I present experimental evaluations of several design decisions and workload parameters on proposed algorithm. We evaluated the behavior of the naïve approach and Zoloc-Miner using synthetic datasets to answer the following questions:

- What is the effect of the number of zones?
- What is the effect of the size of a zone?
- What is the effect of the amount of overlap in a zone?

2.5.1 Experimental Setup

Figure 2.4 shows the experimental setup to evaluate the impact of design decisions on the performance on both algorithms.

Synthetic datasets were generated based on the spatial data generator proposed in [28]. Datasets were generated for spatial frame size $D \times D$. For simplicity, the datasets were divided into regular grids whose side lengths had neighborhood relationship $R$. First, sub-sets of future-types were generated using the parameters average co-location size $S_{co-loc}$ and number of maximal patterns $N_{maximal}$. Feature-types and sizes of each pattern were chosen randomly. Next, instances of the patterns were generated based on the average number of co-location instances $S_{instance}$. Instances were randomly located to randomly chosen grid cells. Finally, using the parameters number of noise instances $Noise_{inst}$ and the ratio of noise features over number of features To generate the datasets, frame size $D$, neighborhood distance threshold $R$, number of feature-types, average number of instance
2.5.2 Experiment Results

of features were set at $10^4, 10, 100$, and $10$ respectively. Average co-location size $S_{\text{co-loc}}$ and number of maximal patterns $N_{\text{maximal}}$ were set at 10 and 4 respectively. The ratio of noise features over number of features $L_{\text{noise}}$ was set at 0.25. The number of the noise instances $L_{\text{noise}}$ was set at 500. In the experiments, the PI threshold and distance were set at 0.3 and 10 respectively. Experiments were conducted on an Intel Pentium IV CPU 2.0GHz with 512MB RAM.

2.5.2 Experiment Results

1) Effect of Number of Zones: Figure 2.5 gives execution times of both algorithms for varying number of zones. We ran the experiments up to 1000 zones. The size of each zone was assigned randomly. As can be seen, the execution time of both algorithms increases, as the number of zones is increased. The naïve approach gives better performance only at the first set of zonal regions, this is because the clQuad-tree requires more time to pre-compute and to store the zonal co-location patterns. Later, Zoloc-Miner runs faster since patterns are already stored in the clQuad-tree.

2) Effect of Size of Zone: Figure 2.6 gives the effect of the size of the zones and the execution time for both algorithms. The size of the zones represents the diagonal distance
2.5.2 Experiment Results

Figure 2.5. Effect of Number of Zones

between the minimum $x$ and $y$-coordinates and the maximum $x$ and $y$-coordinates. The minimum $x$ and $y$-coordinates were fixed at (0,0) and the maximum $x$ and $y$-coordinates was incremented by 1K. At zone size <3K, both algorithms have similar performance. This is due to the fact that there were similar numbers of patterns in the Zoloc-Miner and number of objects in the naïve approach. At zone size >3K, Zoloc-Miner outperform the naïve approach. This is because the number of candidates exponentially increase as the number of objects increase from the larger zone.

Figure 2.6. Effect of Size of Zone

3) Effect of Zonal Overlap: Figure 2.7 gives the effect of varying the amount of zonal overlap and execution time for both algorithms. The zonal overlap refers to maximizing the amount of intersection with the quad regions in the index methods. The zone size is
the diagonal distance between the minimum $x$ and $y$-coordinates and the maximum $x$ and $y$-coordinates. However, the zone at 1K is started at the minimum $x$ and $y$-coordinates and maximum $x$ and $y$-coordinates of (45K,45K) and (55K,55K) respectively and incremented by 1K by expanding the zone. At zone size $\leq$3K, both approaches have similar performance due to the limited number of zonal co-location patterns and feature-types within the zone. At zone sizes $>3K$, the Zoloc-Miner outperform the naïve approach, which generates unnecessary candidates as the region increases.

![Figure 2.7. Effect of Zonal Overlap](image)

### 2.6 Conclusions and Future Work

I defined the zonal co-location patterns and mining problem. I developed a novel computationally efficient zonal co-location pattern mining algorithm (Zoloc-Miner) for mining these patterns. I developed an indexing structure (clQuad-tree) to store co-locations and their instances and to handle dynamic parameters, i.e., changing user parameter specifications. The proposed Zoloc-Miner is compared with the Naïve approach, which indexes the space using a classical Quad-tree and runs the co-location mining algorithm for each user-defined zones by retrieving intersecting quads. I also evaluated the proposed algorithms experimentally. As future work, I plan to explore new structures to optimize the storage of the patterns. I also plan to extend the proposed algorithm for mining spatio-temporal patterns [9, 13].
In the classical data mining literature, researchers also developed constrained-based data mining techniques [39, 27]. Constraints are classified as knowledge-type, data, dimension/level, rule, and interestingness constraints [27]. However, these developed techniques are not applicable to the co-location mining, since they do not take into account the spatial aspect of datasets. I plan to extend these approaches for spatial datasets to mine zonal co-location patterns in future.
Mixed-Drove
Spatio-Temporal
Co-occurrence Pattern
Mining

Abstract

Mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) represent subsets of two or more different object-types whose instances are often located in spatial and temporal proximity. Discovering MDCOPs is an important problem with many applications such as identifying tactics in battlefields, games, and predator-prey interactions. However, mining MDCOPs is computationally very expensive because the interest measures are computationally complex, datasets are larger due to the archival history, and the set of candidate patterns is exponential in the number of object-types. We propose a monotonic composite interest measure for discovering MDCOPs and novel MDCOP mining algorithms. Analytical results show that the proposed algorithms are correct and complete. Experimental results also show that the proposed methods are computationally more efficient than naïve alternatives.
3.1 Introduction

As the volume of spatio-temporal data continues to increase significantly due to both the growth of database archives and the increasing number and resolution of spatio-temporal sensors, automated and semi-automated pattern analysis becomes more essential. As a result, spatio-temporal co-occurrence pattern mining has been the subject of recent research. Given a moving object database, the aim is to discover mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) representing subsets of different object-types whose instances are located close together in geographic space for a significant fraction of time. Unlike the objectives of some other spatio-temporal co-occurrence pattern identification approaches where the pattern is the primary interest, in MDCOPs both the pattern and the nature of the different object-types are of interest.

A simple example of an MDCOP is in ecological predator-prey relationships. Patterns of movements of rabbits and foxes, for example, will tend to be co-located in many time-frames which may or may not be consecutive. Rabbits may attempt to move away from foxes, and the foxes may attempt to stay with the rabbits. Other factors such as available food and water may also affect the patterns.

More example MDCOPs may be illustrated in American football where two teams try to outscore each other by moving a football to the opponent’s end of the field. Various complex interactions occur within one team and across teams to achieve this goal. These interactions involve intentional and accidental MDCOPs, the identification of which may help teams to study their opponent’s tactics. In American football, object-types may be defined by the roles of the offensive and defensive players, such as quarterback, running back, wide receiver, kicker, holder, linebacker, and cornerback. An MDCOP is a subset of these different object-types (such as \{kicker, holder\} or \{wide_receiver, cornerback\}) that occur frequently. One example MDCOP involves offensive wide receivers, defensive linebackers, and defensive cornerbacks, and is called a Hail Mary play. In this play, the objective of the offensive wide receivers is to outrun any linebackers and defensive backs and get behind them, catching an undefended pass while running untouched for a touch-
Figure 3.1. An example Hail Mary play in American football in 4 time slots

down. This interaction creates an MDCOP between wide receivers and cornerbacks. An example Hail Mary play is given in Figure 3.1. It shows the positions of four offensive wide receivers (W.1, W.2, W.3, and W.4), two defensive cornerbacks (C.1 and C.2), two defensive linebackers (L.1 and L.2), and a quarterback (Q.1) in four time slots. The solid lines between the players show the neighboring players. The wide receivers W.1 and W.4 cross over each other and the wide receivers W.2 and W.3 run directly to the end zone of the field. Initially, the wide receivers W.1 and W.4 are co-located with cornerbacks C.1 and C.2 respectively and the wide receivers W.2 and W.3 are co-located with linebackers L.1 and L.2 at time slot t=0 (Figure 3.1 (a)). In time slot t=1, the four wide receivers begin to run, while the linebackers run towards the quarterback and the cornerbacks remain in their original position, possibly due to a fake handoff from the quarterback to the running back (Figure 3.1 (b)). In time slot t=2, the wide receivers W.1 and W.4 cross over each other and try to drift further away from their respective cornerbacks (Figure 3.1 (c)). When the quarterback shows signs of throwing the football, both cornerbacks and linebackers run to their respective wide receivers (Figure 3.1(d)). The overall sketch of the game tactics can be seen in Figure 3.1(e). In this example, wide receivers and cornerbacks form an MDCOP since they are persistent over time and they occur 2 out of 4 time slots. However, wide receivers and linebackers do not form an MDCOP due to the lack of temporal persistence.

There are many applications for which discovering co-occurring patterns of specific combinations of object-types is important. Some of these include military (battlefield plan-
ning and strategy), ecology (tracking species and pollutant movements), homeland defense (looking for significant “events”), and transportation (road and network planning) [24, 31].

However, discovering MDCOPs poses several non-trivial challenges. First, current interest measures (i.e. the spatial prevalence measure) are not sufficient to quantify such patterns, so new composite interest measures must be created and formalized [28, 44]. Second, the set of candidate patterns grows exponentially with the number of object-types. Finally, since spatio-temporal datasets are huge, computationally efficient algorithms must be developed [46].

3.1.1 Contributions

The earlier version of this study is published in the proceedings of the 6th IEEE International Conference on Data Mining (ICDM) [13], where I introduced an MDCOP mining problem, proposed a new monotonic composite interest measure, developed two MDCOP algorithms, and evaluated these using real datasets. This chapter makes the following new contributions:

- It defines problem of mining mixed-drove spatio-temporal co-occurrence patterns.
- It proposes new and computationally efficient MDCOP mining algorithms
- It presents experimental results with real and synthetic datasets for all MDCOP algorithms.
- It shows that the proposed algorithms are correct and complete in finding mixed-drove prevalent MDCOPs

3.1.2 Scope and Outline

This chapter focuses on MDCOPs (typed collections of moving objects) by extending interest measures for spatial co-location patterns given a user-defined participation index threshold [28, 44]. The following issues are beyond the scope of this chapter: (i) determining thresholds for MDCOP interest measures; (ii) similarity measures for tracking moving
3.2 Related Work

Data analysis can be broadly categorized into statistical approaches and data mining approaches. In statistical approaches, there are bodies of work in both spatial and temporal analysis. Spatial point patterns are often described by metrics such as the intensity function and Ripley’s K [40, 42]. Other measures such as complete spatial randomness (CSR) and spatial covariance functions are used to describe the spatial relationships of adjacent areas and continuous variables as random fields [14]. Temporal patterns have been extensively studied in models such as moving averages, first and second order autoregression, integration, and periodic patterns such as seasonality [49]. Granger has looked at co-occurring temporal patterns under an assumption of cointegration [21]. There has also been some recent research in combining spatial and temporal analysis, such as Brix and Diggle’s extended intensity function and the extended K(r,t) function [6, 34]. Most attempts to combine the fields suffer from limitations such as the inability to model space-time interactions, treating time as merely another dimension of space and assuming separability and independence between space and time [42]. Statistical research specifically focused on spatio-temporal co-occurrence patterns and their possible interactions has been limited.

Previous data mining studies for mining spatio-temporal co-occurrence patterns can be classified into two categories: mining of uniform groups of moving objects, and mining of mixed groups of moving objects.

To mine uniform groups of moving objects, the problems of discovering flock pat-
terns [33, 23, 22] and moving clusters [30] are defined. A flock pattern is a moving group of the same kind of objects, such as a sheep flock or a bird flock. Gudmundsson et al. proposed algorithms for detection of the flock pattern in spatio-temporal datasets [23, 22]. Kalnis et al. defined the problem of discovering moving clusters and proposed clustering-based methods to mine such patterns [30]. In this approach, if there is a large enough number of common objects between clusters in consecutive time slots, such clusters are called moving clusters. These methods do not take object-types into account, and thus are not effective for mining MDCOPs [13].

To mine mixed groups of moving objects, the problems of discovering collocation episodes [9] and topological patterns [47] are important. Both generalize co-location patterns [28] (subsets of object-types that are frequently located together in space) to the spatio-temporal domain. A collocation episode is a sequence of co-location patterns with some common object-types across consecutive time slots. However, if there is no common object-type in consecutive time slots, the proposed approach will not identify any pattern. For example, the collocation episodes algorithm will not be able to find any pattern from the dataset given in Figure 3.1 if the window length (which is used to find co-location patterns) is 2. For this case, the algorithm tries to find co-location patterns that are persistent in 2 consecutive time slots, but there is no such pattern in the dataset because wide receivers and cornerbacks are forming co-locations in time slots t=0 and t=3 and wide receivers and linebackers are forming co-locations in time slots t=0. Thus, there may not be any co-location patterns and collocation episodes identified in the dataset.

A topological pattern [47] is a subset of object-types whose instances are close in space and time. An interest measure for a topological pattern \{A,B\} (e.g. participation index or support) is a spatio-temporal join of instances of A and instances of B [28]. This statistic may be high even if many instances of A and many instances of B are not spatially together for a moment in time. The semantics of topological patterns are not well-defined for moving objects. For example, this approach can not find an answer to the question of what fraction of time the pattern occurs. The answer of this approach may also be
to the question of when (which time slots) the pattern occurs since there is no time slot notion. In the dataset given in Figure 3.1, this approach will discover the two patterns of \{W, C\} and \{W, L\}. Both patterns have the same support, but pattern \{W, C\} occurs in 2 time slots out of 4 (a persistent pattern) and pattern \{W, L\} occurs in 1 time slot out of 4 (a transient pattern) since tracks of objects are represented as spatio-temporal instances. The persistent pattern \{W, C\} occurs in time slots \(t=0\) and \(t=3\) and its instances \{W1, C1\} and \{W4, C2\} occur in time slot \(t=0\) and \{W1, C2\} and \{W4, C1\} in time slot \(t=1\). The transient pattern \{W, L\} occurs in time slot \(t=0\) and its instances \{W2, L1\}, \{W3, L1\}, \{W2, L2\}, and \{W3, L2\} occur in time slot \(t=0\).

In contrast, the proposed interest measure and algorithms will efficiently mine mixed groups of objects (e.g. MDCOPs) which are close in space and persistent (but not necessarily close) in time. Unlike a number of the techniques just described, the proposed approach will discover persistent patterns that co-occur in most but not all spatio-temporal intervals, so consecutive co-occurrences are not mandatory. For example, the proposed MDCOP mining approach will find the MDCOP \{wide\_receiver, cornerback\} pattern in Figure 3.1, if the fraction of time slots where the pattern occurs over the total number of time slots is no less than a defined threshold, e.g., 0.5. It may reject the pattern \{W,L\} in Figure 3.1 given the lack of time persistence of the \{wide\_receiver, linebacker\} pattern. In fact, instances of MDCOP \{wide\_receiver, cornerback\} are co-located in 2 time slots out of 4 and instances of \{wide\_receiver, linebacker\} are co-located in 1 time slot out of 4. The instances of MDCOP \{wide\_receiver, cornerback\} are \{W.1, C.1\} and \{W.4, C.2\} in time slot \(t=0\), and \{W.4, C.1\} and \{W.1, C.2\} in time slot \(t=3\).

## 3.3 Basic Concepts and Problem Definition

### 3.3.1 Spatial Prevalence Measure

The focus of this study is to discover mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) over a spatio-temporal framework and a neighborhood relation \(R\). First I explain the modeling of mixed groups of object-types in space, e.g., spatial co-locations [44].
In the next sections, I explain how I model MDCOPs by extending spatial co-location mining to include time information and then propose algorithms to mine these MDCOPs.

Spatial co-location mining algorithms are used to discover sets of mixed object-types that are frequently located together in a spatial framework for a given set of spatial object-types, their instances, and a spatial neighbor relationship $R$ [28,44]. For example, in Figure 3.2(a), in time slot $t=0$, $\{A.1, C.1\}$ is an instance of a co-location if the distance between the objects is no more than a given neighborhood distance threshold. In Figure 3.2(a), the solid lines show the distance between the objects that satisfies the neighborhood distance threshold. The participation index is used to determine the strength of the co-location pattern, that is, whether the index is greater than or equal to a threshold [28, 44]. Such a co-location is called spatial prevalent. The participation index is defined as the minimum of the participation ratios (the fraction of the number of instances of object-types forming co-location instances to the total number of instances). For example, in Figure 3.2(a), $\{A, B\}$ is a co-location in time slot $t=0$, and its instances are $\{A.1, B.1\}$, $\{A.2, B.1\}$, $\{A.3, B.2\}$, and $\{A.3, B.3\}$. In the dataset, object-type A has 4 instances and three of them (A.1, A.2, and A.3) are contributing to the co-location $\{A, B\}$, so the participation ratio of A is 3/4. The participation ratio of B is 3/5 since 3 out of 5 instances are contributing to the co-location $\{A, B\}$. The participation index of the co-location $\{A, B\}$ is 3/5, which is the minimum of the participation ratios of object-types A and B. It has been shown that the participation index is anti-monotone in the size of co-locations [28, 44]. In other words, $\text{participation\_index}(P_j) \leq \text{participation\_index}(P_i)$ if $P_i$ is a subset of $P_j$. In addition, [28, 44] show that the participation index has a spatial statistical interpretation as an upper bound on the cross-K function [14].

### 3.3.2 Modeling MDCOPs

Given a set of spatio-temporal mixed object-types and a set of their instances with a neighborhood relation $R$, an MDCOP is a subset of spatio-temporal mixed object-types whose instances are neighbors in space and time.
3.3.2 Modeling MDCOPs

(a) An input spatio-temporal dataset

(b) A set of output mixed-drove spatio-temporal co-occurrence patterns

**Figure 3.2.** An example spatio-temporal dataset

**Definition 3.3.1** Given a spatio-temporal pattern and a set $TF$ of time slots, such that $TF = [T_0, ..., T_n]$, the time prevalence or persistence measure of the pattern is the fraction of time slots where the pattern occurs over the total number of time slots.

For example, in Figure 3.2(a), the total number of time slots is 4 and pattern $\{A, B\}$ occurs in all 4 time slots, so its time prevalence is $4/4$. Pattern $\{A, C\}$ occurs in 3 time slots, namely, time slots $t=0$, $t=1$, and $t=2$, and its time prevalence index is $3/4$.

**Definition 3.3.2** Given a spatio-temporal dataset of mixed object-types $ST$, and a spatial prevalence threshold $\theta_p$, the mixed-drove prevalence measure of a spatio-temporal pattern $P_i$ is a composition of the spatial prevalence and the time prevalence measures as shown below.

$$\text{Prob}_{t_m \in \text{all time slots}}(s_{\text{prev}}(\text{pattern } P_i, \text{time slot } t_m) \geq \theta_p)$$  \hspace{1cm} (3.1)
where $Prob$ stands for probability of overall prevalence time slots and $s_{\text{prev}}$ stands for spatial prevalence, e.g., the participation index, described in Section 3.3.1.

**Definition 3.3.3** Given a spatio-temporal dataset of mixed object-types $ST$ and a threshold pair $(\theta_p, \theta_{\text{time}})$, MDCOP $P_i$ is a mixed-drove prevalent pattern if its mixed-drove prevalence measure satisfies the following.

\[
Prob_{t_m \in \text{all time slot}} [s_{\text{prev}}(\text{pattern } P_i, \text{time slot } t_m) \geq \theta_p] \geq \theta_{\text{time}}
\]  \hspace{1cm} (3.2)

where $Prob$ stands for probability of overall prevalence time slots, $s_{\text{prev}}$ stands for spatial prevalence, $\theta_p$ is the spatial prevalence threshold, and $\theta_{\text{time}}$ is the time prevalence threshold.

For example, in Figure 3.2(a), \{A,B\} is an MDCOP because it has mixed object-types, is spatial prevalent in time slots $t=0$, $t=1$, $t=2$, and $t=3$ since its participation indices are no less than the given threshold 0.4 in these time slots, and is time prevalent since its time prevalence index of 1 is above the time prevalence index threshold 0.5. In contrast, \{B,D\} is not an MDCOP. Although it has mixed object-types and is spatial prevalent in time slot $t=2$, it is not time prevalent since its time prevalence index is no more than the given time prevalence index threshold 0.5.

### 3.3.3 Problem statement

**Given:**

- A set $P$ of distinct Boolean spatio-temporal object-types over a common spatio-temporal framework $STF$.
- A neighbor relation $R$ over locations.
- A spatial prevalence threshold, $\theta_p$.
- A time prevalence threshold, $\theta_{\text{time}}$. 

Find: \{P_i | P_i \text{ is a subset of } P \text{ and } P_i \text{ is a prevalent MDCOP as in Definition 3.3.3}\}.

Objective: Minimize computation cost.

Constraints: To find a correct and complete set of MDCOPs.

Example: In American football, each play (e.g., Figure 3.1) may represent a spatio-temporal dataset and Boolean object-types may be identified by the role of the players (e.g., wide receiver, cornerback, and linebackers). Each object-type is considered as Boolean because I am interested in its presence or absence at any location and time. Figure 3.1(a)-(d) shows the position of the Boolean object-types for four time units. The straight lines between the players show the neighboring objects. The neighbor relation \( R \) may be defined by a distance less than one meter or an average arm’s length. For example, in Figure 3.1(a), wide receiver W.1 is a neighbor of cornerback C.1. However, these players are not neighbors in Figure 3.1(b) since they are separated by more than a meter. In this example, \{wide\_receiver, corner\_back\} forms a candidate MDCOP, given \( \theta_p = 0.5 \), and \( \theta_{time} = 0.5 \).

Threshold values selected for MDCOP interest measures (e.g. spatial prevalence measure and time prevalence measure) have important implications on the mining processes and results. Selection of a small interest measure threshold (close to 0) increases the computational complexity of the algorithms and the number of generated prevalent patterns. This may cause generation of insignificant patterns. Selection of a large interest measure threshold (close to 1) decreases the computational complexity of the algorithms and the number of prevalent patterns. This may cause pruning of some of the significant patterns. Nevertheless the selection of interest measure threshold values is dependent on the application and/or purpose of the analysis.

3.4 Mining MDCOPs

In this section, I discuss a naïve approach and then propose two novel MDCOP mining algorithms - MDCOP-Miner and FastMDCOP-Miner - to mine MDCOPs. I also give execution traces of these algorithms.
3.4.1 Naïve approach

A naïve approach can use a spatial co-location mining algorithm for each time slot to find spatial prevalent co-locations and then apply a post-processing step to discover MDCOPs by checking their time prevalence. To mine co-locations, Huang, Shekhar and Xiong proposed a join-based approach, Yoo, Shekhar and Celik proposed a partial join-based approach and a join-less approach, and Zhang et al. proposed a multi-way spatial join-based approach [8,28,44,53–55,57]. This study will be based on the join-based spatial co-location pattern mining algorithm proposed by Huang et al., but it is also possible to use other approaches. The naïve approach will generate size $k + 1$ candidate co-locations for each time
3.4.2 MDCOP-Miner

To eliminate the drawbacks of the Naïve approach, I propose an MDCOP mining algorithm (MDCOP-Miner) to discover MDCOPs by incorporating a time-prevalence based filtering step in each iteration of the algorithm. The algorithm, first, will discover all size $k$ spatial prevalent MDCOPs and then will apply a time-prevalence based filtering to discover MDCOPs. Finally, the algorithm will generate size $k + 1$ candidate MDCOPs using size $k$ MDCOPs (Figure 3.3(b)). The participation index is used as a spatial prevalence interest measure to check if the pattern is spatial prevalent at a time slot [28]. The time prevalence (i.e., persistence measure in definition 3.3.1) is used as a time prevalence interest measure.

First I give the pseudo code of the algorithm, and then I provide an execution trace of it using the dataset from Figure 3.2(a). Algorithm 3 gives the pseudo code of the MDCOP-Miner algorithm. This pseudo code is used to explain two algorithms: MDCOP-Miner and FastMDCOP-Miner. FastMDCOP-Miner will be discussed in the next section. The choice of the algorithm is provided by the user. The inputs are algorithm choice $alg\_choice$ with value $MDCOP\_Miner$, a set of distinct spatial object-types $E$, a spatio-temporal dataset $ST$, a spatial neighborhood relationship $R$, and thresholds of interest measures, i.e. spatial prevalence and time prevalence; the output is a set of MDCOPs. In the algorithm, steps 1 include initialization of the parameters, steps 2 through 14 give an iterative process to mine MDCOPs, and step 15 gives a union of the results. Steps 2 through 14 continue until there are no candidate MDCOPs to be generated. The functions of the algorithm are explained below.
Algorithm 3 MDCOP-Miner and FastMDCOP-Miner

Inputs:
- \texttt{alg.choice}: Choice of algorithm, "MDCOP-Miner" or "FastMDCOP-Miner"
- \(E\): a set of distinct spatial object-types
- \(ST\): a spatio-temporal dataset \(<\text{object.type}, \text{object.id}, x, y, \text{time slot}\>
- \(R\): spatial neighborhood relationship
- \(TF\): a time slot frame \(\{t_0, \ldots, t_{n-1}\}\)
- \(\theta_p\): a spatial prevalence threshold
- \(\theta_{time}\): a time prevalence threshold

Output: MDCOPs whose spatial prevalence indices, i.e., participation indices, are no less than \(\theta_p\), for time prevalence indices are no less than \(\theta_{time}\)

Variables:
- \(k\): co-occurrence size
- \(t\): time slots \((0, \ldots, n-1)\)
- \(T_k\): set of instances size \(k\) co-occurrences
- \(C_k\): set of candidate size \(k\) co-occurrences
- \(SP_k\): set of spatial prevalent size \(k\) co-occurrences
- \(TP_k\): set of time prevalent size \(k\) co-occurrences
- \(MDP_k\): set of mixed-drove size \(k\) co-occurrences

Algorithm:
(1) initialization : co-occurrence size \(k = 1, C_1 = E, MDP_k(0) = ST\)
(2) while (not empty \(MDP_k\)) {
(3) \(C_{k+1}(0) = \text{gen.candidate.co-occur}(C_k, MDP_k)\)
(4) for each time slot \(t \in (0, \ldots, n-1)\) {
(5) \(T_{k+1}(t) = \text{gen.co-occur.instance}(C_{k+1}(t), T_k(t), R)\)
(6) \(SP_{k+1}(t) = \text{find.spatial.prevalent.co-occur}(T_{k+1}(t), C_{k+1}(t), \theta_p)\)
(7) if \(\text{alg.choice} == \text{"FastMDCOP-Miner"}\) {
(8) \(TP_{k+1}(t) = \text{find.time.prevalence.index}(SP_{k+1}(t))\)
(9) \(MDP_{k+1}(t) = \text{find.time.prevalent.co-occur}(TP_{k+1}(t), \theta_{time})\)
(10) \(C_{k+1}(t) = MDP_{k+1}(t)\) } }
(11) if \(\text{alg.choice} == \text{"MDCOP-Miner"}\) {
(12) \(TP_{k+1} = \text{find.time.prevalence.index}(SP_{k+1})\)
(13) \(MDP_{k+1} = \text{find.time.prevalent.co-occur}(TP_{k+1}, \theta_{time})\) }
(14) \(k = k + 1\) }
(15) return union \((MDP_2, \ldots, MDP_{k+1})\)

Generating candidate co-occurrence patterns (step 3): This function uses an apriori-based approach to generate size \(k + 1\) candidate co-locations \(C_{k+1}\) for each time slot, using all size \(k\) mixed-drove co-occurrence patterns \(MDP_k\) [7].

Generating spatial co-occurrence instances (step 5): The instances of candidate \(C_{k+1}\) are generated by joining neighbor instances of size \(k\) MDCOPs for each time slot. This is similar to the instance generation step of the co-location miner algorithm [28].

Finding spatial prevalent co-occurrence patterns (step 6): All spatial prevalent size \(k + 1\) patterns \(SP_{k+1}\) are found by pruning the ones whose spatial prevalence indices, i.e., participation indices, are less than a given threshold for each time slot. Computation of the
participation indices follows the same algorithmic ideas as those in the co-location mining algorithm [28].

In the for loop, the algorithm finds size $k+1$ spatial prevalent co-location for each time slot. MDCOP-Miner skips steps 8, 9, and 10 which are activated using the FastMDCOP-Miner.

**Forming a time prevalence table (steps 8 and 12):** In steps 8 and 12, the time prevalence indices of the mined spatial prevalent patterns are calculated in FastMDCOP-Miner and MDCOP-Miner algorithms respectively. The time prevalence index of a spatial prevalent co-location is the fraction of the number of time slots where the pattern occurs over the total number of the time slots. Step 8 is activated in FastMDCOP-Miner algorithm and it is used to calculate the time prevalence index of the size $k$ patterns before generating size $k$ patterns of next time slot. Step 12 is activated in MDCOP-Miner and it is used to calculate time prevalence indices of patterns after size $k$ patterns of all time slots are generated.

**Finding mixed-drove co-occurrence patterns (step 9 and step 13):** These steps discover MDCOPs by checking the time prevalence indices of the spatial prevalent co-locations if they are no less than a given time prevalence threshold $\theta_{\text{time}}$. The patterns whose time prevalence indices do not satisfy the given threshold are pruned at this stage. The remaining patterns will be MDCOPs and will be used to generate candidate supersets of the MDCOPs in step 3. In step 13, MDCOP-miner prunes time non-prevalent patterns after all size $k$ patterns in all time slots are generated. In step 9, FastMDCOP-Miner prunes time non-prevalent patterns before generating size $k$ patterns in the next time slot.

The algorithm will run iteratively until there are no more candidate MDCOPs to be generated. The algorithm outputs the union of all size MDCOPs.

**An Execution Trace of MDCOP-Miner:** The execution trace of the MDCOP-Miner is given in Figure 3.4 using the dataset given in Figure 3.2. This dataset contains four object-types A, B, C, and D and their instances in four time slots. A has 4 instances, B has 5 instances, C has 3 instances, and D has 4 instances. The instances of each object-type
have a unique identifier, such as A.1. Some of the patterns of these object-types form an MDCOP. To discover MDCOPs I propose a monotonic composite interest measure (the mixed-drove prevalence measure) which is a composition of the spatial prevalence and time prevalence measures applied to mixed object-types. The spatial prevalence measure (participation index) shows the strength of the spatial co-location when the index is greater than or equal to a given threshold \([28, 44]\). The time prevalence measure (time prevalence index) shows the frequency of the pattern over time.

In Figure 3.4(a), in step 1, candidate pairs of the distinct object-types and their instances are generated for each time slot. The spatial co-locations whose participation indices are less than a given threshold are then pruned. A spatial non-prevalent pair \\{A,D\} is pruned in time slot \(t=0\), \\{C, D\} is pruned in time slots \(t=2\) and \(t=3\), and \\{B,D\} is pruned in time slots \(t=3\), because their participation indices are less than the given threshold 0.4. A time prevalence table of pairs of spatial prevalent co-locations is then formed by entering a 1 if the participation index of the corresponding pattern satisfies a given participation index threshold. Time-prevalence indices are then found. For example, in the time prevalence table (step 2 in Figure 3.4(b)), spatial prevalent pattern \{A,B\} is persistent for all time slots and its time prevalence index is 4/4, and spatial prevalent pattern \{A,C\} is persistent in time slots \(t=0\), \(t=1\), and \(t=2\) and its time prevalence index is 3/4. The MDCOPs whose time prevalence indices are no less than a given threshold are selected for generating superset candidate MDCOPs.

Spatial prevalent patterns \{A,B\}, \{A,C\}, and \{B,C\} are selected as MDCOPs since they are also time prevalent (their time prevalence indices satisfy the given time prevalence index threshold 0.5). In contrast, spatial prevalent patterns \{A,D\}, \{B,D\}, and \{C,D\} are pruned since they are time non-prevalent. Using MDCOPs \{A,B\}, \{A,C\}, and \{B,C\}, the next candidate MDCOP \{A,B,C\} is generated. The next step is to generate instances of candidate \{A,B,C\} in time slots where its subsets exist and to check its participation indices in these time slots. Since all subsets of MDCOP \{A,B,C\} are MDCOPs and exist in time slots \(t=1\) and \(t=2\), there is no need to generate instances of them for time slots \(t=0\) and \(t=3\).
3.4.3 Modified MDCOP-Miner (FastMDCOP-Miner)

In this section, I propose a new algorithm, called FastMDCOP-Miner, which improves the computational efficiency of the MDCOP-Miner discussed in Section 3.4.2. As can be seen in Figure 3.3(b) and in Algorithm 3, MDCOP-Miner waits to prune time non-prevalent patterns until all size $k$ spatial prevalent patterns are generated for all time slots and then prunes time non-prevalent patterns to discover MDCOPs. However, it is possible
to optimize the MDCOP-Miner. I propose to prune time-non prevalent patterns as early as possible by moving “prune non-prevalent patterns” between the time slots shown in Figure 3.3(c) where candidate size 2 pattern generation is illustrated. The pseudo-code of the FastMDCOP-Miner is given in Algorithm 3. When the FastMDCOP-Miner is chosen, the algorithm will activate steps 8, 9, and 10 and deactivate steps 12 and 13. This will allow the algorithm to check the time prevalence of a pattern after every time slot is processed. The functions of the algorithm are as described in Section 3.4.2. In step 8, FastMDCOP-Miner checks whether the time prevalence indices of size $k$ patterns (size 2 patterns in Figure 3.3(c)) satisfy the time prevalence index threshold before generating size $k$ patterns for the next time slot. Early discovered time non-prevalent patterns are pruned in Step 9 and time prevalent patterns are used as candidate co-occurrences (Step 10) in the next time slot. For example, assume that there are 10 time slots and the time prevalence index threshold is 0.5. In this case, a size $k$ pattern should be present for at least 5 time slots to satisfy the threshold. If the time prevalence index of a pattern is 0 for the first (or any) 6 time slots, there is no need to generate this pattern and check the prevalence of it for the rest of the time slots. Even if it is time persistent for the remaining 4 time slots, it will not be able to satisfy the given time prevalence index threshold.

An Execution Trace of FastMDCOP-Miner: The execution trace of the FastDCOP-Miner is given in Figure 3.5 using the dataset given in Figure 3.2, which has four time slots. Assume that the spatial prevalence index threshold is 0.4 and the time prevalence index threshold is 0.75. If a pattern is not consistent in more than 1 out of 4 time slots, it can be pruned whenever it is discovered. In Step 1(a) pairs and their instances are generated. Pattern \{A,D\} is pruned at this step since it is spatial non-prevalent. Based on the outcomes of Step 1(a), the prevalence table is updated by entering a 1 for spatial prevalent patterns (Step 1(b)). The time prevalence table initially contains all possible pairs of subsets of object-types. The algorithm checks if time non-prevalent patterns can be discovered at this step. Since the result of one time slot are not enough to make a decision, instances for
3.5 Analysis of the MDCOP-Miner

This section gives the analysis of the mixed-drove prevalence index measure, and correctness and completeness derivations for the MDCOP mining algorithms.
3.5.1 The Mixed-Drove Prevalence Index Measure is Monotonic

Lemma 3.5.1 Spatial prevalence measure participation index and participation ratio are monotonically non-increasing in the size of the MDCOPs at each time slot \([28, 44]\).

Proof The participation ratio \(pr\) is monotonically non-increasing because an instance of a spatial object-type that is contributing to a co-location \(c_i\) is also contributing to a co-location \(c_j\) where \(c_j \subseteq c_i\). The spatial prevalence measure participation index \(pi\) is also monotonic because 1) participation ratio is monotonic and 2)

\[
pi(c \cup f_{k+1}) = \min_{i=1}^{k+1} \{pr(c \cup f_{k+1}, f_i)\}
\]

\[
\leq \min_{i=1}^{k} \{pr(c \cup f_{k+1}, f_i)\} \leq \min_{i=1}^{k} \{pr(c \cup f_i)\} = pi(c)
\]

Lemma 3.5.2 A mixed-drove prevalence index measure is monotonically non-increasing with the size of MDCOP over space and time. In other words, it is monotonically non-increasing, if MDCOP \(P_i\) is a subset of MDCOP \(P_j\) and

\[
Prob_{t_m \in \text{all time slot}}(s_{\text{prev}}(P_i, t_m) \geq \theta_p), \text{ and } Prob_{t_m \in \text{all time slot}}(s_{\text{prev}}(P_j, t_m) \geq \theta_p),
\]

where \(Prob\) stands for the probability of overall prevalence time units, \(s_{\text{prev}}\) stands for spatial prevalence, \(\theta_p\) is the spatial prevalence threshold, and \(t_m\) is the time slot.

Proof The basic proof sketch follows. Let \(TS(P_j, \theta_p) = \{t_m | pi(P_j, t_m) \geq \theta_p\}\).

Lemma 3.5.1 implies that participation index \(pi(P_j, t_m) \geq \theta_p\) for all \(t_m \in TS(P_j, \theta_p)\), since \(P_i\) is a subset of \(P_j\). Thus, \(Prob_{t_m \in \text{all time slot}}[s_{\text{prev}}(P_i, t_m) \geq \theta_p] \in \theta_{\text{time}}\), where \(\theta_{\text{time}}\) is the time prevalence threshold.

3.5.2 Correctness and Completeness

Theorem 3.5.1 The FastMDCOP-Miner, MDCOP-Miner, and naive approach are complete.

Proof The FastMDCOP-Miner, MDCOP-Miner, and naive approach are complete if they find all MDCOPs that satisfy a given participation index threshold and time prevalence
threshold. We can show this by proving that none of the functions of the algorithm miss any patterns, i.e., filter out a prevalent MDCOP.

The \textit{gen\_candidate\_co-occur} function does not miss any patterns given the anti-monotone nature of the MDCOP interest measure. The input of this function is size \( k \) MDCOPs and the output is candidate size \( k+1 \) MDCOPs. If \( c_1 = \{f_1, \ldots, f_k\} \) and \( c_2 = \{f_1, \ldots, f_{k-1}, f_{k+1}\} \) are size \( k \) MDCOPs, candidate size \( k+1 \) pattern \( C_{k+1} = \{f_1, \ldots, f_{k-1}, f_k, f_{k+1}\} \) will be produced by joining size \( k \) MDCOPs.

The \textit{gen\_co-occur\_instance} function does not miss any patterns. This function generates instances of candidate size \( k+1 \) MDCOPs by joining instances of size \( k \) MDCOPs if they are in the neighborhood distance and forming a clique.

The \textit{find\_spatial-prevalent\_co-oc} function does not miss any patterns. It finds spatial prevalent patterns whose participation indices satisfy a given threshold.

The \textit{find\_time-prevalence\_index} function does not miss any patterns. This function calculates time prevalence indices of the patterns found in steps 4 through 8 and does not do any pruning.

The \textit{find\_time-prevalent\_co-occur} function does not miss any MDCOPs. The function finds all the MDCOPs whose time prevalence indices are no less than a given threshold.

\textbf{Theorem 3.5.2} \textit{The FastMDCOP-Miner, MDCOP-Miner, and naive approach are correct. In other words, if a MDCOP pattern \( P \) is returned by MDCOP-Miner and FastMDCOP-Miner algorithms, then \( P \) is a prevalent MDCOP.}

\textbf{Proof} The proof is easy to establish due to the pruning steps of \textit{find\_spatial\_prevalent\_co-occur}, and \textit{find\_time\_prevalent\_co-occur}, which weed out candidates not meeting the given thresholds.

\subsection*{3.5.3 Algebraic Cost Model}

In this section, we give the algebraic cost models of the MDCOP-Miner and the FastMDCOP-Miner algorithms. The cost model of the naive approach is not given since it is the worst
case of the MDCOP-Miner and FastMDCOP-Miner and applies the pruning strategy in a post processing step. Let $T_{MDCOP}$ and $T_{Fast}$ represent the total computational costs of the MDCOP-Miner and the FastMDCOP-Miner respectively. The total respective cost functions will be

$$T_{MDCOP} = \sum_{k>1} T_{MDCOP}(k, \theta_p, \theta_{time}, TF, S_{instance})$$

$$T_{Fast} = \sum_{k>1} T_{Fast}(k, \theta_p, \theta_{time}, TF, S_{instance})$$

(3.3)

where $T_{MDCOP}(k, \theta_p, \theta_{time}, TF, S_{instance})$ and $T_{Fast}(k, \theta_p, \theta_{time}, TF, S_{instance})$ represent the generation of the total cost of size $k$ ($k > 1$) MDCOPs for parameters $\theta_p$ and $\theta_{time}$, time slots $TF$, and the average number of co-occurrence instances $S_{instance}$.

$$T_{MDCOP}(k, \theta_p, \theta_{time}, TF, S_{instance}) = T_{gen\_candi}(MDP_{k-1}, TF, S_{instance})
+ T_{prune\_sp\_co\_occ}(C\_SP_{k}, \theta_p, TF, S_{instance})
+ T_{prune\_time\_co\_occ}(SP_{k}, \theta_{time}, TF, S_{instance})$$

$$\approx T_{gen\_candi}(MDP_{k-1}, TF, S_{instance}) + T_{prune\_sp\_co\_occ}(C\_SP_{k}, \theta_p, TF, S_{instance})$$

(3.4)

$$T_{Fast}(k, \theta_p, \theta_{time}, TF, S_{instance}) = T_{gen\_candi}(MDP_{k-1}, TF, S_{instance})
+ T_{prune\_sp\_co\_occ}(C\_SP_{k}, \theta_p, TF, S_{instance})
+ T_{prune\_time\_co\_occ}(SP_{k}, \theta_{time}, TF, S_{instance})$$

$$\approx T_{gen\_candi}(MDP_{k-1}, TF, S_{instance}) + T_{prune\_sp\_co\_occ}(C\_SP_{k}, \theta_p, TF, S_{instance})$$

$MDP_{k-1}$ is the size $k - 1$ MDCOP sets. $T_{gen\_candi}$ represents the total cost of generating candidate MDCOPs for all time slots $TF$. $C\_SP_{k}$ represents the number of candidate size $k$ MDCOPs. $T_{prune\_sp\_co\_occ}$ represents the total cost of pruning candidate MDCOPs. $SP_{k}$ represents the number of size $k$ spatial co-location patterns of MDCOP mining algorithms.
$T_{\text{prune\_time\_co\_occ}}$ represents the total cost of pruning time non-prevalent patterns. MDCOP-Miner will generate all size $k$ spatial prevalent patterns and then it will prune size $k$ time non-prevalent patterns. In contrast, FastMDCOP-Miner will apply time non-prevalent pattern pruning as early as possible to eliminate generating instances of some size $k$ time non-prevalent patterns. The cost of generating size $k$ patterns of FastMDCOP-Miner will be no more than that of MDCOP-Miner. The FastMDCOP-Miner approach will run steps for the number of time slots in the dataset to find all size $k$ MDCOPs. The cost of this process is negligible since it only checks the count of the patterns whenever a new pattern is processed. If there is no pattern to be pruned early, the cost of the both approaches will be same. As a result, the cost of $MDP^F_{k-1}$ will be no more than that of $MDP^M_{k-1}$ and the cosf of $CSP^F_k$ will be no more then that of $CSP^M_k$.

**Lemma 3.5.3** The total cost of time based pruning is negligible with respect to total cost of spatial prevalence based pruning, such that,

$$T_{\text{MDCOP\_or\_Fast}} = T_{\text{gen\_candi}} + T_{\text{prune\_sp\_co\_occ}} + T_{\text{prune\_time\_co\_occ}} \approx T_{\text{gen\_candi}} + T_{\text{prune\_sp\_co\_occ}}$$ (3.5)

**Proof** The total cost $T_{\text{gen\_candi}} + T_{\text{prune\_sp\_co\_occ}}$ includes the cost of generating all candidate patterns and their instances and the cost of calculating the participation index and ratio values of the generated patterns and pruning the non-prevalent patterns, respectively. This process is computationally complex due to the spatial join operation to generate the candidate patterns and their instances in each time slot. That is, the bulk of time is consumed in generating instances and pruning spatial non-prevalent MDCOPs.

In contrast, the total cost of $T_{\text{prune\_time\_co\_occ}}$ includes the cost of calculating the time prevalence indices of the patterns for all time slots. This process is computationally very cheap due to the count of the existence of the patterns in whole dataset.

**Lemma 3.5.4** The total cost of generating candidate MDCOPs of the proposed FastMDCOP-Miner is no more than the total cost of generating MDCOPs of the MDCOP-Miner, assuming the cost of the time prevalence based pruning is negligible (Lemma 3.5.3), such that,
3.5.3 Algebraic Cost Model

\[ T_{\text{gen.candi}}(\text{MDP}_{k-1}^{\text{Fast}}, TF, S_{\text{instance}}) \leq T_{\text{gen.candi}}(\text{MDP}_{k-1}^{\text{MDCOP}}, TF, S_{\text{instance}}) \]  (3.6)

**Proof** Due to the early pruning of irrelevant candidates (steps 8 and 9 of Algorithm 3), the number of sets of size \( k \) MDCOPs generated by the FastMDCOP-Miner will be no more than that of the MDCOP-Miner size \( k \) patterns, e.g., \( \text{MDP}_{k-1}^{\text{Fast}} \leq \text{MDP}_{k-1}^{\text{MDCOP}} \), since size \( k \) patterns of the FastMDCOP-Miner will prune time non-prevalent patterns earlier than MDCOP-Miner.

The other parameters affecting equation 3.6 are the number of time slots \( TF \) and average number of co-occurrence instances \( S_{\text{instance}} \). If the number of time slots increases, the cost of the MDCOP-Miner will increase due to the increasing unnecessary generation of time non-prevalent patterns. Similarly, for greater average numbers of co-occurrence instances, the cost of the MDCOP-Miner increases. In the worst case, the number of sets of size \( k \) patterns generated will be equal for both algorithms, if there is no early time prevalence based pruning. In that case, equation 3.6 will be true.

**Lemma 3.5.5** The total cost of pruning the spatial non-prevalent candidate patterns of the FastMDCOP-Miner will be no more than that of MDCOP-Miner, assuming the cost of the time prevalence based pruning is negligible (Lemma 3.5.3), such that

\[ T_{\text{prune.sp.coocc}}(C_{SP_k}^{\text{Fast}}, \theta_p, TF, S_{\text{instance}}) \leq T_{\text{prune.sp.coocc}}(C_{SP_k}^{\text{MDCOP}}, \theta_p, TF, S_{\text{instance}}) \]  (3.7)

**Proof** Due to early pruning of irrelevant candidates (step 6 of Algorithm 3), the number of size \( k + 1 \) candidate pattern sets generated by the FastMDCOP-Miner will be no more than that of the MDCOP-Miner e.g., \( C_{SP_k}^{\text{Fast}} \leq C_{SP_k}^{\text{MDCOP}} \), and so equation 3.7 will be true.

**Lemma 3.5.6** The total cost of pruning time non-prevalent candidate MDCOPs of the proposed FastMDCOP-Miner algorithm is no more than the total cost of pruning
time non-prevalent candidate patterns of the MDCOP-Miner, assuming the cost of the time prevalence based pruning is negligible (Lemma 3.5.3), such that

\[ T_{\text{prune,co-occ}}(SP_{F,k}, \theta_{\text{time}}, TF, S_{\text{instance}}) \leq T_{\text{post,NA}}(SP_{M,k}, \theta_{\text{time}}, TF, S_{\text{instance}}) \] (3.8)

**Proof** Due to early pruning of irrelevant candidates (step 10 of Algorithm 3), the number of size \( k \) spatial prevalent patterns generated by the FastMDCOP-Miner will be no more than that of the MDCOP-Miner, e.g., \( SP_{F,k} \leq SP_{M,k} \) and so equation 3.8 will be true.

**Theorem 3.5.3** The total cost of the FastMDCOP-Miner is no more than the total cost of the MDCOP-Miner assuming the cost of the time prevalence based pruning is negligible.

**Proof** Based on Lemmas Lemma 3.5.3, 3.5.4, 3.5.5, and 3.5.6 the total cost of the FastMDCOP-Miner algorithm will be no more than the total cost of the MDCOP-Miner.

### 3.6 Experimental Evaluation

In this section, I present experimental evaluations of several design decisions and workload parameters on MDCOP mining algorithms. I used a real-world training dataset and synthetic datasets. I evaluated the behavior of the FastMDCOP-Miner, MDCOP-Miner and naïve approach to answer the following questions:

- What is the effect of the number of timeslots?
- What is the effect of the number of object-types?
- What is the effect of the spatial prevalence index threshold?
- What is the effect of the time prevalence index threshold?
- What is the effect of the number of noise instances?
What is the effect of the average number of instances?

Figure 3.6 shows the experimental setup to evaluate the impact of design decisions on the performance on the three algorithms. Experiments were conducted on an IBM Netfinity Linux Cluster, 2.6 GHz Intel Pentium 4 with 1.5 GB of RAM.

3.6.1 Datasets

Real Dataset

The real dataset contains the location and time information of moving objects. It includes 15 time snapshots and 22 distinct vehicle types and their instances. The minimum instance number is 2, the maximum instance number is 78, and the average number of instances is 19.

Synthetic Dataset Generation

To evaluate the performance of the algorithms, spatio-temporal datasets were generated based on the spatial data generator proposed by Huang et al. [28]. Synthetic datasets were generated for spatial frame size $D 	imes D$ (the first part of Figure 3.6). For simplicity,
the datasets were divided into regular grids whose side lengths had neighborhood relationship \( R \). First, subsets of object-types were generated using the parameters average co-occurrence size \( S_{\text{co-occur}} \) and number of maximal patterns \( N_{\text{maximal}} \). Object-types and sizes of each pattern were chosen randomly. The generated patterns were then divided into two categories - persistent patterns and transitory patterns - using the persistent pattern ratio \( L_{\text{persistent}} \). Persistent patterns are ones whose time prevalences are strong over time, while transitory patterns are ones whose spatial prevalences are strong at a specific time slot. Next, instances of the patterns were generated based on the average number of co-occurrence instances \( S_{\text{instance}} \). Instances were chosen at randomly located grid cells. This process was applied for each time slot \( T_F \). Finally, using the parameters number of noise instances \( N_{\text{noise inst}} \) and ratio of noise objects over number of features \( L_{\text{noise}} \), noise object and their instances were generated and added to the dataset.

The parameters of the synthetic dataset generator and their definitions are listed in Table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E )</td>
<td>Number of object types</td>
<td>Syn-1</td>
</tr>
<tr>
<td>( S_{\text{instance}} )</td>
<td>Average number of co-occurrence instances*</td>
<td>10</td>
</tr>
<tr>
<td>( N_{\text{maximal}} )</td>
<td>Number of maximal co-occurrence patterns*</td>
<td>10</td>
</tr>
<tr>
<td>( S_{\text{co-occur}} )</td>
<td>Minimum co-occurrence pattern size*</td>
<td>4</td>
</tr>
<tr>
<td>( L_{\text{persistent}} )</td>
<td>Ratio of persistent patterns over transitory patterns*</td>
<td>0.5</td>
</tr>
<tr>
<td>( L_{\text{noise}} )</td>
<td>Ratio of noise object-types over number of object-types</td>
<td>0.25</td>
</tr>
<tr>
<td>( N_{\text{noise inst}} )</td>
<td>Number of noise instances</td>
<td>1000</td>
</tr>
<tr>
<td>( T_F )</td>
<td>Number of timeslots</td>
<td>10-50</td>
</tr>
<tr>
<td>( D )</td>
<td>Spatial framework size ((D \times D))</td>
<td>(10^6)</td>
</tr>
<tr>
<td>( R )</td>
<td>Spatial Neighborhood relationship</td>
<td>10</td>
</tr>
</tbody>
</table>

*: For initial co-occurrence patterns

### 3.6.2 Experiment Results for Real Datasets

#### Effect of Number of Time Slots

In the first experiment, I evaluated the effect of the number of time slots on the execution time of the MDCOP algorithms using the real dataset. The participation index, time preva-
lence index, and distance were set at 0.2, 0.8, and 150m respectively. Experiments were run for a minimum of 1 time slot and a maximum of 14 time slots. As can be seen in Figure 3.7(a), the FastMDCOP-Miner requires less execution time than the MDCOP-Miner and naïve approaches, since it prunes out time non-prevalent MDCOPs as early as possible. It can also be seen that as the number of time slots increases, the ratio of the increase in execution time is smaller for FastMDCOP-Miner than for the MDCOP-Miner and naïve approaches. Figure 3.7(b) shows the number of generated size 2 and size 3 instances for algorithms. The FastMDCOP-Miner generates fewer patterns due to its early pruning of time non-prevalent patterns. The MDCOP-Miner and naïve approaches generate the same number of size 2 instances. The MDCOP-Miner applies time pruning after it generates all possible size 2 patterns and the naïve approach applies time pruning in the post-processing step.

As discussed in Section 3.5.3, the number of time slots \( TF \) is one of the parameters that affects the cost of the algorithms. As predicted in equation 3.3, the cost of the algorithms increases as the number of time slots increases and the FastMDCOP-Miner outperforms the other approaches due to the early pruning strategy (Theorem 3.5.3). The FastMDCOP-Miner algorithm examines fewer instances than the other approaches since it deals with the
MDCOP prevalent patterns as early as possible (Lemmas 3.5.4, 3.5.5, and 3.5.6).

**Effect of Number of Object-types**

In the second experiment, I evaluated the effect of the number of object-types on the execution time of the algorithms using the real dataset. The participation index, time prevalence index, number of time slots and distance were set at 0.2, 0.8, 15, and 150m respectively. The FastMDCOP-Miner outperforms the other approaches as the number of object-types increases (Figure 3.8(a)-(b)). It is observed that the increase in execution time for the naïve approach is bigger than that of the MDCOP-Miner and the FastMDCOP-Miner as the number of object-types increases for datasets.

![Figure 3.8](image)

**Figure 3.8.** Effect of number of object-types in MDCOP mining algorithms using real dataset

The trends are consistent with the algebraic cost models given in equation 3.3. The cost of the algorithms increases as the number of the object-types increases due to the increase in the number of join operations. The MDCOP-Miner and naïve approaches generate the same number of size 2 instances. MDCOP-Miner applies time pruning after it generates all possible size 2 patterns and naïve approach applies time pruning in the post-processing step (Figure 3.8(b)). In contrast, The FastMDCOP-Miner generates patterns by pruning non-prevalent patterns as early as possible (Lemmas 3.5.4, 3.5.5, and 3.5.6). Because of the
3.6.2 Experiment Results for Real Datasets

early pruning strategy of FastMDCOP-Miner, its cost is no more than that of the MDCOP-Miner and naïve approaches as shown in Figure 3.8(a) (Theorem 3.5.3).

**Effect of the Time Prevalence Index Threshold**

In the third experiment, I evaluated the effect of the time prevalence index threshold on the execution times of the MDCOP mining algorithms for the real dataset. The fixed parameters were participation index, number of time slots, and distance, and their values were 0.2, 15, and 150m respectively. For the naïve approach, the effective cost in execution time to generate spatial prevalent co-locations will be constant since it generates the same number of spatial prevalent patterns as the time prevalence index increases. In that case, the cost of the post-processing step will reflect the trend of the naïve approach. Experimental results show that the FastMDCOP-Miner is more computationally efficient than the other approaches because of the early pruning strategy (Figure 3.9(a)). The execution times of the FastMDCOP-Miner and MDCOP-Miner decrease as the time prevalence index threshold increases. It is also observed that the naïve approach is computationally more expensive as the time prevalence index threshold decreases because of the increase in the number of MDCOPs to be discovered.

![Graph](image)

(a) Execution time of the algorithms  
(b) Number of generated instances

**Figure 3.9.** Effect of the time prevalence index threshold in MDCOP mining algorithms using real dataset
The trends are consistent with the algebraic cost models given in equation 3.3. The cost of FastMDCOP-Miner is no more than that of MDCOP-Miner (Lemma 3.5.6 and Theorem 3.5.3). The naïve approach is not sensitive to the time prevalence index threshold (Lemma 3.5.6). Without the post-processing step, the cost of the naïve approach is constant. The trend of the naïve approach in Figure 3.9 is characterized by the cost of the post-processing step.

**Effect of the Spatial Prevalence Index Threshold**

In the fourth experiment, I evaluated the effect of the spatial prevalence index threshold on the execution times of the MDCOP algorithms. The fixed parameters were time prevalence index, number of time slots, and distance, with values of 0.2, 15, and 100m respectively. Figure 3.10(a) shows the execution times of the algorithms and Figure 3.10(b) shows the number of generated size 2 and 3 instances for the algorithms. FastMDCOP-Miner and MDCOP-Miner do not generate more than size 3 instances for a spatial prevalence index threshold of greater than 0.2. The FastMDCOP-Miner outperforms the MDCOP-Miner and naïve approaches as the spatial prevalence index threshold increases (Figure 3.10(a)-(b)). The cost of the naïve approach will be higher than that of the FastMDCOP-Miner and MDCOP-Miner for low values of the spatial prevalence index threshold.

The trends are consistent with the algebraic cost models given in equation 3.3. The algorithms are sensitive to the spatial prevalence index threshold (Lemma 3.5.5).

**3.6.3 Experiment Results for Synthetic Datasets**

**Effect of Number of Time slots**

In this experiment, I evaluated the effect of the number of time slots on the execution time of the algorithms using synthetic datasets. To generate the datasets, I used a framework size of $10^6 \times 10^6$, a square proximity neighborhood size of 10 x 10, a noise feature ratio of 0.25, a noise instance number of 1000, an average number of co-occurrence instances of 10, and a maximal co-occurrence pattern number of 10 (Table 3.1, column syn-1). In
the experiments, the participation index, time prevalence index, and distance were set at 0.3, 0.9, and 10m respectively. Experiments were for a minimum of 10 time slots and a maximum of 50 time slots. The results showed that the FastMDCOP-Miner requires less execution time than the other approaches, since it prunes out time non-prevalent MDCOPs as early as possible (Figure 3.11(a)). The generated size 2 and size 3 instances are given in Figure 3.11(b). The naive approach generated up to size 7 spatial prevalent subsets before the post-processing step. In contrast, FastMDCOP-Miner and MDCOP-Miner generated up to size 4 subsets.

The trends are consistent with the algebraic cost models given in equation 3.3. The cost of the algorithms increases as the number of time slots increases and the FastMDCOP-Miner outperforms the other approaches due to its early pruning strategy ((Theorem 3.5.3). As can be seen in Figure 3.11(b), the FastMDCOP-Miner algorithm examines fewer size 2 and 3 instances than the other approaches due to the early pruning strategy (Lemmas 3.5.4, 3.5.5, and 3.5.6).

**Effect of Number of Object-types**

I evaluated the effect of the number of object-types on the execution time of the algorithms for synthetic datasets. The parameters used to generate the datasets are given in Table 3.1,
3.6.3 Experiment Results for Synthetic Datasets

Effect of the number of time slots in MDCOP mining algorithms using synthetic dataset

The participation index, time prevalence index, number of time slots, and distance were set at 0.3, 0.8, 20, and 10m respectively. The FastMDCOP-Miner outperforms the MDCOP-Miner and naïve approaches as the number of object-types increases (Figure 3.12(a)-(b)). The ratio of the increase in the execution time of the naïve approach is greater than that of the MDCOP-Miner and FastMDCOP-Miner as the number of object-types increases. Figure 3.12(b) shows the number of generated size 2 and 3 instances for the algorithms.

The trends are consistent with the algebraic cost models given in equation 3.3. The cost of the algorithms increases as the number of the object-types increases due to the increase in the number of join operations.

Effect of the Time Prevalence Index Threshold

I evaluated the effect of the time prevalence index threshold on the execution times of the algorithms for synthetic datasets. The parameters used to generate the datasets are given in Table 3.1 column syn-3. The fixed parameters were participation index, distance, and number of time slots, and their values were 0.4, 10m, and 50 respectively. Experimental results show that the FastMDCOP-Miner is more computationally efficient than the MDCOP-miner and naïve approaches because of the early pruning strategy (Figure
3.6.3 Experiment Results for Synthetic Datasets

![Graphs showing execution time and number of generated instances for different algorithms.](image)

(a) Execution time of the algorithms  
(b) Number of generated instances

Figure 3.12. Effect of number of object-types in MDCOP mining algorithms using synthetic dataset

3.13(a)). The execution time of the FastMDCOP-Miner decreases as the time prevalence index threshold increases. The execution time of the naïve approach is almost constant since it does not prune time non-prevalent pattern before the post-processing step and it is computationally more expensive as the time prevalence index threshold decreases because of the increase in the number of MDCOPs to be generated (Figure 3.13(b)).

The naïve approach is not sensitive to the time prevalence index threshold (Lemma 3.5.6) which causes the increase of its cost. Without the post-processing step, the cost of the naïve approach is constant. The trend of the naïve approach in Figure 3.9 is characterized by the cost of the post-processing step (Figure 3.13(a)).

**Effect of the Spatial Prevalence Index Threshold**

I evaluated the effect of the spatial prevalence index threshold on the execution times of MDCOP mining algorithms. To generate the dataset, I used a spatial framework size of $10^6 \times 10^6$, a square proximity neighborhood size of 10 x 10, an average number of co-occurrence instances of 10, a noise feature ratio of 0.25, a noise instance number of 1000, a maximal co-occurrence pattern number of 10, and a time slot number of 50 (Table 3.1, column syn-4). In the experiments, the value of the time prevalence index was 0.8. The FastMDCOP-Miner outperforms the other approaches (Figure 3.14(a)-(b)). The cost of the
Figure 3.13. Effect of the time prevalence index threshold in MDCOP mining algorithms using synthetic dataset.
3.6.3 Experiment Results for Synthetic Datasets

FastMDCOP-Miner is more robust than the MDCOP-Miner and naïve approaches as the number of noise features increases (Figure 3.15(a)).

The trends are consistent with the algebraic cost models given in equation 3.3. The FastMDCOP-Miner is more robust than that of the other approaches (Lemmas 3.5.6 and 3.5.5 and Theorem 3.5.3). In other words, the naïve approach is more sensitive to the noise features, which causes both its cost and the number of generated instances to increase (Figure 3.15(a)).

**Effect of the Average Number of Instances**

I evaluated the average number of total instances on the execution times of the MDCOP mining algorithms using synthetic datasets. The parameters for data generation are listed in column syn-6 of Table 3.1. The time prevalence index threshold, the spatial prevalence index threshold, and distance were 0.3, 0.8, and 10m respectively. The FastMDCOP-Miner algorithm outperformed the other approaches as the average number of total instances increases (Figure 3.15(b)).


3.7 Conclusions and Future Work

I defined mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) and the MDCOP mining problem and proposed a new monotonic composite interest measure which is the composition of distinct object-types, spatial prevalence measures, and time prevalence measures. I presented a novel and computationally efficient algorithm (the MDCOP-Miner) for mining these patterns. I also presented an improved MDCOP-Miner algorithm (the FastMDCOP-Miner) which prunes time non-prevalent patterns at an early stage and offers even greater computational efficiency than the MDCOP-Miner algorithm. I compared the proposed algorithms with a naïve approach, which runs the spatial co-location mining algorithm at each time slot and then discovers MDCOPs using a post-processing step. I proved that the proposed algorithms are correct and complete in finding mixed-drove prevalent (i.e., spatial prevalent and time prevalent) MDCOPs. The experimental results using a real and synthetic datasets provide further evidence of the viability of the proposed approach.

For future work, I would like to explore the relationship between the proposed MDCOP interest measures and spatio-temporal statistical measures of interaction [8]. Another problem of interest is the characterization of the probability distribution of the proposed interest measure to help in making the choice of thresholds in the proposed measures. I plan to ex-
plore other potential interest measures for MDCOPs by evaluating similarity measures for tracks of moving objects. I plan to investigate new monotonic composite interest measures and develop other new computationally efficient algorithms for mining MDCOPs.

In the literature, there are also other studies that have focused on defining spatio-temporal patterns and algorithms [23, 25, 30, 32, 38, 52]. Laube et al. defined several spatio-temporal patterns, such as leadership and convergence [33]. Query processing algorithms have been proposed to extract such patterns [33]. I hope to extend the proposed algorithm to mine these patterns.
Sustained Emerging Spatio-Temporal Co-occurrence Pattern Mining: A Summary of Results

Abstract

Sustained emerging spatio-temporal co-occurrence patterns (SECOPs) represent subsets of object-types that are increasingly located together in space and time. Discovering SECOPs is important due to many applications, e.g., predicting emerging infectious diseases, predicting defensive and offensive intent from troop movement patterns, and novel predator-prey interactions. However, mining SECOPs is computationally very expensive because the interest measures are computationally complex, datasets are larger due to the archival history, and the set of candidate patterns is exponential in the number of object-types. I propose a monotonic interest measure for mining SECOPs and a novel SECOP mining algorithm. Analytical and experimental results show that the proposed algorithm is correct, complete, and computationally faster than related approaches. Results also show
the proposed algorithm is computationally more efficient than naive alternatives.

4.1 Introduction

Sustained emerging spatio-temporal co-occurrence patterns (SECOPs) represent subsets of object-types that are increasingly located together in space and time. Formally, given a collection of Boolean spatio-temporal (ST) features (object-types) and their instances (objects) over a common ST framework, a neighborhood relation over neighbors, and interest measure thresholds, an SECOP mining algorithm aims to discover correct and complete sets of interesting and non-trivial SECOPs while minimizing computational cost.

Discovering and characterizing SECOPs is an important problem with many application domains [31], including public health (predicting emerging infectious disease), the military (battlefield planning and strategy), ecology (tracking species and pollutant movements), and homeland defense (looking for significant “events”, biodefense)

However, discovering SECOPs poses several challenges. The first challenge is that the process is computationally very expensive since the interest measures are computationally complex. The second challenge is creating and formalizing composite interest measures to mine interesting and non-trivial SECOPs, since current interest measures such as the spatial prevalence measure are not sufficient to mine such patterns [28, 44]. The third challenge is the exponential number of object-types in the set of candidate patterns. The fourth challenge is to develop computationally efficient algorithms to mine massive spatio-temporal datasets. This paper describes an approach which meets all of these challenges.

4.1.1 An Application Domain Example

SECOPs are of great concern in public health, where there is a frequent need to identify emerging infectious diseases in order to take timely action [17, 36]. Emerging infectious diseases (EIDs) are diseases whose incidence has increased within the past two decades and that threaten to increase in the near future [36]. EIDs can be caused by previously undetected microbes (i.e. SARS, AIDS), the evolution of previously known microbes (In-
fluenza), the spreading of known microbes to new locations or populations (West Nile Virus), and the re-emergence of old infections such as tuberculosis. The World Health Organization reported that approximately 26 percent of the 57 million annual deaths worldwide in 2002 were caused by infectious diseases, the second cause of death after cardiovascular disease [2]. Examples of EIDs throughout the world can be seen in Figure 4.1.

![Figure 4.1. Examples of EIDs. Adapted from [17]. Black circles represent newly emerging diseases and white circles represent re-emerging diseases](image)

Populations around the world have had to contend with a number of recent outbreaks of EIDs. In a 2005 outbreak of dengue fever in Singapore, the number of cases (instances) rose rapidly throughout the year. Because of the sudden emergence of this disease, hospitals were forced to cancel some surgeries to provide more beds for dengue patients [1]. Figure 4.2 shows the weekly dengue fever cases in 2005 in Singapore.

Scientists are trying to predict or to prevent EIDs. The main focus is to discover the patterns that have cause or effect on EIDs. For example, it is found out that there is a strong relationship between increase of dengue fever cases, stagnant water (i.e., flowerpots), and high temperature. These phenomena are highly likely to co-occur. Similarly, influenza and migrating birds (i.e., waterfowl, shorebirds) frequently co-occur. SECOPs are subsets of object-types e.g., \{dengue\_disease, stagnant\_water, high\_temperature\} and
An Application Domain Example

4.1.1 An Application Domain Example

Figure 4.2. Weekly dengue fever cases in 2005 in Singapore [1]

{influenza_disease, migrating_birds}, whose instances are increasing over time. Discovering such SECOPs is crucial for timely response to outbreaks of diseases and to be fully prepared for such outbreaks. For example, the dengue fever outbreak was controlled after eliminating stagnant water in Singapore; stagnant water is a breeding ground for mosquitoes transmitting the disease.

Another kind of emerging pattern can occur when certain invasive species are introduced at a location where they did not previously occur naturally. For example, the increase in the population of Brown Tree Snake that was accidentally transported to the snake-free Guam Island after World War II, increased the presence of insect pests, which has caused forest defoliation and decrease in crop yields [2]. The invasive Brown Tree Snake preyed upon most of the insectivorous birds, bats and lizards on the island. The decimation of insectivorous birds, bats and lizards caused an increase in presence of insect pests. It is found out that there is a strong relationship between increase of Brown Tree Snake population and increase of presence of insect pests. Discovering such SECOP i.e. {brown_tree_snake, insect_pests} is important to prevent harmful affects on the ecology and environment.
4.2 Related Work

The closest related work focuses on spatio-temporal episode "formation" to identify pattern characterizing evolution of spatial relationship objects (or features) over time [52]. A "formation" event occurs when the number of instances of a pattern changes from zero to non-zero. The algorithm to efficiently mine "formation" events is not specified explicitly. This model is limited due to several reasons. First "formation" is different from "emerging". While a sustained emerging pattern may imply a "formation", there may be a long time-lag between "formation" and emergence. Furthermore, a formation may not lead to sustained emergence. In addition, detection of "formation" may require tracking extremely rare patterns (i.e. pattern with very low number of instances), thus hampering prevalence-based filtering [7] and leading to exorbitant computational costs. Support and realization were used in [52] as interest measures to characterize the importance of the "formation" pattern. The support of "formation" pattern p is defined as the number of timeslots (snapshots) in the database where p occurs. The realization of "formation" pattern p is defined as the minimum of the number of instances of p in each timeslot. However, scaling the algorithm to large spatio-temporal databases is challenging since the interest measure used in [52], i.e. realization, is not monotonic.

In contrast, this paper defines sustained emerging co-occurrence patterns, formulizes new monotonic interest measures, and proposes algorithms to mine SECOPs from massive spatio-temporal datasets in a computationally efficient manner. Contributions: The paper makes the following contributions:

1. It defines sustained emerging spatio-temporal co-occurrence patterns (SECOPs) and the SECOP mining problem.

2. It proposes a new monotonic composite interest measure to discover and mine SECOPs.

3. It proposes a novel and computationally efficient SECOP mining algorithm (SECOPs-Miner).
4. It shows that the proposed algorithm is correct and complete in finding sustained emerging prevalent (e.g., spatial prevalent and time prevalent) SECOPs.

5. It experimentally evaluates the proposed composite interest measures and SECOP mining algorithms using real datasets.

4.2.1 Scope and Outline

This paper focuses on the sustained emerging co-occurrence pattern on a typed collection of moving objects extending interest measures for spatial co-location patterns given a user defined participation index threshold [28,44]. The following issues are outside the scope of this paper: (i) Determining thresholds for SECOP interest measure, (ii) similarity measures for tracking moving objects due to the focus on object-types rather than objects, and (iii) indexing and query processing issues related to mining objects.

The rest of the paper is organized as follows. Section 4.3 presents basic concepts and the problem statement of mining SECOPs. Section 4.4 presents the proposed SECOP mining algorithm. Analysis of the SECOP mining algorithm is given in Section 4.5. Section 4.6 presents the experimental evaluation and Section 4.7 discusses the conclusions and future work.

4.3 Basic concepts and problem statement

4.3.1 Spatial prevalence measure

The focus of this study is to discover sustained emerging spatio-temporal co-occurrence patterns (SECOPs) over a spatio-temporal framework and a neighborhood relation $R$. First I explain the modeling of groups of object-types in space, e.g., spatial co-locations, and then I explain modeling SECOPs and propose algorithms to mine SECOPs [44].

Spatial co-location mining algorithms are used to discover sets of object-types that are frequently located together in a spatial framework for a given set of spatial object-types, their instances and a spatial neighbor relationship $R$ [28,44]. For example, in Figure 4.3(a),
4.3.1 Spatial prevalence measure

(a) An input spatio-temporal dataset

(b) A set of output SECOPs

(c) Trends of spatial prevalence indices of SECOPs

Figure 4.3. An example spatio-temporal dataset

in time slot t=0, \{A.1, C.1\} is an instance of a co-location if the distance between the objects is less than or equal to a given neighborhood distance threshold. The lines show the distances between objects which satisfy the neighborhood distance threshold. The participation index is used to determine the strength of the co-location pattern, that is, whether the index is greater than or equal to a threshold [28,44]. Such a co-location pattern is called spatial prevalent. The participation index is defined as the minimum of the participation ratios (the fraction of the number of instances on object-types forming co-location instances to the number of instances). For example, in Figure 4.3(a), \{A, B\} is a co-location for time slot t=0, and its instances are \{A.1, B.1\} and \{A.2, B.2\}. In the dataset, object-type A has 4 instances and two of them (A.1 and A.2) are contributing to the co-location \{A, B\}, so the participation ratio of A for the co-location \{A,B\} is 2/4. The participation ratio of
object-type B is 2/5 since 2 out of 5 instances are contributing to form the instances of the co-location \{A, B\}. The participation index of the co-location \{A, B\} is 2/5, which is the minimum of the participation ratios of object-types A and B.

It has been shown that the participation index is anti-monotone in the size of co-locations [28, 44]. In other words, \(\text{participation index}(P_i) \leq \text{participation index}(P_j)\) if \(P_i\) is a subset of \(P_j\). In addition, [28, 44] show that the participation index has a spatial statistical interpretation as an upper bound on the cross-K function [14].

### 4.3.2 Modeling SECOPs

Given a set of spatio-temporal object-types and a set of their instances with a neighborhood relationship \(R\), an SECOP is a subset of spatio-temporal object-types whose instances are neighbors in space and time.

**Definition 4.3.1** Given a spatio-temporal pattern and a set \(T\) of time slots, such that \(T = [T_0, \ldots, T_{n-1}]\), the time prevalence or persistence measure of the pattern is the fraction of time slots where the pattern occurs over the total number of time slots.

For example, in Figure 4.3(a), the total number of time slots is 3 and pattern \{A, B\} occurs in all 3 time slots, so its time prevalence index is 3/3.

**Definition 4.3.2** Given a spatio-temporal dataset \(ST\), and a spatial prevalence threshold \(\theta_p\), the sustained emergence prevalence measure of a spatio-temporal pattern \(P_i\) is a composition of the spatial prevalence measure and the time prevalence measure where the spatial prevalence measure is getting stronger (monotonically increasing) over time, such as

\[
\text{Prob}_{t_m \in \text{all time-slot}}(s_{\text{prev}}(\text{pattern } P_i, \text{time-slot } t_m) \geq \theta_p) \quad \text{and} \quad s_{\text{prev}}(\text{pattern } P_i, \text{time-slot } t_m) \text{ monotonically increasing over time.}
\]
where $s_{\text{prev}}$ stands for spatial prevalence index, e.g., the participation index (described in section 2.1), of pattern $P_i$ and $\theta_p$ is a spatial prevalence threshold. Prob stands for the probability of overall prevalence time slots.

**Definition 4.3.3** Given a spatio-temporal dataset $ST$ and a threshold pair $(\theta_p, \theta_{\text{time}})$, a SECOP $P_i$ is a sustained emergence prevalent patterns, if its sustained emergence prevalence measure satisfies the following.

$$\Pr_{m \in \text{all time slot}} [s_{\text{prev}}(pattern \ P_i, \text{time slot } t_m) \geq \theta_p] \geq \theta_{\text{time}} \quad (4.2)$$

where $s_{\text{prev}}$ stands for spatial prevalence index, e.g., the participation index (described in section 2.1) of pattern $P_i$. Prob stands for the probability of overall prevalence time slots, $\theta_p$ is a spatial prevalence threshold, and $\theta_{\text{time}}$ is a time prevalence threshold.

For example, in Figure 4.3(a), $\{A, B\}$ is a SECOP because: (a) it is spatial prevalent in time slots $t=0$, $t=1$, and $t=2$ since its participation indices are above the participation index threshold 0.25; and (b) it is time prevalent since its time prevalence index, i.e., 1, is above the time prevalence index threshold 0.5. SECOPs are extensions of traditional spatial co-location patterns, whose instances are increasing over time.

### 4.3.3 Problem statement

**Given:**

- A set $P$ of Boolean spatio-temporal object-types over a common spatio-temporal framework.
- A neighbor relation $R$ over locations.
- A spatial prevalence threshold, $\theta_P$.
- A time prevalence threshold, $\theta_{\text{time}}$. 
Find: \( \{ P_i | P_i \text{ is a subset of } P \text{ and } P_i \text{ is a sustained emergence prevalent SECOP as 4.3.2} \} \).

Objective: Minimize computation cost.

Constraints: To find a correct and complete set of SECOPs.

Example: The spatio-temporal dataset given in Figure 4.3(a) contains 3 Boolean object-types, A, B, and C for 3 time slots. A distance between the objects may define the neighborhood relation R. For example, A.4 is a neighbor of C.2 in time slot 0, but not in time slots 1 and 2. In this example dataset \{A, B\}, \{A, C\}, \{B, C\}, and \{A, B, C\} form a candidate SECOP, given \( P = 0.25 \), and time = 0.5. Figure 4.3(c) gives the trend of the spatial prevalence indices, i.e. participation indices, of the SECOPs. As can be seen pattern \{A, B\} is above the threshold for the time interval [0,2] and rest of the patterns are above the threshold for time interval [1,2].

4.4 Mining SECOPs

In this section, I first discuss a na"ive approach to mining SECOPs and then propose a novel SECOP mining algorithm (SECOP-Miner).

4.4.1 Na"ive Approach

A na"ive approach can generate all spatial co-locations for each time slot and then can apply a post-processing step to discover sustained emergence prevalent co-occurrence patterns by checking their spatial and time prevalence indices. The na"ive approach will generate size \( k + 1 \) candidate co-location patterns for each time slot using size \( k \) subclasses until there are no more candidate spatial co-locations. After finding all spatial co-location patterns in each time slot, a post-processing step can be used to discover sustained emergence prevalent SECOPs by pruning out spatial and time non-prevalent co-location patterns. This approach will lead to unnecessary computational costs since it does not prune out sustained emergence non-prevalent SECOPs before the post-processing step.
4.4.2 SECOP-Miner

I propose a SECOP mining algorithm (SECOP-Miner) to discover sustained emergence prevalent SECOPs by incorporating a pruning step into the algorithm. It will generate size $k + 1$ candidate SECOPs using size $k$ sustained emergence prevalent subclasses. The participation index is used as a spatial prevalence interest measure to check if the pattern is spatial prevalent at a time slot [28]. The time prevalence (i.e., persistence measure in definition 2.1) is used as a time prevalence interest measure. First I give the pseudocode of the algorithm, and then I provide an execution trace of the algorithm using the spatio-temporal dataset from Figure 3(a).

Algorithm 4 gives the pseudocode of the SECOP-Miner algorithm. The inputs to the algorithm are a set of spatial event types $E$, a spatio-temporal dataset $ST$, a spatial neighborhood relationship $R$, and thresholds of interest measures such as, spatial prevalence and time prevalence. The output of the algorithm is a set of SECOPs.

In the algorithm, steps 1 and 2 include initialization, steps 3 through 12 give an iterative process to mine SECOPs, and step 13 gives a union of the results of the iterative steps. Steps 3 through 12 continue until there is no candidate SECOP to be generated (mined). The functions of the algorithm are explained below.

**Generation of candidate co-occurrence patterns (step 5):** This function uses an apriori-based approach to generate size $k + 1$ candidate co-locations $C_{k+1}$ for each time slot, using all sustained emergence prevalent size $k$ SECOPs $EP_k$ [7].

**Generation of spatial co-occurrence instances (step 6):** The instances of candidate $C_{k+1}$ are generated by joining neighbor instances of sustained emergence prevalent size $k$ patterns for each time slot. This is similar to the instance generation step of the co-location miner algorithm [28].

**Finding spatial prevalent co-location patterns (step 7):** All spatial prevalent size $k + 1$ patterns $SP_{k+1}$ are found by pruning the patterns whose spatial prevalence indices, i.e., participation indices, are less than a given threshold for each time slot. Computation of participation indices follows the same algorithmic ideas as those in the co-location mining
Algorithm 4  SECOP-Miner

**Inputs:**
- \( E \): a set of distinct spatial object-types
- \( ST \): a spatio-temporal dataset <object type, object id, x, y, time slot
- \( R \): spatial neighborhood relationship
- \( TF \): a time slot frame \( \{ t_0, ..., t_{n-1} \} \)
- \( \theta_p \): a spatial prevalence threshold
- \( \theta_{time} \): a time prevalence threshold

**Output:** Sustained emerging spatio-temporal co-occurrence patterns (SECOPs) spatial prevalence indices, i.e., participation indices, are no less than \( \theta_p \), for time prevalence indices are no less than \( \theta_{time} \)

**Variables:**
- \( k \): co-occurrence size
- \( t \): time slots \( (0, ..., n-1) \)
- \( T_k \): set of instances size \( k \) co-occurrences
- \( C_k \): set of candidate size \( k \) co-occurrences
- \( SP_k \): set of spatial prevalent size \( k \) co-occurrences
- \( TP_k \): set of time prevalent size \( k \) co-occurrences
- \( EP_k \): set of mixed-drove size \( k \) co-occurrences

**Algorithm:**

1. Initialize parameters
2. co-occurrence size \( k = 1 \), \( C_k = E, EP_k(0) = ST \)
3. while (not empty \( EP_k \)) {
4.   for each time slot \( t \) in \( (0, ..., n-1) \) {
5.     \( C_{k+1} = \text{gen candidate co-occur}(C_k, EP_k, time slot) \)
6.     \( T_{k+1} = \text{gen co-occur instance}(C_{k+1}, T_k, \text{time slot}, R) \)
7.     \( SP_{k+1} = \text{find spatial - prevalent co-occur}(T_{k+1}, C_{k+1}, TF, \theta_p) \)
8.   }
9. \( TP_{k+1} = \text{find time prevalence index}(SP_{k+1}) \)
10. \( EP_{k+1} = \text{find time - prevalent co-occur}(TP_{k+1}, \theta_{time}) \) }
11. \( k = k + 1 \)
12. }
13. return union \( \{ EP_2, ..., EP_{k+1} \} \)

In steps 5 through 7, the algorithm finds size \( k + 1 \) spatial prevalent co-location patterns for each time slot.

**Finding time prevalence index (step 9):** This step checks the behavior of the spatial prevalence index of a pattern over time, which can be classified into three categories: 1) monotonically increasing, such that the prevalence index is getting stronger over time, 2) monotonically decreasing, such that the prevalence index is getting weaker over time, and 3) has extremums oscillating over time. To find SECOPs, the algorithm checks the behavior of the spatial prevalence index of each size \( k + 1 \) co-occurrence pattern. If the spatial prevalence index of a pattern is monotonically increasing over time, the pattern is recorded in candidate SECOP \( TP_{k+1} \); otherwise it is eliminated. If it is monotonically decreasing
over time (that is, the participation index is getting weaker), the pattern is eliminated and is not included in the set $TP_{k+1}$. If it has extremums (one or more roots), it is divided into time intervals such that the spatial prevalence index is monotonically increasing or decreasing. The monotonically decreasing parts are eliminated but the rest are saved as candidate SECOPs. After the elimination of the irrelevant patterns, the time prevalence indices of candidate SECOPs are calculated.

**Finding sustained emerging co-occurrence patterns (step 10):** This step discovers SECOPs by checking the time prevalence indices of the patterns if they are above or equal to a given time prevalence threshold time. The patterns whose time prevalence indices do not satisfy the given threshold are pruned. The remaining patterns will be sustained emergence prevalent SECOPs and will be used to generate candidate supersets of the SECOPs in step 5. The algorithm will run iteratively until there are no candidate SECOPs to be generated. The algorithm outputs the union of all size sustained emergence prevalent SECOPs.

**An Execution Trace:** The execution trace of the algorithm is given in Figure 4.4 using the spatio-temporal dataset given in Figure 4.3(a). The dataset contains three object-types A, B, and C and their instances in three time slots. A has 4 instances, B has 5 instances, and C has 3 instances. Each instance has a unique identifier, such as A.1. Some of the patterns of these object-types form a SECOP. To discover SECOPs I compute the sustained emergence prevalence measure, which is a composition of the spatial prevalence and time prevalence measures. The spatial prevalence measure, (participation index) shows the strength of the spatial co-location pattern and whether the index is greater than or equal to a given threshold. The time prevalence measure, (time prevalence index), shows the frequency of the pattern over time.

In Figure 4.4(a), candidate spatial co-location pairs of the object-types and their instances are generated for distinct time slots and then the spatial co-location patterns whose participation indices are less than a given threshold are pruned since they are spatial non-prevalent. A spatial non-prevalent co-location pattern \{B, C\} is pruned in time slot t=0
Figure 4.4. An execution trace of mining sustained emerging spatio-temporal co-occurrence patterns (SECOPs) because its participation index is less than the given threshold 0.25.

SECOPs whose participation indices are increasing over time are then determined and their time prevalence indices are calculated. For example, in Figure 4.4(b), the time prevalence index of pattern \{A, B\} is 3/3 because it is spatial prevalent in all time slots and its participation indices increase monotonically over time. The time prevalence indices of pattern \{A, C\} and \{B, C\} are 2 since they are SECOPs in time interval [1,2]. Pattern \{A, C\} is not a SECOP in time interval [0,1] since its participation index is decreasing from time slot t=0 to time slot t=1. Pattern \{B,C\} is not a SECOP in time interval [0,1] since it is not spatial prevalent in that interval. The SECOPs whose time prevalence indices are greater than or equal to a given time prevalence index threshold are selected for generating superset candidate patterns. Spatial prevalent patterns \{A, B\}, \{A, C\} and \{B, C\} are selected as sustained emergence prevalent SECOPs since they are also time prevalent (they exceed the time prevalence index of 0.5). Using SECOPs \{A,B\}, \{A, C\}, and \{B, C\}, the next candidate pattern \{A, B, C\} is generated. The next step is to generate instances of candidate pattern \{A, B, C\} in time slots where its subsets exist and to check its participation indices in the corresponding time slots. Since all subsets of SECOP \{A, B, C\} are sustained emergence prevalent and exist in time slots t=1 and t=2, there is no need to generate
instances for time slot $t=0$. The instances of candidate SECOP $\{A,B,C\}$ are generated and participation indices of pattern $\{A, B, C\}$ are found, which are $2/5$ and $2/4$ for time slots $t=1$ and $t=2$ respectively (Figure 4.4(b)). Since both participation indices are greater than the spatial prevalence threshold 0.25, pattern $\{A, B, C\}$ is spatial prevalent in these time slots.

In Figure 4.4(b), SECOP $\{A,B,C\}$ is determined since its spatial prevalence indices, i.e., participation indices, are increasing over time over given threshold 0.5. Since there are not enough subsets to generate the next candidate SECOPs, the algorithm stops at this stage and outputs the union of all sustained emergence prevalent SECOPs: $\{A\}$, $\{A, C\}$, $\{B, C\}$, and $\{A, B, C\}$. Figure 4.3(c) gives the trend of the spatial prevalence indices, i.e. participation indices, of the SECOPs. As can be seen pattern $A, B$ is above the threshold for the time interval $[0,2]$ and rest of the patterns are above the threshold for time interval $[1,2]$.

### 4.5 Analysis of the SECOP mining algorithm

#### 4.5.1 Sustained emergence prevalence measure is monotonic

**Lemma 4.5.1** A chosen spatial prevalence measure, such as, participation index, is monotonically non-increasing in the size of the SECOPs at each time slot.

**Proof** Let a SECOP $P_i$ be a subset of a SECOP $P_j$. Then, This follows from the anti-monotone property of the participation index for co-location patterns using the data subset for time slot $t$ [28]. If an instance $I$ of object-type $O$ in intersection $(P_j, P_i)$ participates in any instance of $P_j$, $I$ must participate in some instance of $P_i$ as well. Thus,

$$\text{participation\_index}(O, P_j, t) \leq \text{participation\_index}(O, P_i, t)$$  \hspace{1cm} (4.3)

for all object-types $O \in \text{intersection}(P_j, P_i)$. This implies the lemma.

**Lemma 4.5.2** A sustained emergence prevalence measure is monotonically non-increasing...
with the size of SECOP over space and time. In other words, if SECOP $P_i$ is a subset of SECOP $P_j$ then

\[ \text{Prob}_{t_m \in \text{all time slot}}(s_{\text{prev}}(\text{pattern } P_i, \text{time_slot } t_m) \geq \theta_p) \text{ and } s_{\text{prev}}(\text{pattern } P_i, t_m) \text{ monotonically increasing over time} \] 

\[ \text{Prob}_{t_m \in \text{all time slot}}(s_{\text{prev}}(\text{pattern } P_j, \text{time_slot } t_m) \geq \theta_p) \text{ and } s_{\text{prev}}(\text{pattern } P_j, t_m) \text{ monotonically increasing over time} \]

where Prob stands for the probability of overall prevalence time units, $s_{\text{prev}}$ stands for spatial prevalence, $\theta_p$ is the spatial prevalence threshold, and $t_m$ is the time slot.

**Proof** Let

\[ TS(P_j, \theta_p) = \{t_m|\text{participation\_index}(P_j, t_m) \geq \theta_p\} \text{ and } \text{participation\_index}(P_j, t_m) \text{ monotonically increasing over time} \]

Lemma 4.5.1 implies that the $\text{participation\_index}(P_j, t_m) \geq \theta_p$ for all $t_m \in TS(P_j, \theta_p)$ and is monotonically increasing, since $P_i$ is a subset of $P_j$. Thus,

\[ \text{Prob}_{t_m \in \text{all time slot}}[s_{\text{prev}}(\text{pattern } P_i, \text{time_slot } t_m) \geq \theta_p] \geq \theta_{\text{time}} \text{ and } s_{\text{prev}}(\text{pattern } P_i, t_m) \text{ monotonically increasing over time} \]

where $\theta_{\text{time}}$ is time prevalence threshold. The participation ratio and participation index have anti-monotonic properties as the number of co-occurrences increases, and both have been successfully used in previous studies [28].
4.5.2 Correctness and completeness

**Theorem 4.5.1** The SECOP-Miner algorithm is complete.

**Proof** The SECOP-Miner is complete if it finds all sustained emergence prevalent SECOPs that satisfy a given spatial prevalence threshold and time prevalence threshold. I can show this by proving that none of the functions of the algorithm miss any patterns, i.e., filter out a prevalent SECOP.

The `gen_candidate_co-occur` function does not miss any patterns given the anti-monotonic nature of the SECOP interest measure. The input to this function is the sustained emergence prevalent size $k$ SECOPs and the output is candidate size $k + 1$ SECOPs. If $c_1 = f_1, \ldots, f_k$ and $c_2 = f_1, \ldots, f_{k-1}, f_{k+1}$ are size $k$ sustained emergence prevalent co-occurrence patterns, candidate size $k + 1$ pattern $C_{k+1} = f_1, \ldots, f_{k-1}, f_k, f_{k+1}$ will be produced by joining sustained emergence prevalent size $k$ SECOPs.

The `gen_co-occur_instance` function does not miss any patterns. This function generates instances of candidate size $k + 1$ SECOPs by joining instances of sustained emergence prevalent size $k$ SECOPs if they are in the neighborhood distance and forming a clique.

The `find.spatial-prevalent_co-occur` function does not miss any patterns. It calculates spatial prevalence indices of the patterns for each time slot and finds spatial prevalent patterns whose participation indices are greater than a given participation index threshold.

The `find.time-prevalence_index` function does not miss any patterns. This function calculates time prevalence indices of the patterns found in steps 4 through 8 and does not do any pruning.

The `find.emergence-prevalent_co-occur` function does not miss any SECOPs. The function finds all sustained emergence prevalent SECOPs whose time prevalence index is greater than or equal to a given time prevalence threshold. SECOPs whose time prevalence indices do not satisfy the given threshold are pruned.

**Theorem 4.5.2** The SECOP-Miner is correct. In other words, if an SECOP pattern $P$ is returned by the SECOP-Miner algorithm, then $P$ is a sustained emergence prevalent
**SECOP.**

**Proof** This is easy to establish due to the pruning steps of "find_spatial_prevalent_co-occur" and "find_emergence_prevalent_co-occur" which weed out candidates not meeting the given thresholds.

### 4.6 Experimental evaluation

In this section, I present experimental evaluations of several design decisions and workload parameters of SECOP-Miner algorithm. I use a real-world training dataset. I evaluated the behavior of the SECOP-Miner algorithm and naive approach by changing the number of time slots, the number of object-types, and the value of the spatial prevalence and time prevalence measures. Figure 4.5 shows the experimental setup to evaluate the impact of design decisions of the performance of the proposed algorithms. Experiments were conducted on an Intel Centrino PIV 1.6 GHz computer with 512 MB of RAM.

![Figure 4.5. Experimental setup and design](image)

The training dataset contains location and time information of vehicle moving objects. This dataset includes 14 time snapshots and 20 distinct object-types and their instances in each time slot. The minimum instance number is 2, the maximum instance number is 78, and the average number of instances is 18. Figure 4.6 shows an instance of a SECOP where object_1 and object_2 are coming together, moving from top right to bottom left. Initially the objects are far away from each other but they get relatively close to each other, that is,
the pattern emerges over time. Such a pattern may be of interest to a planner if it indicates an emerging imminent maneuver by object_1 under the protection of object_2.

Figure 4.6. One instance of a SECOP

Figure 4.6 gives a statistical summary of the participation index of a SECOP object_1, object_2 for 10 time slots. The participation of the SECOP increases over time between time intervals [4,5] and [6,9]. If the time prevalence threshold is 0.3, the time interval [4,5] will be pruned and the time interval [6,9] will be used for further data generation.

Figure 4.7. Statistical summary of participation index of SECOP
4.6.1 Effect of number of time slots

In the first experiment, I evaluated the effect of the number of time slots on the execution time of the SECOP-Miner algorithm and naive approach. The participation index threshold, time prevalence index threshold, and neighborhood distance threshold were set at 0.2, 0.5, and 100m respectively. As shown in Figure 4.8, the execution time of the SECOP-Miner algorithm outperforms the naive approach, since it prunes out emergence non-prevalent SECOPs early. It can also be seen that, as the number of time slots increases, the execution time for the naive approach increases drastically.

![Figure 4.8. Effect of number of time slots](image)

4.6.2 Effect of number of object-types

In the second experiment, I evaluated the effect of the number of object-types on the execution time of two algorithms. The fixed parameters were the participation index threshold, time prevalence index threshold, and neighborhood distance and their values were 0.2, 0.5, and 100m respectively. As shown in Figure 4.9, the SECOP-Miner algorithm outperforms the naive approach as the number of object-types increases. It can also be seen that, as the number of time slots increases, the execution time for the naive approach increases drastically.
4.6.3 Effect of time prevalence index threshold

In the third experiment, I evaluated the effect of the time prevalence index threshold value on the execution time of both algorithms. The fixed parameters were participation index threshold, number of time slots, and neighborhood distance and their values were 0.2, 11, and 150m respectively. The effective cost of generation of spatial prevalent co-location patterns on the execution time of the naive approach will be constant since it generates the same number of spatial prevalent patterns as the time prevalence index increases. In that case, the cost of the post-processing step will reflect the trend of the naive approach as the time prevalence index threshold increases. Experimental results show that the SECOP-Miner algorithm is more computationally efficient than the naive approach because of the early pruning strategy (Figure 4.10).

4.6.4 Effect of spatial prevalence index threshold

In the fourth experiment, I evaluated the effect of the value of the spatial prevalence index threshold on the execution times of both algorithms. The fixed parameters are time prevalence index threshold, number of time slots, and distance threshold; the values are 0.5, 11, 150m respectively. As can be seen in Figure 4.11, the SECOP-Miner algorithm outperforms the naive approach as the spatial prevalence index threshold increases.
4.7 Conclusions and future work

I defined sustained emerging spatio-temporal co-occurrence patterns (SECOPs) and the SECOP mining problem and proposed a new monotonic composite interest measure, the sustained emergence prevalence measure, which is the composition of the spatial prevalence and time prevalence measures. I also devised a novel, computationally efficient SECOP mining algorithm, SECOP-Miner, for mining these patterns. I compared the proposed algorithm with the naive approach, which finds all spatial co-locations of each time slot and then discovers SECOPs by applying sustained emergence prevalence to prune irrelevant co-locations using a post-processing step. I proved that the proposed algorithms
are correct and complete in finding sustained emergence prevalent (e.g., spatial-prevalent and time prevalent) SECOPs. The experimental results using a real dataset provide further evidence of the viability of the proposed approach.

There are also other studies defining the emerging pattern mining problem in classical data mining, notably [16]. However, the problem defined by Dong et. al. is different than what I am dealing with [16]. The problem they define is to capture the significant differences between two classes, i.e., normal tissues vs cancer tissues. The significant change is measured by a growth rate ratio (ratio of the supports of two classes) [16,48]. The proposed approaches in [16,48] are not applicable to the mining of SECOPs from spatio-temporal datasets since they cannot handle the spatial and temporal characteristics of spatio-temporal datasets.

Other studies in the literature, however, have defined spatio-temporal patterns of interest to us. For example, Kalnis et. al. proposed a moving clusters problem and clustering-based algorithms to mine this pattern [30]. Hadjieleftheriou et. al. defined spatio-temporal pattern queries which can use various types of spatial predicates (range search, nearest neighbor, etc) associated with temporal constraints (time-instant or time-interval) and proposed spatio-temporal index structures and algorithms [25]. Gudmundsson proposed algorithms to query flock, leadership, convergence, and encounter patterns from spatio-temporal databases [23]. Yoo et. al. proposed a method to query co-evolving spatial event sets [56]. In the future I plan to apply my algorithms to mine these patterns. I also plan to investigate other pruning methods and test the proposed algorithm on different datasets and to develop new computationally efficient algorithms for mining SECOPs.
Conclusions and Future Work

In this dissertation, I proposed a new taxonomy of the spatio-temporal (ST) co-occurrence patterns. I defined new spatial and ST co-occurrence patterns and new composite monotonic interest measures, designed computationally efficient algorithms, and evaluated proposed algorithms using real and synthetic datasets.

I defined the problem of zonal co-location pattern discovery where zonal co-location patterns represent subsets of object-types that are frequently located in a subset of space (i.e., zone). I proposed an indexing structure that stores co-location instances to discover zonal co-location patterns efficiently for repeated specifications of the parameter values, e.g., interest measure thresholds and zones, and experimentally evaluated the proposed algorithms.

I defined mixed-drove spatio-temporal patterns (MDCOPs) that represent subsets of different object-types whose instances are located close in geographic space for a significant fraction of time. I proposed a new composite interest measure and novel, computationally efficient algorithms for mining MDCOPs. The proposed methods have been evaluated analytically and experimentally.

I analyzed the trends of mixed groups of moving objects. I defined sustained emerging spatio-temporal co-occurrence patterns (SECOPs) that represent subsets of object-types
that are increasingly located together in space and time. I proposed a monotonic interest measure for mining SECOPs and a novel SECOP mining algorithm.

5.1 Future Work

In the short term, I would like to advance the solutions of the problems that I am currently dealing with.

- I would like to extend my work for incremental (or on-line) pattern mining and to explore new efficient pruning strategies, data structures, and algorithms to support the repetitive mining process of spatio-temporal co-occurrence patterns for different user-defined thresholds.

- I plan to extend proposed zonal co-location pattern mining algorithm for the discovery of spatio-temporal patterns [9, 13].

- I would like to explore the relationship between the proposed composite interest measures and spatio-temporal statistical measures of interaction [8].

- Another problem of interest is the characterization of the probability distribution of the proposed interest measures to help in making the choice of thresholds in the proposed measures.

- I am interested in adapting the problems that I have been working on for emerging data mining applications.

In the long term, I would like to define new interest measures and patterns, design new algorithms for the remaining categories of the proposed taxonomy. I would like to develop techniques to mine cascade and periodic co-occurrence patterns.

In broader manner, I would like to continue working in the fields of location-based services, data mining, and databases. Recently several grand research challenges have emerged. Four of these challenges can be listed as
5.1 Future Work

- discovering patterns and anomalies from enormous frequently updated spatial and spatio-temporal datasets,
- addressing the representation of real geographic phenomena in digital form,
- developing an ontological framework for spatial and spatio-temporal analysis, and
- integrating spatial and spatio-temporal data from multiple agencies, distributed data, and multi-scale data [4, 5, 19, 50].

In the years to come, I would like to focus on finding solutions for these challenges.
Bibliography


[27] J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based multidimensional data mining. *IEEE Comp.*, 32(8), 1999.


