University of Alberta

Statistically Sound Interaction Pattern Discovery from Spatial Data

by

Sajib Barua

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Spring 2014
Edmonton, Alberta

To my parents

Abstract

Spatial interaction pattern mining is the process of discovering patterns that occur due to the interaction of Boolean features from a spatial domain. A positive interaction of a subset of features generates a co-location pattern, whereas a negative interaction of a subset of features generates a segregation pattern. Finding interaction patterns is important for many application domains such as ecology, environmental science, forestry, and criminology.

Existing methods use a prevalence measure, which is mainly a frequency based measure. To mine prevalent patterns, the known methods require a user defined prevalence threshold. Deciding the right threshold value is not easy and an arbitrary threshold value may result in reporting meaningless patterns and even not reporting meaningful patterns. Due to the presence of spatial auto-correlation and feature abundance, which are not uncommon in a spatial domain, random patterns may achieve prevalence measure values higher than the used threshold just by chance, in which case the existing algorithm will report them. To overcome these limitations, we introduce a new definition of interaction patterns based on a statistical test. For the statistical test, we propose to design an appropriate null model which takes spatial auto-correlation into account. To reduce the computational cost of the statistical test, we also propose two approaches.

Existing mining algorithms also use a user provided distance threshold at which the algorithm checks for prevalent patterns. Since spatial interactions, in reality, may happen at different distances, finding the right distance threshold to mine all true patterns is not easy and a single appropriate threshold may not even exist. In the second major contribution of this thesis, we propose an algorithm to mine true co-locations at multiple distances. Our approach does not need thresholds for the prevalence measure and the interaction distance. An approximation algorithm is also proposed to prune redundant patterns that could occur in a statistical test. This algorithm finally reports a minimal set of patterns explaining all the detected co-locations. We evaluate the efficacy of our proposed approaches using synthetic

and real data sets and compare our algorithms with the state-of-the-art co-location mining approach.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Jörg Sander, for the continuous support of my Ph.D. study and research, for his patience, encouragement, enthusiasm and immense knowledge. I feel very privileged to have had him as my advisor. He taught me a great deal about the field of spatial data mining and spatial statistics by sharing with me the joy of discovery and investigation that is at the heart of research. His insightful criticism and review helped me to conduct solid research.

I would like to thank Dr. Shashi Shekhar, who as a part of my examining committee gave me valuable comments and suggestions that helped to improve the quality of my thesis. My sincere thanks also goes to Dr. Ian Parsons, who was also part of my supervisory committee and gave useful feedback especially during our bi-weekly research meetings and during my candidacy and final oral examination. I extend my gratitude to Dr. Mario Nascimento, Dr. Osmar R. Zaïane, Dr. Davood Rafiei, and Dr. Arturo Sanchez-Azofeifa who were part of my examining committee at different stages of my thesis. Their feedback has significantly contributed to the improvement of my thesis. Thanks are also due to Dr. José Nelson Amaral who helped me to start a Ph.D. at the University of Alberta.

I would like to thank Abhishek Srivastava, who as a good friend was always willing to help and give his best suggestions. I am also thankful to several fellow Ph.D. students and colleagues: Evandro De Souza, Han Liang, Pirooz Chubak, Reza Sadoddin, and Davoud Moulavi. I will always cherish their friendship and support.

I dedicate this thesis to my father (Samiran Barua) and mother (Reba Barua). I feel very fortunate to have parents that share my enthusiasm for academic pursuits. They have been extremely understanding and supportive of my studies and longed to see this achievement come true. I am very much indebted to them. A big thanks to my wife Anupa who was by my side cheering me on during the final stage, the toughest part, of my Ph.D. study. I would also like to thank my brother Rajib and my uncle Ashish Barua who were always encouraging me with their best wishes.

Finally, I would like to thank the Department of Computing Science at the University of Alberta and Syncrude who provided me with financial and research support.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ARM	Association Rule Mining
CPMNDC	Co-location Pattern Mining with No Distance Constraint
CSR	Complete Spatial Randomness
maxPI	MAXimum Participation Index
PCF	Pairwise Correlation Function
PDA	Personal Digital Assistant
PI	Participation Index
PIC	Pattern Instance Count
PID	Pattern Instance Distance
RCM	Reverse Cuthill-Mckee
SPACE	Spatial Point pAttern ClustEring
SSCSP	Statistically Significant Co-location and Segregation Pattern
ST	SpatioTemporal

Chapter 1

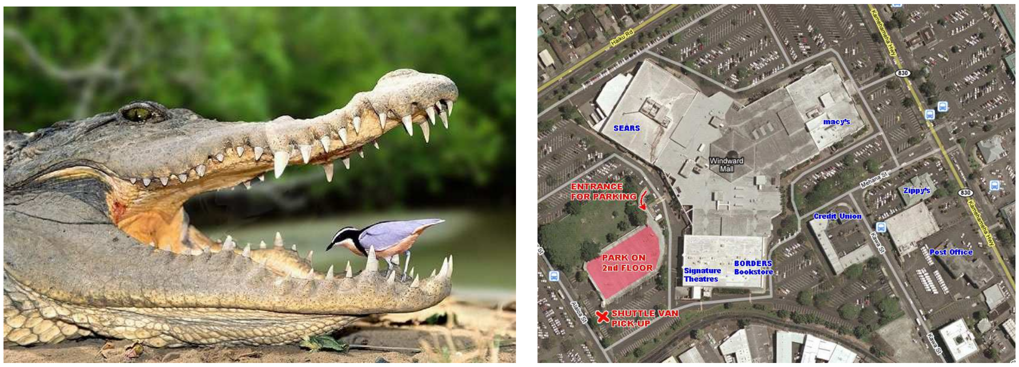
Introduction

The increasing use of geographically distributed, rich and massive spatial data poses an increasing scientific challenge in effective mining of interesting and useful but implicit spatial patterns. In this thesis, we focus on a challenging problem that exists in many application domains such as ecology, forestry, urban planning, criminology where domain scientists look for interaction patterns occurring due to some form of spatial dependency among Boolean spatial features. A Boolean spatial feature could be either present or absent for a given spatial location. Examples of Boolean spatial features could be species (trees or animals), catastrophic events (earthquakes, tsunamis, forest fires), climatological events (droughts, high precipitation, El Nino), urban features (residences, hospitals, schools, restaurants, bars), and crime events (assaults, drunk driving, robberies).

In spatial domains, interaction between Boolean spatial features gives rise to two types of interaction patterns. A positive interaction or an aggregation brings a subset of features close to each other whereas a negative interaction or an inhibition results in subsets of features segregating from each other. Co-location patterns have been defined as subsets of Boolean spatial features whose instances are often seen to be located at close spatial proximity [35]. Whereas segregation patterns are subsets of Boolean spatial features whose instances are infrequently seen to be located at close spatial proximity (i.e., whose co-locations are "unusually" rare). Interaction pattern mining can lead to important domain related insights in areas such as biology, epidemiology, earth science, and transportation.

1.1 Illustrative Application Domains

In nature, a symbiotic relationship among different species brings them to live close to each other and generates co-location patterns. When a Nile crocodile needs teeth cleaning, it opens up its mouth widely to let the Egyptian plover (a bird) in and then the bird cleans out the crocodile's teeth [40, 57]. Thus the bird works as the crocodile's dental floss and gets food in return. As a consequence, the Nile crocodile and the Egyptian plover are often seen to be co-located, which gives rise to a co-location pattern {Nile crocodile, Egyptian plover} (shown in Fig. 1.1(a)). A similar example of a co-location due to symbiotic relationship is the co-location of the hermit crab and the sea anemone [28]. In urban areas, we also see co-location patterns such as {shopping mall, parking} (shown in Fig. 1.1(b)), {shopping mall, restaurant}. In ecological domains, events observed in different but nearby locations can generate co-location patterns. For instance, the smoke aerosol index of a location influences the likelihood of rainfall in nearby locations [60]. In criminology, identifying crime attractors or generators is important for public safety. An example of co-locations in this domain consists of bar-closings, assaults, robberies, and drunk driving events which likely co-occur together at nearby locations [44]. Co-location patterns are also seen in *temporal* and *spatiotemporal* (*ST*) domains, for instance, homicides followed by a suicide event within a short period. Our research focuses only on spatial domains but can be extended for ST domains.



(a) Symbiotic relationship of crocodile and plover [59] (b) A shopping center with a surrounding parking lot [48]

Figure 1.1: Examples of co-location patterns.

In addition to co-location patterns, segregation patterns are also common in ecology, where they arise as a manifestation of processes such as the competition between plants or the territorial behavior of animals. For instance, in a forest some tree species are less likely found together at a particular distance from each other due to their competition for resources (such as minerals and sunlight) and for the space required for the stem and canopy growth. In evolutionary ecology, niches of species are seen segregated in the way of using different habitats, and different food resources. Segregation of similar species may happen due to the result of natural selection. Interspecific segregation prevents interspecific hybridization. The off-springs from interspecific hybridization (from two species within the same genus) are found very often sterile due to the differences in their chromosome structure, which prevent appropriate pairing during meiosis. Segregation patterns are seen among shorebirds. Shorebirds with long legs and long bills such as dowitchers can feed in slightly deeper water. Semipalmated sandpiper is also a shorebird with a short bill and short legs. This small shorebird roams at less deeper water or water edges to collect food. These two types of shorebirds are often seen separated from each other for their food and thus present an example of segregation pattern [52]. In astronomy, elliptical galaxies (early galaxies) are not seen together with spiral galaxies (late galaxies) [5]. These two types of galaxies is also another example of a segregation pattern.

1.2 Current State-of-the-Art

Existing co-location mining algorithms are inspired by the concept of Association Rule Mining (ARM) [2]. Given a set of transactions from market basket data, ARM looks for sets of items that are purchased together frequently; for example, bread and butter. The algorithm counts the number of times an item set (such as, {bread and butter}) occurred in some transactions and reports an item set if the frequency (*support*) of occurrence in transactions is higher than a user defined threshold (a *support threshold*) value. One important property of this notion is that an item set can not be frequent if its subsets are not frequent, which helps to reduce the search space in mining item sets of different sizes. Interaction pattern mining has been considered as a similar problem where we look for groups of features that are spatially interacting based on a neighbor relationship. Such a group of features if observed frequently is treated as a co-location.

Unlike the market basket data, there is no natural notion of a transaction in a spatial domain [35]. Yet most of the co-location mining algorithms [15, 35, 73] adopt an approach similar to the Apriori algorithm proposed for ARM in [2], by introducing some notion of transaction over the space. To "transactionize" a spatial data set, Shekhar *et al.* in [60] discuss three models - *feature centric model*, *window centric model*, and *event centric model*. Feature centric model is proposed by Koperski *et al.* in [38]. Window centric model is used in spatial statistics for exploratory spatial data analysis. For instance, the distribution of plants over a large area is studied by taking a sample from the whole population. For that, ecologists place a grid over the study area and prepare a sample from the plants found in a grid cell (i.e. window). Such method is also known as the Quadrat method [17]. Shekhar *et al.* propose an event centric model [60] which is the current state-of-the-art provides a neighborhood graph based methodology to eliminate the need for generating transactions from a spatial data. In an event centric model, a transaction is generated from a proximity neighborhood of feature instances. A proximity neighborhood is defined based on a spatial relationship such as metric relationship (e.g. Euclidean distance). Feature instances present in such a neighborhood become neighbors of each other and form a clique. Similar to the *support* measure of the ARM algorithm, a prevalence measure called *Participation Index (PI)* is proposed that is anti-monotonic and helps to prune the search space of prevalent co-location patterns.

A positive association among the spatial features results in co-location pattern, whereas a negative association or a repulsion causes segregation pattern. However unlike co-location patterns, segregation patterns have not received much attention in the spatial data mining communities. Munro *et al.* in [46] first introduce a mixed type of interaction (a combination of positive and negative), called "complex pattern" and later Arunasalam *et al.* in [5] propose a complex pattern mining algorithm using a prevalence measure called *maximum participation Index ($maxPI$)*. Huang *et al.* in [34] first proposed $maxPI$ which is used to find co-locations with rare events. *maxPI* has a weak anti-monotonic property and can be used to reduce the search space of prevalent pattern mining.

1.2.1 Limitations of the Existing Approaches

In existing co-location mining algorithms [35, 60, 69, 71, 72], a subset of features is declared as a prevalent co-location pattern and finally reported, if its PI-value is greater than a user specified threshold. The complex pattern mining algorithm proposed in [5] defines a subset as prevalent if its $maxPI$-value is greater than a user defined threshold. Finding a prevalent pattern based on either a given PI-threshold or a given $maxPI$-threshold, is reasonably efficient since the PI is anti-monotonic and the $maxPI$ is weakly anti-monotonic.

However, proper threshold selection for these methods is very critical. With a small threshold value, meaningless patterns could be reported and with a large threshold value, meaningful patterns could be missed. Unfortunately, the threshold is a user specified parameter that is domain specific and typically difficult to set. Hence using the existing threshold based approaches may not be meaningful from an application point of view. Another drawback of the existing approaches is that they use a single threshold to mine patterns of different sizes. This increases the chance of missing meaningful patterns as the pattern size becomes larger. We argue that the prevalence measure threshold should not be global and pre-defined, but should be decided based on the distribution and the total number of instances of each individual feature involved in an interaction. Spatial auto-correlation and feature abundance which are not uncommon in a spatial domain, may mislead an existing approach in mining prevalent patterns. Due to the presence of spatial auto-correlation and feature abundance, random patterns may achieve prevalence measure values higher than the used threshold just by chance, in which case the existing algorithm will report them.

Besides the prevalence measure threshold, interaction neighborhood information is another pre-requisite of the existing algorithms in mining prevalent patterns. Neighborhood information is given in the form a distance threshold which is the maximum inter-distance of instances of any two participating features of a pattern. For a given distance threshold, these algorithms aim to find all prevalent patterns. Determining a suitable distance threshold to mine prevalent patterns is not easy for many spatial domains. In reality, spatial interactions between features occur at multiple distances. Hence the use of one single distance threshold to mine all true patterns is a severe limitation. Even if the interaction distances are known ahead, existing algorithms might report random subsets of features as prevalent. To mine a true pattern which occurs in a large neighborhood, existing algorithms require to use a large distance threshold value. While using such a large distance threshold value, a random subset which has features with large number of instances can attain a high prevalence measure value; hence may be reported as prevalent.

1.3 Challenges

In spatial data sets, the value of a prevalence measure like the PI is not necessarily, whether high or low, indicative of a positive or negative interaction between features. It is not uncommon to see subsets of features with a very high prevalence measure value due to randomness, presence of spatial autocorrelation, and abundance of feature instances alone, i.e., without true interaction between the involved features. It is also possible that the prevalence measure value of a group of positively interacting features is relatively low if one of the participating features has a low participation ratio. There are similar issues with negative interactions. Not every negative interaction has a low participation index in absolute terms, and not every pattern with a low prevalence measure value represents necessarily a segregation pattern e.g., non-interacting features with few instances may also have a very low PI-value. Clearly, in such cases, the existing co-location mining algorithms will report meaningless "prevalent" patterns or miss meaningful patterns, they may even report a subset of features as a prevalent co-location (i.e., an aggregation pattern), when it is truly a segregation pattern.

To overcome the limitations of the existing approaches, when using global prevalence thresholds, we propose to define the notion of a spatial interaction (co-location or segregation) based on a statistical test, develop appropriate null models for such tests, and propose computational methods to find statistically significant co-location and segregation patterns. Instead of a threshold based approach, our approach relies on a statistical test to decide whether an observed interaction is significant or is likely to have occurred by chance. To capture the spatial dependency among features in an interaction, we use a prevalence measure. Given a particular observed value of the prevalence measure for a possible spatial interaction (in the given data set), we then test the null hypothesis H_0 of no spatial dependency against an alternative hypothesis H_1 of spatial dependency (positive or negative) between the spatial features in the interaction. Using randomization tests, we estimate an empirical distribution of the prevalence measures under the null hypothesis. We reject the null hypothesis H_0 if the observed prevalence measure value is sufficiently large or sufficiently small. If the observed prevalence measure value is sufficiently large, a positive interaction likely exists among the participating features, giving rise to a co-location. If the observed prevalence measure value is sufficiently small, then a negative interaction likely exists among the participating features giving rise to a segregation pattern.

One of the main objectives of this thesis is to design a statistical framework to test the significance of an interaction behavior observed from a spatial domain. In this regard, the main challenge is designing an appropriate null hypothesis based on which the significance test can draw the right statistical inference for an observed interaction. Modeling the null hypothesis is an important part of any statistical significance test. Failing to appropriately model the null hypothesis of a significance test results in erroneous inference. Our objective is to test the significance of an observed spatial interaction against a null hypothesis assuming no spatial interaction between different features. Modeling such a null hypothesis for a spatial domain is not straightforward. Here the null model should take into account the spatial properties (distributions) of the features found in the observation but exclude spatial interactions among features. In many scenarios, finding an analytical mathematical expression to describe an observed spatial distribution of a feature may not even be possible due to the presence of spatial auto-correlation, feature abundance, and spatial heterogeneity. In such cases statisticians use a set of parameters to describe an observed spatial distribution and conduct simulations to estimate the values of these parameters for a given set of constraints [6]. Spatial features could show spatial auto-correlation behavior and/or uneven distribution of various intensities within the study area. All these issues make the design of a null model challenging when the spatial distribution of a feature is complex or unknown. In this thesis, we propose to design an appropriate null model which takes the spatial distribution of each feature into account and in doing so uses spatial models proposed in spatial statistics to characterize different spatial distributions.

Even after considering the spatial distributions of features in modeling our null hypothesis, the significance test may mistakenly report a random pattern R when subsets and/or supersets of the R happen to be true patterns. This happens as the null hypothesis does not take those true interactions into consideration. In this regard, we improve our null hypothesis which also takes true interactions into account. However, the challenge in designing such a null hypothesis lies in modeling a spatial interaction for which participating features are many and may have different spatial distributions. In spatial statistics, we find approaches that model interactions between two different features. However, to the best of our knowledge, no methods are constructed to simulate an interaction between more than two different features. We have proposed a heuristic in this regard which was found effective in our conducted experiments. Another challenge for the significance test of spatial interaction is designing a test statistic that can provide a numerical summary of an observed spatial interaction between features and can also be computed efficiently. We use one popular prevalence measure from the literature and propose two new test statistics that work well in our mining approach.

In spatial statistics, we can find models for the distribution of a test statistic (prevalence measure) as closed form analytical expressions for *pairs* of features. However a theoretical model to compute such a distribution for patterns of size larger than two have not been devised. With the increase of the pattern size, finding a theoretical model to compute the distribution is complex, especially since the individual features may have unknown distributions or may be auto-correlated. Hence in spatial analysis, an estimation of a cumulative distribution of the statistics is commonly obtained through a randomization test. During randomization tests, we generate data according to a null hypothesis. Finding a data generation model to simulate the null hypothesis is challenging as (1) each individual feature of the observed data can have more than one type of spatial distribution, such as: auto-correlation, regularity, randomness, and (2) simulating these distributions requires a mathematically sound and computationally efficient model. We propose a null model that can generate data sets where the spatial distribution properties of each individual feature of the observation are maintained. In addition, randomization tests pose some computational challenges. The first one is the computational cost that is incurred in simulating the large amount of data. The second one is the cost of computing the prevalence measure in the observed data and in all simulations. This computation requires first identifying instances of different patterns and then computing the prevalence measure for each pattern. Since "being statistically significant" is not an anti-monotone property, in a naïve approach, this has to be done for every possible pattern. We propose strategies which reduce the cost of the data generation step as well as the prevalence measure computation step. As a result, the statistical significance test becomes more efficient and computationally feasible compared to a naïve approach. We also show that our adopted strategies will correctly identify true patterns that existing method will miss.

Spatial interaction among features can occur not only at one single distance, but at different distances. Existing co-location mining algorithms look for prevalent co-locations only for a defined co-location neighborhood. Finding the right prevalence measure and the right distance threshold to find all true patterns without reporting any random pattern by the existing algorithms is not easy even sometimes not possible. As the distance threshold increases, more instances of a feature will get involved in co-locations. Hence the PI-value of any pattern increases with the increase of the distance threshold. This fact leads the chance of a random being reported as prevalent by a standard co-location algorithm using a substantially large distance threshold. A random pattern may also attain a high PI-value even at a smaller distance if the participating features are abundant or spatially auto-correlated, in which case a standard co-location algorithm will report a random pattern. Thus existing threshold based co-location approaches fails in detecting only true patterns at different distances. To test if a pattern is a true pattern at a distance d, we should test the statistical significance of its co-location property (measured by a prevalence measure) observed at d. At which distances a statistical test will be performed to determine the co-location distance of a true pattern $\mathcal{C}$? We propose to perform statistical tests only at those distances where unique instances of $\mathcal{C}$ are identified.

A pattern $\mathcal{C}$ may be reported significant at more than one distances. Then, the next question will be determining the co-location distance from one of those distances at which $\mathcal{C}$ is found significant. We propose to select the distance which will involve the highest number of features instances of $\mathcal{C}$ into co-location. Redundant patterns may also be generated from a statistical test that uses a null hypothesis based on an assumption of the independence of all features. Redundant patterns are not true patterns but could appear as significant due the presence of true patterns. With the increase of the multiple hypothesis tests performed at different distances, the chance of generating redundant pattern increases. We also propose an approach to prune redundant patterns and finally report a minimal set of patterns that can explain all the detected patterns. Our approach performs a statistical test using a *constrained null hypothesis* which assumes the independence of features for a given set of rules (co-locations). To the best of our knowledge, no model is proposed which can simulate such a constrained null hypothesis. We propose a heuristic to simulate a constrained null hypothesis and our approach can successfully identify and prune redundant patterns if exist in the results.

1.4 Thesis Contributions

The contributions of this thesis are the followings:

- Current algorithms for spatial co-location mining depend on user specified thresholds for prevalence measures; they do not take spatial auto-correlation into account, and may report co-locations even if the features are randomly distributed. We propose a method for finding co-located patterns that is based on a statistical test in order to avoid reporting co-locations generated by chance. We also have introduced a new type of spatial pattern called “segregation pattern” which occurs due to the presence of an inhibition relationship or negative association among a subset of features. To the best of our knowledge, only one related approach based on $maxPI$ value pruning has been proposed, and this method has similar limitations as other existing co-location mining algorithms, because it is also a threshold-based approach. We present a unified definition of spatial interaction patterns between features, for both co-location as well as segregation patterns.

- In a statistical significance test, a probability of seeing the observed value of a test statistic under the null hypothesis is computed. Modeling a null hypothesis under the assumption of independence of features is not straight forward, especially when features can have different spatial distributions. To this end, we develop an appropriate null model that takes the individual spatial distribution of a feature into account. The estimation of the null distribution of the test statistic is obtained through randomization tests, which is widely used in spatial statistics, since no closed form expression exists in the literature that models the joint distribution of more than two features. Analytical models that exist for *pairs* of features, will be used in some cases to validate our approach on pairs of features.

- We propose two approaches to mine interaction patterns. Our first approach is an *all instance based* approach where the test statistic (the prevalence measure) value is computed from all the instances of a pattern. In computing this test statistic, we introduce two strategies which reduce the total cost of computation. Due to the large number of simulations conducted in randomization tests, the statistical significance tests can become computationally expensive. We improve the runtime by introducing a pruning strategy to identify candidate patterns for which the prevalence measure computation is unnecessary. Taking spatial auto-correlation of features into account, we also show that in a simulation, we do not need to generate all instances of an auto-correlated feature and can reduce the runtime of the data generation phase in these cases.

  Our second approach is a *sampling approach* which improves runtime further. Here we propose a different test statistics, which can be viewed as an approximation of the PI-value, using only a subset of the total instances of a pattern. Complete neighborhood materialization to find all different sized interaction pattern instances is related to the problem of finding all maximal cliques, which is an NP-hard problem. Identifying the instances of different patterns incurs the major computational cost in a spatial interaction pattern mining algorithm. To design a more efficient method, we propose an approximation of the prevalence measure, computed from a subset of the total instances of a pattern, that can act effectively as the test statistic. We propose to compute the test statistic from instances located in a set of sub-regions sampled from the study area, so that a sampled sub-region forms a subset of the complete, circular neighborhood of a feature instance participating in an interaction. For an unbiased sampling approach, we place a grid over the study area. In a grid, a sampled sub-region is defined as a set of grid cells which partially cover and are completely inside the complete neighborhood of a feature instance. We study grids of different resolutions and show even very coarse grids allow us to draw mostly the same statistical inferences regarding statistically significant patterns as the full circular neighborhoods, while doing so at a substantially lower computational cost. Thus using such an approximation provides a good trade-off in accuracy versus runtime compared to the approach where all instances are identified.

- Furthermore, we show in this thesis that our statistical model can further be extended to mine all true *co-location patterns at multiple spatial distances*. The proposed mining algorithm does not require the interaction distance parameter from the users. To test the distance at which a true co-location pattern becomes significant, we introduce a different prevalence measure (the pattern instance count) as the test statistic. We also propose an approximation algorithm to find a minimal number of subsets that can “explain” all statistically significant co-location patterns and that represents all true positive interactions present in the data.

1.5 Thesis Outline

The organization of the thesis is as follows:

Chapter 2 starts with a brief discussion on the theory and popular measures of spatial statistics that are related to our research. Then, a comprehensive discussion on the current literature of co-location mining is provided.

Chapter 3 states the motivation of our research. It also formulates the objective of our research by introducing concepts and definitions.

In Chapter 4, we first formulate a baseline algorithm for mining interaction patterns. Then a new algorithm *SSCSP* (Statistically Significant Co-location and Segregation Pattern) is proposed which can improve the runtime of the baseline algorithm. To improve the runtime further, we propose a new prevalence measure. A *grid based sampling approach* is then introduced which allows us to compute this new prevalence measure more efficiently. A mathematical analysis is also provided to explain the behavior of this sampling approach. Finally, a complexity analysis of all of our approaches is provided. We conducts experiments to validate our model with a wide variety of synthetic and real data sets and compare our approaches with the state-of-the-art co-location mining algorithm.

In Chapter 5 we introduce a new prevalence measure which successfully works as a test statistic to test the significance of a true pattern. To find co-location patterns at different distances we propose a mining algorithm named *CPMNDC* (Co-location Pattern Mining with No Distance Constraint). This algorithm also determines the co-location distance of a co-location. To prune redundant patterns that could occur in a statistical test when the independence of all features is assumed for the null hypothesis, in this chapter we also propose an approximation algorithm. For redundancy checking, we propose a *constrained null model* for the statistical test and use a heuristic to simulate this null model. This algorithm finally reports a minimal set of patterns which can explain all the detected co-locations from a given data set. The conducted experiments show that our proposed co-location mining algorithm without a distance threshold can successfully mine all true patterns occurring at different distances from synthetic and real data sets.

Chapter 6 provides a discussion of the achievements and limitations of this thesis. We also provide some ideas that could improve some of the limitations of this thesis and further could lead to interesting problems in the area of spatial data mining.

The concepts and results of the proposed interaction pattern mining approach presented in Chapters 3 and 4 have been published in [9, 10].

Chapter 2

Background and Related Work

Work done in co-location or segregation pattern mining belongs to one of the following two areas - spatial statistics and spatial data mining. In this chapter, we highlight some work from these two domains. Researchers of spatial statistics have contributed by developing theories and methods for modeling spatial distributions and explorative analysis of spatial data. These methods work well for analyzing inter-point interaction of point features and modeling these interactions in specific cases. A detailed discussion is out of the scope of this thesis. However some basic concepts, terminology, and notations from spatial statistics are listed briefly in the next section. Our work adapts and uses these concepts for experimental validation; hence this discussion could lead to a better understanding for the reader. Subsequently a co-location mining algorithm using measures from spatial statistics is discussed.

The second section of this chapter discusses various co-location mining approaches proposed by the data mining community. There is a significant body of literature on co-location pattern mining that vary based on the types of transactionizing technique proposed to improve the runtime and computational efficiency. Inspired by the ARM algorithm, most of these techniques use one prevalence measure: PI and utilize its anti-monotonic property that reduces the search space for prevalent patterns. This discussion helps readers compare these methods and understand their limitations.

Segregation patterns have not received much attention in spatial data mining. However, in market basket analysis and the corresponding ARM literature, work is available that looks for occurrence of patterns due to a negative correlation. We discuss some of this work. Two spatial data mining methods are also discussed that mine patterns occurring due to a negative association. We also discuss two methods that try to mine patterns without requiring a distance threshold. Finally, we point out a major weakness in the existing approaches that is the major motivating factor of this research and state our goals achieved in this thesis.

2.1 Spatial Statistics

2.1.1 Basic Concepts

A spatial point process is a “stochastic mechanism which generates a countable set of events x_i in the plane” [20]. A point pattern is a realization of such a process that comprises a collection of events or objects occurring in a study region. An event set is made up of locations defined by some set of coordinates. Beside location information, additional information, mark, can also be attached with an event. Marks are often categorical such as class, sex, species, disease but can also be continuous, as in the case of temporal information.

The simplest way of treating a spatial point pattern is assuming that the pattern is random and is a realization of complete spatial randomness (CSR) which is a homogenous Poisson process with a constant intensity. CSR has two properties:

1. Equal probability: Any event has equal probability of being in a position of the study area.
2. Independence: The position of an event is independent of the positioning of any other event.

The second property implies that no interaction exists between events in the given point pattern [20]. In many real scenarios, the spatial data sets do not follow CSR due to interaction among events. This is due to certain events causing the occurrence of other events at nearby locations. The departure from CSR results in either (i) clustering (aggregation) or (ii) regularity (segregation) for an event set [20]. A positive interaction or positive dependency among events with different marks causes their instances to be found at nearby location. In the spatial data mining community, such type of patterns are termed as co-location patterns. A negative interaction or negative association on the other hand results in events of different types more regularly spaced from each other. Such patterns are called as segregation patterns. To identify positive or negative interactions, we test whether the observed point pattern (or a given event data set) is deviating from a CSR.

Estimation of the theoretical distribution of a spatial stochastic point process is difficult.

In most cases the Monte Carlo simulation is the only way to estimate the mean and the distribution of a test statistic. Selection of the test statistic depends on the type of mark. Classical techniques to determine properties of the distribution of single features and pairs of features (e.g., clustering tendency), use distance based methods such as pair-wise distance, nearest neighbor distance, and empty space distance to measure inter-point dependency [37]. The cumulative distribution function of the nearest neighbor distance $\big(G(d) = \frac{\text{no.}[d_{\text{min}}(x_i)<d]}{n}$, where x_i is an event in the point pattern X of n events and $d_{\text{min}}(x_i) = \mathbf{min}_{j\in(1,n)\ \&\ j\neq i}\mathit{dist}(x_i, x_j)$ where *dist* is the Euclidian distance$\big)$ [37] or the empty space distance $\big(F(d) = \frac{\text{no.}[d_{\text{min}}(u_i,X)<d]}{m}$ where u_i is a member of a randomly set of locations $\{u_1, \ldots, u_m\}$ and $d_{\text{min}}(u_i, X) = \mathbf{min}_{j\in(1,n)\ \&\ x_j\in X}\mathit{dist}(u_i, x_j)\big)$ [37]. J-function [37] which is the combination of F and G functions $\left(J(d) = \frac{1-G(d)}{1-F(d)}\right)$ is also a good choice. Second order analysis such as Ripley's K-function [51] or the *pair correlation function (PCF)* [37] are other alternatives and also popular techniques to detect clustering of events. K-function is defined by Brian D. Ripley for stationary point process. For event density λ (number per unit area), $\lambda K(d)$ is the expected number of other events within a distance d of a randomly chosen event of the process [51]. Formally, $K(d) = \frac{1}{\lambda}E[\text{number}(X \cap b(u, d)\backslash\{u\})|u \in X]$, u is a point of X and $b(u, d)$ is a disc of radius d centered on u. In a homogenous Poisson process, the expected number of points falling in $b(u, d)$ is $\lambda\pi d^2$, thus $K_{\text{pois}}(d)$ equals to πd^2. A K-function value of a point pattern greater than πd^2 suggests clustering, while a value less than πd^2 suggests regularity. *PCF* is another way of interpreting K-function and is formally defined as $g(d) = \frac{K'(d)}{2\pi d}$ [37] where $K'(d)$ is the derivative of K. These measures are designed for at most two types of events, i.e., bivariate point processes. In our work, we propose a method that can find co-location as well as segregation patterns for more than two types of events.

2.1.2 Related Work

Spatial statistics treats the co-location or segregation pattern mining problem in a little different manner than the data mining community. Here mining co-location or segregation patterns are similar to the problem of finding associations or interactions in multi-type spatial point processes. Association or interaction in a spatial point process is known as the second order effect. The second order effect is a result of the spatial dependence and represents the tendency of neighboring values to follow each other. There are several measures used to compute spatial interaction such as Ripley's K-function [51], distance based measures (e.g., F function, G function) [37], and co-variogram function [16]. These measures can summarize a point pattern and are able to detect clustering tendency (if it exists in the data) at different scales. With a large collection of Boolean spatial features, computation of the above measures becomes expensive as the number of candidate subsets increases exponentially in the number of different features.

K-function has the power to analyze the clustering tendency of points at different scales. The K-function for a bivariate spatial point pattern is defined as $K_{ij}(r) = \lambda_j^{-1}$(Expected number of type j events within distance r of a randomly chosen event of type i), where λ_j is the density (number per unit area) of event of type j. For two marked point processes $i = \{i_1, i_2, \ldots, i_n\}$ and $j = \{j_1, j_2, \ldots j_m\}$ observed over an area A, an estimate of the K-function without any edge correction is defined as [37]:

$$\widehat{K}_{ij}(r) = An^{-1}m^{-1}\sum_{x=1}^{n}\sum_{y=1}^{m} I_r(d_{i_x j_y}) \tag{2.1}$$

In equation (2.1), $I_r(d_{i_x j_y})$ is an indicator function which equals zero if the inter-distance between points i_x and j_y is greater than r, otherwise 1. With edge correction the above equation becomes $\widehat{K}_{ij}(r) = An^{-1}m^{-1}\sum_{x=1}^{n}\sum_{y=1}^{m} w_{i_x j_y} I_r(d_{i_x j_y})$ where $w_{i_x j_y}$ is the fraction of the circumference of a circle centered at i_x and radius r that falls inside the area A [37]. Under the assumption of complete spatial randomness (CSR), the expected value of $\widehat{K}_{ij}(r)$ is πr^2. If $\widehat{K}_{ij}(r)$ computed from the observation is less than πr^2, there is no clustering between i and j while if $\widehat{K}_{ij}(r) > \pi r^2$, these two point processes show clustering at distance r. As K-function is not linear in r, Besag's L-function [16] is used which is linear and has a constant variance. The L-function is defined as $L_{ij}(r) = \sqrt{\frac{K_{ij}(r)}{\pi}}$. Under negative association or repulsion, $(\widehat{L}_{ij}(r) - r)$ will give a negative value, whereas under positive association or clustering it will be a positive value.

Mane *et al.* in [41] use the above bivariate K-function as a statistical measure with a data mining tool to find the clusters of female chimpanzees' locations and investigate the dynamics of spatial interaction of a female chimpanzee with other male chimpanzees in the community. There, each chimpanzee is assigned with a unique mark based on its gender. Two clustering methods (SPACE-1 and SPACE-2) are proposed which use L-function to find clusters among different marked point processes. For a given a set of marks $M = \{m\}$, SPACE-1 first computes $\widehat{L}_{m_i,m_j}(r)$ for each pair of marks m_i and m_j. Subsequently, a hierarchical clustering for marks is obtained by using a dissimilarity matrix $M_{\widehat{L}(r)} = [l_{ij}], l_{ij} = \widehat{L}_{m_i m_j}(r)$ and applying the complete-link clustering algorithm.

Finally, a dendrogram of the hierarchical clustering results gives the visualization on the number of clusters found. However, to analyse such a dendrogram researchers still need to have domain knowledge to tell the correct number of clusters. Besides a minor variation in the dissimilarity values affects the dendrogram structure. As a result the dendrogram is not stable. To mitigate all these concerns, Mane *et al.* come up with an alternative where the Reverse Cuthill-McKee (RCM) ordering algorithm is used instead of complete-link algorithm to block diagonalize the matrix $M_{\widehat{L}(r)}$. This approach becomes now more stable and does not depend on an assumption of any hierarchical nature of the data set.

K-function is a popular measure to detect aggregation among events at different scale. However, a high value of K-function sometimes mislead by reporting a positive association in some scenarios of a bivariate point process. Consider a realization of a bivariate point pattern with event type i and j where only a few instances of type i are surrounded by all the instances of type j, leaving most of the instances of type i alone and without any instance of type j nearby. This could happen if event j is auto-correlated and instances of j appears in few clusters. Event i is randomly distributed and if a few instances i incidently fall in the clusters of j, we would possibly see a high value of K_{ij}-function. A value higher than πr^2 indicates a clustering tendency among i and j even though most instances of i's are not associated with any instance of j. Hence a high value of K-function can not always be an indication of a positive association of features. Another limitation of the K-function is that the function is defined only for univariate and bivariate point processes. The function is not defined for a point process where a higher order interaction (among more than two types events) is exhibiting.

2.2 Spatial Data Mining

While spatial statistics can find spatial association patterns of only size two, spatial data mining look for association patterns of any size. This section reviews some of the recent work done in this respect.

2.2.1 Spatial Association Rule Mining

In the data mining community, co-location pattern mining approaches are mainly based on spatial relationship such as "close to" proposed by Han and Koperski in [29, 38]. Koperski *et al.* present a method in [38] to mine frequently occurring patterns in geographic information systems. Such a pattern is presented in the form of a spatial association rule indicating a strong relationship among a set of spatial and some non-spatial predicates. A spatial association rule of the form $X \rightarrow Y$ states that in a spatial database if a set of feature X is present, another set of features Y is more likely to be present. In such rule at least one feature needs to be a spatial predicate in either X or Y. A support value (s) i.e. joint probability of this rule is computed which is the probability of seeing X and Y together in the database. A confidence value (c%) i.e. conditional probability is also computed which indicates that Y is found with X in $c\%$ of the total cases (transactions) of the database where X is found. Rules are built in an apriori-like fashion and a rule is defined as strong if it has enough support and confidence [2]. The anti-monotonic property of the support measure helps to reduce the total search space in finding prevalent rules comprising of spatial and non-spatial predicates.

2.2.2 Frequent Neighboring Class Set Mining

Morimoto in [45] proposes a method to find groups of various service types originating from nearby locations and reports a group if its frequency of occurrences is above a given threshold. Finding such groups can give important insight for location based service providers (cellular phones or PDAs) for attractive location-sensitive advertisements, portals, promotions etc. Groups of different sizes are searched by using an *Apriori-like* strategy. Here a group of k-different services occurring together more often, is defined as a k-neighboring class set. A k-neighboring class set is built by checking its subsets of $k-1$-size, each of which is also a neighboring class set. To build a k-neighboring class set from a $k-1$-neighboring class set, we look for a service type S which is different from the service types that are already in the $k-1$-neighboring class set. Besides S should have instances (points) that are frequently found close to the instances (group of points) of the $k-1$-neighboring class set. To find a nearest neighbor of a set of points in a plane, Morimoto uses Voronoi diagram [18]. To index these Vornoi points, a quaternary tree indexing is used which also keeps the run-time cost constant in finding a nearest service instance for an instance of a k-neighboring class set.

Limitation: Morimoto in his work [45] identifies instances of a co-location pattern by grouping neighboring feature instances with a constraint that one feature instance can not be included in more than one instance of a candidate co-location pattern. Hence another way of grouping feature instances for a candidate co-location gives different co-location

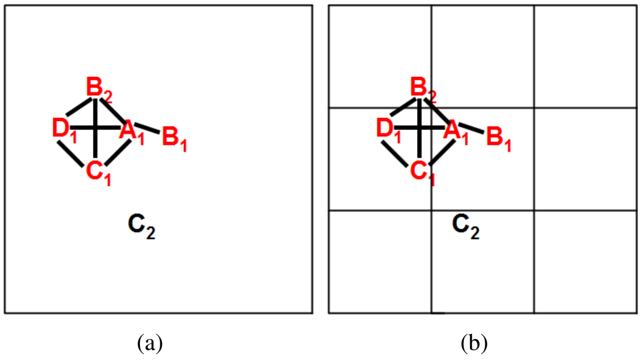


Figure 2.1: a) Grouping of points, co-located feature instances are shown in red. b) Grid based transactionization.

instances and poses the chance of missing instances of candidate co-locations. In Fig. 2.1(a), although A_1 is a neighbor of both B_1 and B_2, it will be grouped only with one to generate an instance of $\{A, B\}$. By grouping with B_1, A_1 generates instance $\{A_1, B_1\}$ of $\{A, B\}$. In this case, instance $\{A_1, B_2\}$ will not be generated and eventually instances of larger patterns such as $\{A_1, B_2, C_1\}$, $\{A_1, B_2, D_1\}$, $\{A_1, C_1, D_1\}$, and $\{A_1, B_2, C_1, D_1\}$ will not be generated.

2.2.3 Co-location Pattern Mining - a General Approach

To materialize transactions in a continuous spatial domain in order to use with an ARM approach, Shekhar *et al.* discuss three models (*reference feature centric model*, *window centric model*, and *event centric model*) [60] and define the co-location patterns for each model.

Feature centric model: Here each instance of a spatial feature generates a transaction [60]. Such a transaction includes other feature instances (relevant to the reference feature) appearing in the neighborhood of the instance that defines the transaction. In a feature centric model, once all transactions defined over the space are enumerated, one can tell how many instances of a spatial feature are in co-location with the instances of other features.

Window centric model: Here a given study area is discretized by placing a uniform grid and all cells (windows) of different sizes of $k \times k$ generate transactions. Instances of different feature types appearing in the same window give an instance of a co-location type.

Event centric model: Event centric model is the state-of-the-art that is followed in recent co-location mining algorithms. The event centric notion of co-location is defined based on a neighborhood relationship R. The relationship could be either spatial relationship (e.g. connected, adjacent), metric relationship (Euclidean distance), or a combination [35]. For instance, two spatial instances are in a neighborhood relationship R_d if they are located within a distance d of each other. In the *event centric model*, a group of feature instances forms a clique if each of them are neighbors and such a clique provides an instance of a particular co-location type. In Fig. 2.1(a) A_1, B_2, and C_1 are instances of spatial features A, B, and C respectively. As A_1, B_2, and C_1 are neighbors (their inter-distances are not more than the given distance threshold) of each other; $\{A_1, B_2, C_1\}$ gives an instance of a co-location type $\{A, B, C\}$. An instance of a candidate co-location is defined as a row instance. A table instance of a candidate co-location includes all of its row instances that are found in the given spatial data set. For the above mentioned example, $\{A_1, B_2, D_1\}$, $\{A_1, C_1, D_1\}$, $\{A_1, B_2, C_1, D_1\}$, are the row instances of $\{A, B, D\}$, $\{A, C, D\}$, and $\{A, B, C, D\}$ respectively. Let $\mathcal{C}$ be a candidate co-location of n different spatial features. We enumerate all the row instances of $\mathcal{C}$ to generate a table instance of $\mathcal{C}$. From this table instance, we know how many instances of a feature of $\mathcal{C}$ are found in various row instances of $\mathcal{C}$ and we then compute the frequency of a feature instance participating in $\mathcal{C}$ which is defined as the participation ratio. Each feature of $\mathcal{C}$ gives a participation ratio and the minimum one is selected as the *participation index (PI)* of $\mathcal{C}$. PI is used as the prevalence measure in [60]. A co-location mining algorithm is also proposed in [60] to find all prevalent co-locations for a user defined PI-threshold. PI has anti-monotonic property which helps to prune the search space while searching for all prevalent co-location patterns. In finding prevalent co-locations, the proposed algorithm uses a bottom-up approach which generates candidate co-locations of size $k + 1$ from the prevalent co-locations of size k. It adapts *Apriori_gen* algorithm of ARM. A candidate prevalent co-location of size $k + 1$ is generated by joining two prevalent co-locations of size k for which the first $k - 1$ features are the same but the k-th features are different. To check its prevalence, a table instance is generated by joining the row instances of its two prevalent subsets (co-locations) of size k. In such a join procedure, two row instances, one from each of the two prevalent subsets of size k, are joined if the first $k - 1$ feature instances are the same and the k-th feature instances are neighbors of each other. The resultant row instance of size $k + 1$ is included in the table instance of the candidate co-location of size $k+1$. After enumerating all the row instances of size $k+1$, the PI-value of the candidate co-location is computed and compared with the PI-threshold.

Limitation: In the proposed co-location mining algorithm of [60], $\mathcal{C}$ is declared as prevalent and finally reported, if its PI-value is not less than PI-threshold. The co-locations reported as prevalent depends on the selection of the threshold value. With a small threshold value, more patterns would be reported as prevalent where some of them would be meaningless and random patterns. The reported patterns should be evaluated from a statistical point of view. On the other hand, with a higher threshold value, less number of patterns would be reported; resulting in the possibility of missing meaningful patterns. Furthermore, the join approach of finding the row instances of a candidate co-location is computationally expensive.

2.2.4 Co-location Pattern Mining - a Multi-Resolution Based Approach

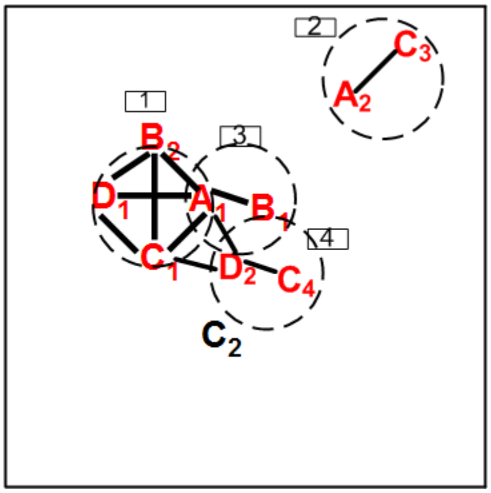
To improve the runtime of [60], Huang *et al.* propose a multi-resolution pruning technique in [35] which can detect non-prevalent co-locations and prune them at a reduced computational cost. Instead of computing the actual PI-value of a candidate co-location from the instance level as done in [60], this technique computes an upper bound of the PI-value at a coarser level. Such a PI-value computed at a coarser level, never underestimates the actual PI-value and is computationally less expensive. In this method, a grid is placed on the study area and feature instances appearing in a cell of the grid are all considered to be co-located. Each cell has at most 8 neighboring cells. To generate a $k + 1$-size co-location from a k-size co-location, the neighboring cells which have the feature instances are joined. The number of feature instances found in each neighboring cell that are participating in the join computation are counted and added up to give an upper bound of the actual participation ratio of a feature in a co-location. By taking the minimum participation ratio, an upper bound of the participation index of a co-location type is computed. A candidate co-location can be pruned if the upper bound of the PI is less than a given threshold. However, in this approach the actual PI-value of a candidate co-location is still computed and compared with the threshold when the upper bound of the PI is not less than the threshold value. This approach requires fewer number of join computations to find all prevalent patterns compared to the number join computations required in the earlier method [60]. [35] also shows that the PI is an upper bound of the cross K-function.

Limitation: Developing a transactionization technique which identifies each instance of a co-location pattern in one transaction without splitting it is a challenging problem. In [35], an explicit transactionization is done by placing a grid over the study area and each cell is considered to be a transaction. The cell size is decided based on the distance threshold. However this approach may still place a co-location instance across different cells (transaction). Fig. 2.1(b) shows an instance of the proposed transaction technique of [35] where instance $\{A_1, B_2, C_1\}$ is split into three different cells (i.e. transactions). To identify $\{A_1, B_2, C_1\}$, an instance level join proposed in [60] is still essential. On the other hand, instance $\{A_1, B_1\}$ placed in one cell can easily be identified by checking their presence in the cell. The computational advantage of [35] over [60] depends (1) on the cell size, i.e. distance threshold, and (2) the number of co-location instances found in one cell, i.e. the spatial distribution of the co-located feature instances. The worst situation is when the participating feature instances of all co-location instances get split across various cells. In such a situation, the approach of [35] computes both the upper bound and the actual value of PI for each candidate pattern. There the approach becomes computationally more expensive than the earlier approach in [60]. Morimoto's work runs the danger of missing an instance of a co-location. However, the work of Huang *et al.* never misses any instance of a candidate co-location. This approach still requires a proper PI-threshold value to report meaningful patterns.

To improve the runtime further, additional instance lookup schemes are also introduced. These schemes reduce the computational cost of pattern instance identification. Yoo *et al.* in [71, 72] propose two instance look-up schemes where a neighbor relationship is materialized in two different ways. In [71], a neighborhood is materialized from a clique type neighborhood and in [72], a neighborhood is materialized from a star type neighborhood.

2.2.5 Co-location Pattern Mining - a Partial-join Approach

In the partial-join approach proposed in [71], space is partitioned for generating neighborhood transactions based on the clique relationship of feature instances. To get a list of neighborhood transactions, in an ideal situation a set of maximal cliques are generated by minimizing the number of neighboring feature instances split on different partitions. In Fig. 2.2, there are four transactions (partitions). In such a partitioning approach, neighbor pairs $\{A_1, D_2\}$ and $\{C_1, D_2\}$ are split across two different transactions (partitions). Feature instances appearing in the same transaction are neighbors of each other. Hence instances of a co-location residing in the same transaction are easily identified without extra computation. However co-location instances that are spilt across different transactions are required to join. Fig. 2.2 shows that instance $\{A_1, C_1\}$ of the co-location type $\{A, C\}$ is identified in transaction 1, whereas instance $\{A_1, D_2\}$ of co-location type $\{A, D\}$ and instance $\{C_1, D_2\}$ of co-location type $\{C, D\}$ split across the transactions 1 and 4. To identify the instance $\{A_1, C_1, D_2\}$ of co-location type $\{A, C, D\}$, the computation of joining instance $\{A_1, D_2\}$ and instance $\{A_1, C_1\}$ is required. Additionally C_1 and D_2 are verified as being a neighboring pair. On the other hand, a co-location instance $\{A_1, B_2, C_1, D_1\}$ and any of its subsets can easily be identified without any join computation as the participating feature instances are located in one single transaction (1). This way the partial-join based framework can reduce the total number of join computation required in the earlier approach in [60]. After identifying all instances of a co-location pattern, the PI-value is computed and compared with a given threshold. Finally, a candidate co-location pattern is reported as prevalent if its participation index is equal or higher than a given threshold.



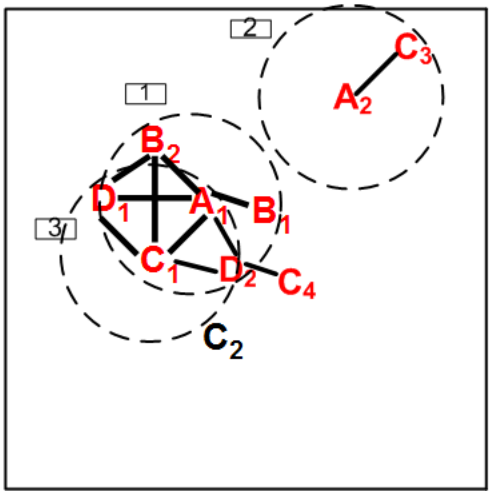
Clique No	Neighbor feature instances
1	A_1, B_2, C_1, D_1
2	A_2, C_3
3	$A_{1,}$ B_1
4	$C_{4,}$ D_2

Neighbor feature instances splitted over transactions
$A_{1,}$ D_2
$C_{1,}$ D_2

Figure 2.2: A clique neighborhood based partitioning approach.

2.2.6 Co-location Pattern Mining - a Join-less Approach

The major computational bottleneck of [35, 60] is the join computation required to identify instances of a candidate co-location. In a join-less approach [72], the cost for identifying candidate co-location instances is minimized. Like the earlier approaches, this method also introduces a strategy to identify and prune non-prevalent co-locations from a coarser level without even identifying their instances. The objective here is to find an efficient neighborhood materialization that helps to identify all maximal cliques at a smaller computational cost than the approach of Yoo *et al.* in [71]. In the join-less approach, a disjoint star type neighbor relationship is materialized. In a star neighborhood, a feature instance acts as a center object and other feature instances which are in a neighbor relationship with the center object are also included in the star neighborhood. Fig. 2.3 gives a star neighborhood ex-



Center object		Neighbor feature
feature	instance	instances
A	A_1	A_1 B_1 B_2 C_1 D_1 D_2
	A_2	A_2 C_3
B	B_1	B_1
	B_2	B_2 C_1 D_1
C	C_1	C_1 D_1 D_2
	C_2	C_2
	C_3	C_3
	C_4	C_4 D_2
D	D_1	D_1
	D_2	D_2

Figure 2.3: A star neighborhood based partitioning approach.

ample where feature instance A_1 forms a star neighborhood by including feature instances B_1, B_2, C_1, D_1, and D_2. Here features are ordered either arbitrarily or using domain related information. A feature instance in a neighbor relationship with the center object can not be included in the star neighborhood of the center object if it is equal or lower than the center object. Using geometric method such as plane sweep or a spatial query method, the first neighboring object pairs are identified and star neighborhoods are constructed by grouping object pairs. For a candidate co-location, its co-location instances are first filtered out from the star neighborhoods. For example, the instances of a candidate co-location $\{A, B, C, D\}$, are filtered from the star neighborhoods with A as the center object. As a star neighborhood is not a clique, the PI of a candidate co-location computed from the star neighborhoods is a coarser value of the actual PI and can not be less than the actual PI. If the coarse PI-value of a candidate co-location is lower than a given threshold, the candidate co-location can not longer be a prevalent pattern; and hence can be pruned. This way a non-prevalent co-location can be identified even without identifying its actual instances. When the coarse PI-value of a candidate $\mathcal{C}$ is equal or higher than a threshold, the actual PI-value needs to be computed. For this we need to identify all instances of $\mathcal{C}$ which can be done efficiently by checking the cliqueness among instances of features found in the star neighborhoods. Instances of a $\mathcal{C}$ can be identified only if the subsets of $\mathcal{C}$ are already identified as prevalent. The experimental evaluation shows that the total number of instances that are checked in finding all prevalent patterns from a star type neighborhood is far less than the number of instances checked in [60].

2.2.7 Co-location Pattern Mining - a Density Based Approach

Xiao *et al.* in [69] improve the runtime of *Apriori_gen* based methods [35, 45, 71] computationally. Instead of identifying all instances of a candidate co-location as done in other *Apriori_gen* based approaches, Xiao's approach does not need all the instances of a candidate to compute its prevalence measure and check the prevalence. The proposed approach is named *"density based co-location mining"* method as the search for co-location patterns first starts from the most dense region of features and progressively proceeds to less dense regions. While checking a dense region, the method identifies the number of instances of a feature that participates in a candidate co-location. Assuming that the remaining instances are in co-location, the method estimates an upper bound of the PI for the candidate co-location and compares it against a given threshold. If the upper bound for a candidate co-location is less than a given threshold, it can be pruned even without identifying its instances for the remaining areas (less dense regions). This way the join computation required for identifying instances of a non-prevalent co-location in the less dense regions can be avoided.

2.2.8 Negative Association Rule Mining

Besides finding patterns occurring due to a positive association among spatial features, researchers often look for patterns that can also occur due to the effect of an inhibition or a negative association. In association rule mining, methods have been proposed to mine patterns occurring due to correlation among items of the market basket data. In this regard, the work of [13] is worth mentioning. The approach proposed in [13] finds rules describing the correlations among market basket items. In generating such rules, the presence as well as the absence of an item in a transaction are taken into consideration. The significance of the generated rules is measured by using a classical test, the χ^2-test used as a measure of correlation. The authors also show that this measure is upward closed which leads to efficiently finding a border between the correlated and uncorrelated item sets. The computed χ^2-value of item sets X and Y is used to determine if X and Y are correlated or not. If two item sets are found correlated, one obvious question is knowing the type of correlation. The approach of Brin *et al.* does not answer this and thus can not distinguish between positive and negative rules that occur due to the positive and negative correlations respectively.

To generate both positive and negative association rules Wu *et al.* in [68] propose a new algorithm which uses *mininterest* on top of the support-confidence framework. Here a rule $X \to Y$ is defined as interesting only if $support(A \cup B) - support(A)support(B) > mininterest$. An itemset whether positive or negative (such as $\{A, B\}$, or $\{A, -B\}$ respectively) will be reported only if the support value and the interest value respectively exceed the minimum values of support and interest measures. However this approach does not discuss how to set up these minimum values and how the results could vary with different minimum values. Antonie *et al.* propose an algorithm in [4] to identify market basket items that complement each other or are in conflict with each other. Their approach extends the support-confidence framework of ARM with a sliding correlation coefficient threshold to find both positive and association rules in a reduced search space. Other work such as [56, 62] also address mining negative rules.

2.2.9 Negative Pattern Mining from Spatial Databases

To the best of our knowledge there is little work in the spatial domain that looks for patterns occurring owing to negative interactions. Munro *et al.* in [46] first discuss more complex spatial relationships and spatial patterns that occur due to such a relationship. A combination of positive and negative correlation behavior among a group of features gives rise to a complex type of co-location pattern. Arunasalam *et al.* in [5] develop a method to mine positive, negative, and complex (mixed) co-location patterns. For mining such patterns, their method uses a user specified threshold on prevalence measure called *maximum participation index* ($maxPI$) which was first introduced in [34] to detect a co-location pattern where at least one feature has a high participation ratio. $maxPI$ of a co-location C is the maximal participation ratio of all the features of C. In using PI, sometimes there is a chance of missing a rule that has a low support value but has a high value of confidence. Let $\{A, B\}$ be a co-location type where most of the instances of A are co-located with instances of B ($pr(\{A, B\}, A) >$ threshold). On the other hand, for feature B, most of its instances are not co-located with the instances of A ($pr\{A, B\}, B) <$ threshold). Hence a method using PI will not report $\{A, B\}$ as the minimum PI which is the $pr(A)$ is less than the given PI-threshold. However such type of patterns can also be interesting in a scenario where a feature has a high participation in a co-location. This is termed as a rare event by Huang *et al.* in [34]. To mine such type of patterns, $maxPI$ can be used which prevents $\{A, B\}$ from being pruned.

A complex type of pattern is the one where features are seen co-located in the absence of other features (e.g. $\{A, -B, C\}$ or $\{A, B, -C\}$). In complex pattern mining problem the total number of candidate patterns of size 1 is doubled for a given number of features. For instance, for a given set of features, $\{A, B, C, D\}$, the candidate 1-item set will be $\{A, B, C, D, -A, -B, -C, -D\}$ as we are now considering both the presence and the absence of a feature in constructing all candidate patterns. The exponential growth of candidate space with an increased number of features makes the pattern mining computationally expensive. Arunasalam *et al.* show that by using $maxPI$, a large number of negative patterns constructed from a group of features can be pruned if the $maxPI$-value of the positive pattern constructed from the same group is greater than the threshold [5]. Let $\mathcal{C}$ be a positive pattern of size k whose $maxPI$-value is greater than a user specified threshold value 0.5 and the $maxPI$-value is equal to the participation ratio of a feature f_i in $\mathcal{C}$. Any negative pattern where one feature from $\mathcal{C} - f_i$ is negatively associated with the other participating features of $\mathcal{C}$ can not have a $maxPI$-value greater than 0.5 and thus can be pruned. Let $\mathcal{C} = \{A, B, C, D\}$ have a $maxPI$-value greater than the threshold ($t = 0.5$) and $maxPI(\{A, B, C, D\}) = pr(\{A, B, C, D\}, A) \geq t = 0.5$, then pattern $\{A, -B, C, D\}$, pattern $\{A, B, -C, D\}$, and pattern $\{A, B, C, -D\}$) can not have a $maxPI$-value greater than t. Using the above rule, the total number of candidate checking is reduced which eventually reduces the total computational cost of mining complex patterns.

Limitation: The proposed pruning technique works only when a threshold value of 0.5 or greater is selected. In their method, selection of the right threshold is important for capturing a pattern occurring due to a true correlation behavior. This method lacks validation of the significance of a pattern statistically when the pattern size is greater than two. Spatial auto-correlation behavior is also not considered in this approach.

2.2.10 Co-location Region and Uncertain Co-location Pattern Mining

Mining co-location regions: Co-location tendency among a set of features could be different in regions. Finding regions where a co-location becomes weaker or stronger than expected in CSR is a fresh problem in co-location mining research. Wang *et al.* in [65] propose a *Bayesian* method to find such type of co-location regions. This method defines a region using cells of a rectangular shaped grid. However, this method can be expensive when a better approximation of a found region which could be of irregular shape is intended.

Mining uncertain co-locations: Location information of spatial features in some application domains such as transportation system is imprecise. In certain cases over-counting is the major computational bottleneck owing to the continuity of space. The over-counting problem progressively worse with the uncertainty in the location of a feature. To find co-locations in an uncertain model, Liu *et al.* [39] propose a probabilistic PI as a prevalence measure. Using this measure, an Uncertain Apriori co-location mining algorithm is proposed. An event level pruning is adapted and an Uncertain Feature Tree based algorithm is also proposed for efficient mining. The proposed approach is validated using a Shanghai taxi trajectory data set.

Adilmagambetov *et al.* propose a method in [1] which finds co-location patterns in data sets with extended spatial objects.

2.2.11 Co-location Pattern Mining in Dynamic neighborhoods

All the above mentioned approaches assume a static neighborhood (fixed distance threshold) and a user defined prevalence measure threshold for finding prevalent patterns. The constraint on a fixed prevalence measure threshold apparently introduces many drawbacks which will be discussed in this thesis. The size of the co-location neighborhood for all co-locations can not be the same due to the fact that interactions among different groups of spatial feature happen at different spatial resolutions. Finding prevalent co-locations patterns without knowing their interaction distance threshold is a relatively new problem. Yo *et al.* in [70] propose a framework to mine meaningful co-locations without user defined distance parameter. Using bi-variate K-function this approach looks for a distance as a co-location distance where the function value is high. As the K-function works only for patterns of size 2, their proposed technique may fail to choose the right distance when a pattern size is greater than 3.

In another approach, Wang *et al.* in [50] investigate limitations of the existing approaches for the preassumption of a static neighborhood in the mining process. They define the mining problem as an optimization task and propose a greedy algorithm to mine co-locations with dynamic neighborhood constraints. However, their proposed method lacks statistical validation.

2.3 Summary

After reviewing the current literature, we find that most of the proposed methods dedicate their efforts towards reducing the computational cost rather than achieving accuracy in mining meaningful patterns. The statistical validation of the mined patterns of size higher than two is not done in any of the proposed methods. The accuracy of these methods highly depends on the proper selection of the prevalence measure threshold and the distance threshold defining the interaction neighborhood. However in certain cases a pair of values for these two thresholds which leads to mining only meaningful patterns is not even possible to find. In this thesis, we try to resolve these issues and propose methods to mine only true patterns without requiring any of these two thresholds. Existing work uses different prevalence measure in mining patterns occurring due to the positive and negative association. However our model uses one prevalence measure to mine both types of patterns. In our mining process, spatial auto-correlation behavior which is common in spatial domains is also considered.

Chapter 3

Statistically Significant Interaction Patterns: Motivation and Basic Concepts

In spatial data sets, it is not uncommon to see subsets of features with a high prevalence measure value due to randomness, presence of spatial autocorrelation, and abundance of feature instances alone, i.e., without true interaction between the involved features. Existing co-location mining algorithms will report such subsets as prevalent patterns if their prevalence measure values are higher than the chosen threshold. In another scenario, the prevalence measure value of a group of features can be low, if one participating feature has a high participation ratio, but other participating features have low participating ratios due to their large number of feature instances. Overall the minimum participation value of such pattern will be low, hence will be ignored by the existing co-location mining approaches. However in epidemiology this type of pattern are more common and is interesting to mine. To capture such patterns using the general co-location mining algorithms, the prevalence threshold should be set very low. Setting such a low value as a global threshold, however, results in reporting potentially a large number of meaningless patterns that also have a prevalence measure value higher than the low threshold. On the other hand, a low prevalence measure value does not necessarily mean a true segregation as features with few instances when randomly distributed could also generate a low prevalence measure value.

3.1 Motivating Examples

Consider the following (sketches of) example scenarios (including a real data set), in order to see the need for a different type of approach that takes the distribution and the total number of feature instances into account. Each of these examples is illustrated with a data set and figure.

Example 1: This case is illustrated in Fig. 3.1 with 4 instances of $\circ$ and 20 instances of $\triangle$. Every instance of $\circ$ is co-located with an instance of $\triangle$, whereas most of the instances of $\triangle$ are not co-located with an $\circ$. The participation index of $\{\circ, \triangle\}$ is 0.2. Existing algorithms using a PI threshold larger than 0.2 will miss the pattern $\{\circ, \triangle\}$.

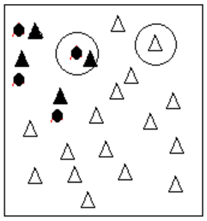


Figure 3.1: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{4}{4}, \frac{4}{20})$.

Example 2: This case is illustrated in Fig. 3.2. Here each of $\circ$ and $\triangle$ has 12 instances which are randomly distributed. Due to randomness, here half of the total instances of $\circ$ are found in co-locations with 5 instances of $\triangle$, resulting in a PI-value of 0.42 which is higher than a typical PI-threshold used in practice. Hence, $\{\circ, \triangle\}$ might be reported although no true spatial dependency exists between $\circ$ and $\triangle$.

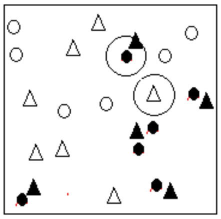


Figure 3.2: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{6}{12}, \frac{5}{12})$.

Example 3: This case is illustrated in Fig. 3.3 where 9 instances of $\circ$ appear in 3 clusters and 7 instances of $\triangle$ appear in 3 clusters. One cluster of $\circ$ and one cluster of $\triangle$ happen to overlap by chance and this results 4 instances of $\circ$ to be in co-location with 3 instances of $\triangle$. The PI-value is 0.43 which is higher than a typical PI-threshold and $\{\circ, \triangle\}$ may be reported by the existing algorithms.

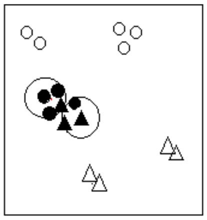


Figure 3.3: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{4}{9}, \frac{3}{7})$.

Example 4: This is illustrated in Fig. 3.4 where 9 instances of feature $\circ$ appear in 4 clusters and 6 instances of feature $\triangle$ are randomly distributed. 3 instances of feature $\triangle$ fall in 3 clusters of feature $\circ$ and thus are co-located with $\circ$s. 7 $\circ$s in those 3 clusters are also co-located with $\triangle$s. The PI-value is 0.5 which is higher than a typical PI-threshold and $\{\circ, \triangle\}$ may be reported by the existing algorithms.

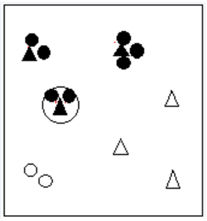


Figure 3.4: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{7}{9}, \frac{3}{6})$.

Example 5: The scenario in Fig. 3.5 represents a realization of an inhibition process generated using a multi-type Strauss process [37]. Here feature $\circ$ and feature $\triangle$ exhibits a spatial inhibition at a distance $R_d = 0.1$, and the study ares is a unit square. Each feature has 40 instances. 25 instances of $\circ$ and 22 instances of $\triangle$ are found in co-location. Hence, the participation ratio of $\circ$ and $\triangle$ are 0.625 and 0.55. Finally, the PI-value is 0.55. Existing co-location mining algorithm will report $\{\circ, \triangle\}$ as a prevalent co-location pattern if a threshold value of 0.55 or less is used. If $\circ$ and $\triangle$ were distributed independently of each other, the expected PI-value is found as 0.71 and most of the time the PI-value under independence assumption is higher than the observed PI-value 0.55. Hence $\{\circ, \triangle\}$ can not be reported as a co-location pattern rather should be reported as a segregation pattern.

Example 6: In Fig. 3.6, feature $\circ$ and feature $\triangle$ are distributed independently of each other. We find only one instance of $\{\circ, \triangle\}$, thus giving a low PI-value (0.2). As the number of instances of each feature is low, seeing a low PI-value is not uncommon even when features are independent of each other. Hence a pattern with a low PI-value does not always mean a segregation pattern.

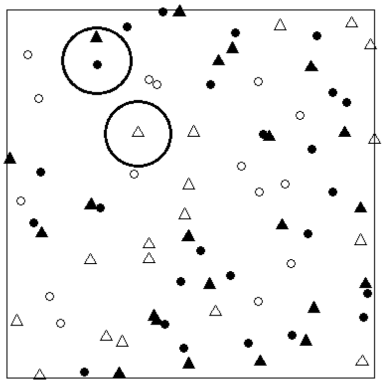


Figure 3.5: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{25}{40}, \frac{22}{40})$.

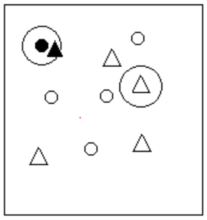
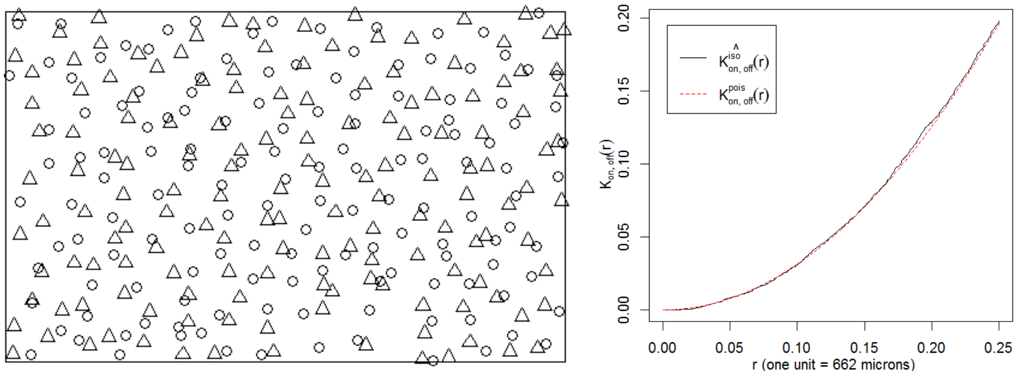


Figure 3.6: Filled $\circ$ and $\triangle$ are co-located. PI = min$(\frac{1}{5}, \frac{1}{5})$.

Example 7 - a real data set: The spatial dependency between the positions of two types of retinal neurons, known as the cholinergic amacrine cells are investigated to understand if these two types of cells emerge independently or from a single undifferentiated population during development. These two types of cells help in detecting motion in a particular direction. Cells found within the inner nuclear layer are termed as "off" cells and cells found in the ganglion cell layer are termed as "on" cells. Wieniawa-Narkiewicz [66] recorded a data set (Fig. 3.7(a)) of 152 "on" (with mark $\circ$) and 142 "off" (with mark $\triangle$) cells from a rectangular section of retina with a dimension of 1060 by 660 μm. We find that at interaction distance 32μm, the PI-value is 0.5, and increases up to 1 when increasing this interaction distance. For instance at distance 46μm, the PI-value is 0.89. This is a high PI-value and existing algorithms will report {on, off} as a prevalent co-location pattern. Diggle [19] showed an independence among these two types of cells. Fig. 3.7(b) shows that the cross K-function [37] curve (estimated from the amacrine data) closely follows the theoretical curve indicating no aggregation tendency between these two types of cells.



(a) PI = 0.89 at 46μm

(b)

Figure 3.7: (a) Amacrine data [7] (b) Cross K-function.

3.2 Basic Idea for Finding Statistically Justified Co-location and Segregation Patterns

We suggest that, instead of using a global threshold, we should estimate, for the given number of As and Bs, how much larger the observed PI-value is compared to a PI-value when A and B have no spatial relationship. If the observed PI-value is significantly larger than a PI-value under no spatial relationship, we conclude that A and B are spatially co-located, and $\{A, B\}$ should be reported as a prevalent co-location pattern. On the other hand, if the observed PI-value is significantly lower than a PI-value under no spatial relationship, we conclude that A and B are spatially inhibitive, and $\{A, B\}$ should be reported as a prevalent segregation pattern. In this manner, the decision of co-location or segregation pattern detection does not depend on a user-defined prevalence measure threshold.

Note that such an approach works with any type of prevalence measures to capture spatial interaction among features and is not dependent on the PI measure. A measure of spatial dependency among features tries to capture the strength of an interaction; the PI is one such measure that we will adopt in our method.

The main idea of our approach is to estimate the probability of the prevalence measure such as the PI-value of a pattern observed in the given data set, under some null hypothesis of spatial independence. In other words, we have to answer the question: what is the chance of obtaining a PI-value at least as extreme as the observed PI-value if the features were spatially independent of each other? The answer to this question gives us a p-value. If the p-value is low, the observed PI-value is a rare phenomenon under the null model, thus indicating a co-location or segregation among the features. The observed PI-value is said to be statistically significant at level α, if $p \leq \alpha$.

If the observed PI-value of a pattern is significantly higher than its PI-value under no spatial relationship, we call the pattern a co-location pattern; if the observed PI-value of a pattern is significantly lower than its PI-value under no spatial relationship, we call the pattern a segregation pattern.

For such an approach to work properly, the distribution of the PI values under the null hypothesis has to be adequately modeled. The current literature does not consider the spatial *auto-correlation* in mining interaction patterns, and we have seen in the motivating examples that spatial auto-correlation can lead to falsely reported patterns. In our null model design, we will take spatial auto-correlation into account, and we show in later sections how the existing PI measure behaves in the presence of spatial auto-correlation.

3.3 Null Model Design

Our null hypothesis must model the assumption that different features are distributed in the space independently of each other. A spatial feature could be either spatially auto-correlated or not spatially auto-correlated. A feature which is spatially auto-correlated in the given data is modeled as a cluster process [37]. To determine if a feature is spatially auto-correlated or not, we compute the value of *pairwise correlation function (PCF)* which is denoted by $g(d)$ (see Section 2.1.1). Values of $g(d) > 1$ suggest clustering or attraction at distance d. A feature has a regular distribution (inhibition) if $g(d) < 1$, and a feature shows CSR if $g(d) = 1$. Hence for $g(d) \leq 1$, the feature is considered to be not spatially auto-correlated. Although d could be of any value but we assume that the value of d equals of at least the co-location neighborhood radius.

To model an aggregated point pattern, Neyman and Scott [47] introduce the Poisson cluster process using the following three postulates:

1. First, parent events are generated from a Poisson process. The *intensity* of the Poisson process could be either a constant (homogenous Poisson process) or a function (inhomogenous Poisson process).

2. Each parent gives rise to a finite set of offspring events according to some probability distribution.

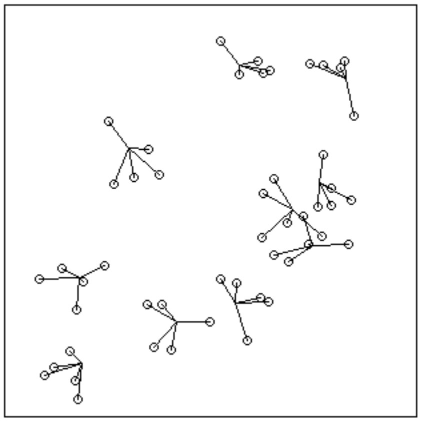


Figure 3.8: A cluster process: offsprings of a parent event are shown by connecting edges.

3. The offspring events are independently and identically distributed in a predefined neighborhood of their parent event.

The offspring sets represent the final cluster process. An example of such a cluster process is shown in Fig. 3.8. In such a model, auto-correlation can be measured in terms of intensity of the parent process and the intensity of the offspring process. In a Matérn's cluster process [37], another model, cluster centers are also generated from a Poisson process with intensity κ. Then each cluster center c is replaced by a random number of offspring points, where the number of points is generated from a Poisson process with intensity μ; the point themselves are uniformly and independently distributed inside a disc of radius r centered at c. Another model for aggregated point patterns is Thomas process [37]. Similar to a Neyman-Scott process, here cluster centers are generated from a Poisson process with intensity κ. But the spatial distribution of the offsprings of each cluster follows an isotropic Gaussian $N(0, \sigma^2 I)$ displacement from the cluster center c. The number of offsprings in each cluster is drawn from a Poisson distribution with mean μ. Another alternative model is the log Gaussian Cox process [37] which can also be used to model a spatially auto-correlated data.

A spatial distribution of a feature can be described in terms of summary statistics, i.e. a set of parameters. If a feature is detected to be spatially auto-correlated, the parameters can be estimated using a model fitting technique. The method of Minimum Contrast [21] fits a point process model to a given point data set. This technique first computes a summary statistics from the point data. A theoretically expected value of the model to fit is either derived or estimated from simulation. Then the model is fitted to the given data set by finding optimal parameter values of the model to give the closest match between the theoretical curve and the empirical curve. For the Matérn Cluster process [37], the summary statistics are κ, μ, and r. For the Thomas cluster process, the summary statistics are κ, μ, and σ.

The data sets we will simulate according to our null hypothesis maintain the following properties:

1. same number of instances for each feature as in the observed data, and
2. similar spatial distribution of each feature by maintaining same distributional properties (summary statistics) estimated from the observed data.

For instance, if a feature is spatially auto-correlated in the given data set, the feature will also be clustered in the same degree and the clusters will be randomly distributed over the study area in the generated data sets under the null hypothesis. For an auto-correlated feature, we estimate the parameters of a Matérn Cluster process which is used to model the auto-correlation in our experiments. If a feature is randomly distributed, we estimate its Poisson intensity by fitting a Poisson point process to the given data. This intensity could be either homogenous (a constant value) or non-homogenous (a function of x and y).

To generate a data set based on our null hypothesis, our approach first generates instances of each feature using a spatial distribution (either Poisson or auto-correlation) and then superimposes the generated instances of features in the study area. The cost here is first estimating the summary statistics (distributional properties) of each individual feature and then generating instances of a feature from a distribution function defined by those estimated summary statistics. Data permutation technique [22] is another alternative that is widely used for random data generation. To generate a spatial random data this approach does an rearrangement or shuffling of the labels (feature type or marks) of the observed data points (feature instances) but preserves the location information (spatial identity) of the data points of the observation. This method is based on an assumption that the labels are exchangeable so that rearrangements of the labels are equally likely [27]. This implies that an observed location is equally probable for being the location of any feature. Such assumption of exchangeability can be violated in the case where features (marks) show spatial auto-correlation instead of showing homogenous Poisson distribution. If an observed location l is labeled for an auto-correlated feature f, the neighboring locations of l are more likely to be labeled as f. If we do a random labeling for the neighboring locations of l, the generated data may fail to preserve the same degree of auto-correlation property of f as seen in the observation. To avoid such a situation, a restricted data permutation can be performed where the spatial dependence (that may exist among instances of a feature) is taken into consideration while labeling the observed locations of the data points. Several procedures in this regard are proposed. Among them the techniques of *regional partitions*, *toroidal shift*, and *randomization by maintaining spatial auto-correlation* are used in ecological domains [22]. Compared to our data generation approach, a general data permutation method is computationally less expensive since it does not need to estimate the summary statistics and simulate the spatial distributions of features. However, it may not be a good choice when auto-correlated features are present in the given data. In such cases, a restricted data permutation approach can be adopted which also requires knowing the distributional properties of each feature before labeling data points. This requires more computations than a simple data permutation approach and may not be much cheaper than our used data generation approach.

3.4 Definitions

We first define an interaction pattern and then state two definitions from the literature [35, 60] since we use the PI as a spatial interaction measure:

Definition 1. *An interaction pattern is a subset of k different features $f_1, f_2, \ldots, f_k$ having a spatial interaction within a given distance R_d. R_d is called as the interaction distance. A group of features are said to have a spatial interaction if features of each possible pairs are neighbors of each other. Two feature instances are neighbors of each other if their Euclidian distance is not more than the interaction distance R_d. Let $\mathcal{C} = \{f_1, f_2, \ldots, f_k\}$ be an interaction pattern. In an instance of $\mathcal{C}$, one instance from each of the k features will be present and all these feature instances are neighbors of each other.*

Definition 2. *The **Participation Ratio** of feature f_i in $\mathcal{C}$, $pr(\mathcal{C}, f_i)$, is the fraction of instances of f_i participating in any instance of $\mathcal{C}$ [35, 60]. Formally,*

$$pr(\mathcal{C}, f_i) = \frac{|(\pi_{f_i}(\textit{all instances of } \mathcal{C}))|}{|\textit{instances of } f_i|}.$$

Here π is the relational projection with duplication elimination.

For instance, let an interaction pattern $\mathcal{C} = \{P, Q, R\}$ and P, Q, and R have n_P, n_Q, and n_R instances respectively. If $n_P^{\mathcal{C}}$, $n_Q^{\mathcal{C}}$, and $n_R^{\mathcal{C}}$ distinct instances of P, Q, and R, respectively, participate in pattern $\mathcal{C}$, the participation ratio of P, Q, R are $\frac{n_P^{\mathcal{C}}}{n_P}$, $\frac{n_Q^{\mathcal{C}}}{n_Q}$, $\frac{n_R^{\mathcal{C}}}{n_R}$ respectively.

Definition 3. *The **Participation Index** (PI) of an interaction pattern $\mathcal{C}$ is defined as $PI(\mathcal{C})$ $= min_k\{pr(\mathcal{C}, f_k)\}$ [35].*

For example, let an interaction pattern $\mathcal{C} = \{P, Q, R\}$ where the participation ratios of P, Q, and R are $\frac{2}{4}$, $\frac{2}{7}$, and $\frac{1}{8}$ respectively. The PI-value of $\mathcal{C}$ is $\frac{1}{8}$.

Lemma 1. *The participation ratio and the participation index are monotonically non-increasing with the increase of pattern size, that is if $\mathcal{C}' \subset \mathcal{C}$ and $f \in \mathcal{C}'$ then $pr(\mathcal{C}', f) \geq pr(\mathcal{C}, f)$ and $PI(\mathcal{C}') \geq PI(\mathcal{C})$ [35, 60].*

3.5 Statistical Significance Test

Let $PI_{\text{obs}}(\mathcal{C})$ denote the participation index of $\mathcal{C}$ in the observed data, and let $PI_0(\mathcal{C})$ denote the participation index of $\mathcal{C}$ in a data set generated under our null hypothesis. Then we estimate, using the distribution of PI-values under the null model, two probabilities: $p_{\text{pos}} = Pr(PI_0(\mathcal{C}) \geq PI_{\text{obs}}(\mathcal{C}))$, the probability that $PI_0(\mathcal{C})$ is at least $PI_{\text{obs}}(\mathcal{C})$, and $p_{\text{neg}} = Pr(PI_0(\mathcal{C}) \leq PI_{\text{obs}}(\mathcal{C}))$, the probability that $PI_0(\mathcal{C})$ is at most $PI_{\text{obs}}(\mathcal{C})$. If $p_{\text{pos}} \leq \alpha$, or $p_{\text{neg}} \leq \alpha$, the null hypothesis is rejected and the $PI_{\text{obs}}(\mathcal{C})$-value is significant at level α.

α is the probability of committing a type I error, which is rejecting a null hypothesis when the null hypothesis is true, i.e. the probability of accepting a spurious co-location or a segregation pattern. If a typical value of $\alpha = 0.05$ is used, there is 5% chance that a spurious co-location or a segregation is reported.

To compute p_{pos} and p_{neg}, we do randomization tests, generating a large number of simulated data sets that conform to the null hypothesis. Then we compute the PI-value of a pattern $\mathcal{C}$, $PI_0(\mathcal{C})$, in each simulation run and compute p_{pos} and p_{neg} respectively as:

$$p_{\text{pos}} = \frac{R^{\geq PI_{\text{obs}}} + 1}{R + 1} \quad (3.1) \qquad p_{\text{neg}} = \frac{R^{\leq PI_{\text{obs}}} + 1}{R + 1} \quad (3.2)$$

Here $R^{\geq PI_{\text{obs}}}$ of equation (3.1) represents the number of simulations where the computed $PI_0(C)$ is not smaller than the PI_{obs}-value. $R^{\leq PI_{\text{obs}}}$ of equation (3.2) is the number of simulations where the computed $PI_0(\mathcal{C})$ is not greater than the PI_{obs}-value. R represents the total number of simulations. In both the numerator and the denominator one is added to account for the observed data.

Using PI as a measure of spatial interaction, we can define a statistically significant interaction pattern $\mathcal{C}$ as:

Definition 4. *An interaction pattern $\mathcal{C} = \{f_1, f_2, \ldots, f_k\}$ is statistically significant co-location pattern at level α, if the probability (p-value) of seeing, in a data set conforming to our null hypothesis, a PI-value of $\mathcal{C}$ larger than or equal to the observed PI-value is not* *greater than α.*

Definition 5. *An interaction pattern $\mathcal{C} = \{f_1, f_2, \ldots, f_k\}$ is* ***statistically significant segregation pattern*** *at level α, if the probability (p-value) of seeing, in a data set conforming to our null hypothesis, a PI-value of $\mathcal{C}$ smaller than or equal to the observed PI-value is not greater than α.*

3.6 Number of Required Simulations

How many simulations do we need to get a good critical region for the test statistic? We do two-tailed tests since we are looking for both positive and negative spatial dependency among interacting features. Marriot's investigation shows that the critical region for the test statistic becomes 'blurred' with smaller number of simulations resulting in a loss of power of the test [42]. This blurring can be reduced for a large value of R which again results an increase in the computational cost. At α, the one-sided critical value is the $\alpha(R+1)$-th largest (in case of positive dependency) or the $\alpha(R+1)$-th smallest (in case of negative dependency) value out of R simulations. To get a good critical region for the test statistic, Besag and Diggle in [12] suggest the number of simulations to be computed as $\alpha(R+1) = 5$. Accordingly, 499 simulations are required for $\alpha = 0.01$, 99 simulations are required for $\alpha = 0.05$.

3.7 Summary

Using examples and a real data set, this chapter first discusses the limitations of the current approaches. The motivation of finding statistically sound patterns is then stated. This chapter then formulates our objective of mining statistically sound patterns. Some key concepts and definitions are also provided at this point for a better understanding of our algorithms presented in the next chapter.

Chapter 4

Statistically Significant Interaction Pattern Mining

Given a set of spatial features, our objective is to mine all statistically significant co-location and segregation patterns of different sizes. In this chapter, we propose two methods to mine statistically significant co-location and segregation patterns of different sizes for a given distance threshold. Then we conduct experimental evaluation to validate our approaches.

4.1 Algorithms

The computational cost of mining statistically significant patterns incurs at two steps - 1) data generation step during the simulation runs, and 2) the PI-value computation step of each candidate interaction pattern. To determine if a pattern will be reported as significant or not, our first approach requires all the pattern instances. In our second approach, we show that the significance of a pattern can be determined using only a subset of the total pattern instances. By using less pattern instances, our second approach thus achieves a computational gain over our first approach. We first describe a naïve approach to mine statistically significant patterns and then propose some strategies for reducing the overall computational cost.

4.1.1 A Naïve Approach

For each interaction pattern $\mathcal{C}$ we have to compute the probability of the observed PI-value under the null hypothesis. For that, we have to determine the PI-value of each interaction pattern $\mathcal{C}$ in each simulation run by identifying all of $\mathcal{C}$'s instances, which naïvely amounts to checking the neighborhoods of each participating feature in $\mathcal{C}$. In the following, we describe both the data generation step and the PI-value computation step.

Data Generation for the Simulation Runs

In a simulation, we generate instances of each feature. A feature will be uniformly and independently distributed over the study area if it is uniformly and independently distributed in the observation. If a feature f_i is found to be auto-correlated in the observation, we first estimate the summary statistics using the Matérn's cluster model. The estimated statistics are 1) the intensity (κ) of the generated cluster centers (or parent events), 2) the cluster radius (r), and 3) the mean number of offsprings per cluster (μ) that are independently and uniformly distributed inside the cluster. Using these summary statistics we generate offsprings which gives the instances of f_i in a simulation run. If the total number of generated instances of f_i becomes less than the number of instances of f_i found in the observation, we distribute the remaining instances evenly on the clusters. On the other hand, if the total number of generated instances of f_i becomes higher than the number of instances of f_i found in the observation, we remove the extra instances by deleting them evenly from the clusters. Fig. 4.1 shows how instances of two auto-correlated features $\circ$ and $\triangle$ are generated in a single simulation run.

The *p*-value Computation for a Candidate Pattern

For each candidate interaction pattern $\mathcal{C}$ in the observed data, we first compute its PI-value, i.e. $PI_{\text{obs}}(\mathcal{C})$ and store them. To calculate the p-values p_{pos} and p_{neg}, we maintain two counters for the PI_{obs}-value of $\mathcal{C}$: $R^{\geq PI_{\text{obs}}}$ and $R^{\leq PI_{\text{obs}}}$. In a single simulation run R_i, for each candidate pattern $\mathcal{C}$ we compute its PI-value, i.e. $PI_0^{R_i}(\mathcal{C})$ and compare with the $PI_{\text{obs}}(\mathcal{C})$. $R^{\geq PI_{\text{obs}}}$ is incremented by one if $PI_0^{R_i}(\mathcal{C}) \geq PI_{\text{obs}}(\mathcal{C})$. The other counter, $R^{\leq PI_{\text{obs}}}$, is incremented if $PI_0^{R_i}(\mathcal{C}) \leq PI_{\text{obs}}(\mathcal{C})$. A candidate interaction pattern $\mathcal{C}$ will be reported as a statistically significant (at level α) co-location or segregation pattern, if $p_{\text{pos}} \leq \alpha$ or $p_{\text{neg}} \leq \alpha$, respectively, after all simulations.

4.1.2 All-instance-based SSCSP Approach

To improve the runtime of our naïve implementation, we propose another approach. We name it as *all-instance-based* SSCSP approach. In this approach we apply the following

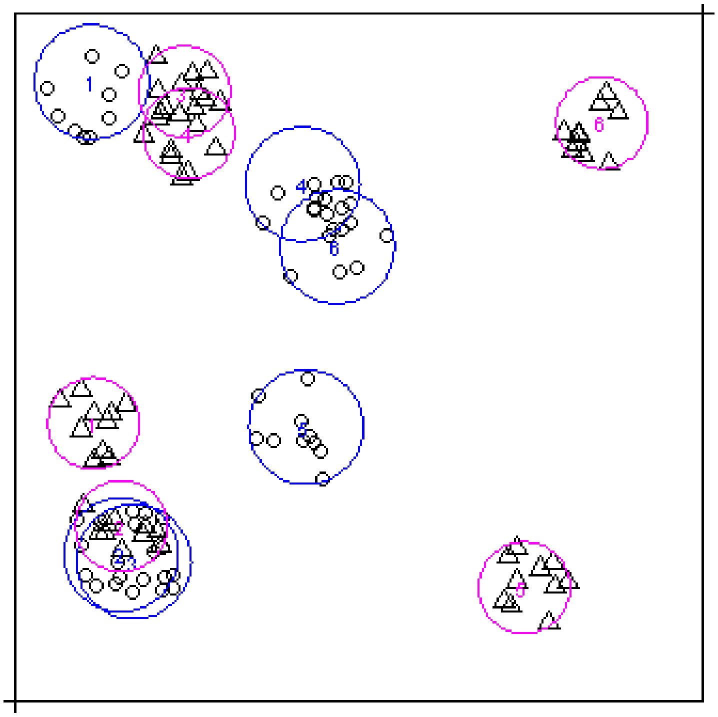


Figure 4.1: All instances of two auto-correlated features ○ and △.

strategies to reduce the cost of the data generation step and the p-value computation step.

Data Generation for the Simulation Runs

In a simulation, we generate instances of each feature. For an auto-correlated feature, we only generate instances of those clusters which are close enough to different features (auto-correlated or not) to be potentially involved in interactions. We avoid generating instances in a cluster c_{f^*} (radius R_{f^*}) of a feature f^* if the center of c_{f^*} is farther away than $R_{f^*} + R_d$ from every instance of different features. For auto-correlated features f_i ($f_i \neq f^*$), we can determine this without looking at the instances of f_i by checking only that the center of c_{f^*} is farther away than $R_{f^*} + R_{f_i} + R_d$ from the centers of all clusters of f_i.

Fig. 4.2 shows the partial amount of instances generated using the above described strategy. The computed PI-value from the data shown in Fig. 4.2 will be the same as the PI-value that would be computed from all instances as in Fig. 4.1. However by generating less feature instances we save computational time.

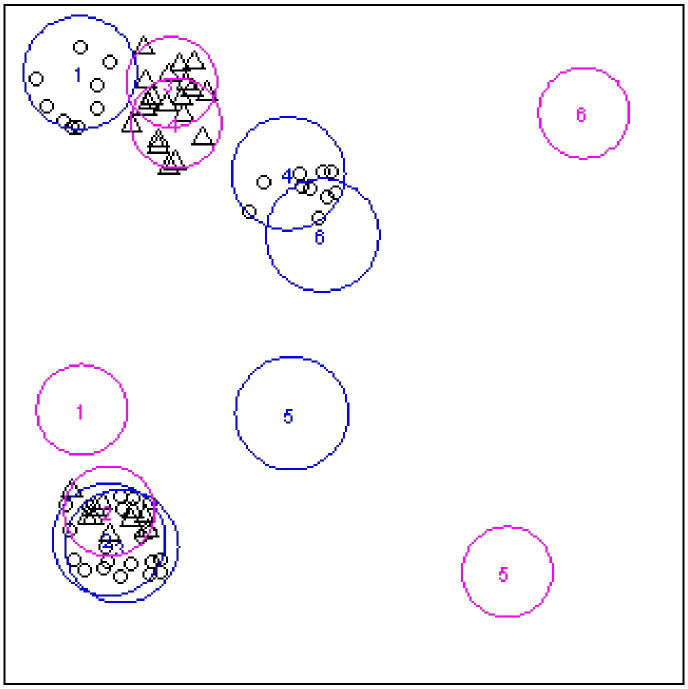


Figure 4.2: Generated instances of two auto-correlated features ○ and △.

The *p*-value Computation for a Candidate Pattern

We can reduce the total number of PI-value computations in a simulation run using the following property. The PI-value of a pattern $\mathcal{C}$ in a simulation run R_i, $PI_0^{R_i}(\mathcal{C})$, must be smaller than $PI_{\text{obs}}(\mathcal{C})$ ($PI_0^{R_i}(\mathcal{C}) < PI_{\text{obs}}(\mathcal{C})$), if a subset $\mathcal{C}' \subsetneqq \mathcal{C}$ exists for which $PI_0^{R_i}(\mathcal{C}') < PI_{\text{obs}}(\mathcal{C})$.

Lemma 2. *For an interaction pattern $\mathcal{C}$ and a simulation R_i, if there is a subset $\mathcal{C}' \subsetneqq \mathcal{C}$ such that $PI_0^{R_i}(\mathcal{C}') < PI_{obs}(\mathcal{C})$, then $PI_0^{R_i}(\mathcal{C}) < PI_{obs}(\mathcal{C})$.*

Proof. Assume $PI_0^{R_i}(\mathcal{C}') < PI_{\text{obs}}(\mathcal{C})$. According to lemma 1, $PI_0^{R_i}(\mathcal{C}') \geq PI_0^{R_i}(\mathcal{C})$, thus it follows that $PI_0^{R_i}(\mathcal{C}) < PI_{\text{obs}}(\mathcal{C})$. □

We can apply lemma 2 to prune the actual computation of $PI_0^{R_i}(\mathcal{C})$ and just increment the counter $R^{\leq PI_{\text{obs}}}$ whenever we know of a subset $\mathcal{C}' \subsetneqq \mathcal{C}$ for which $PI_0^{R_i}(\mathcal{C}') < PI_{\text{obs}}(\mathcal{C})$. Again $R^{\geq PI_{\text{obs}}}$ of $\mathcal{C}$ will not be incremented when $PI_0^{R_i}(\mathcal{C}') < PI_{\text{obs}}(\mathcal{C})$ holds. To apply lemma 2 efficiently, the PI-values of $\mathcal{C}$'s subsets that we want to check have to be readily available. If we check the PI-values of patterns in order of increasing pattern size, we could, in principle, store the PI-values of shorter patterns so that they are available when checking patterns of larger sizes. However, this approach could require a large space overhead, and actually checking too many subsets may overall not reduce the computational cost. Another issue with such an approach would be that the PI-values of some subsets would not computed because of the same pruning strategy. Therefore, we propose to use only the subsets of size 2 for checking, since their PI-values are all computed initially. Although storing $\binom{n}{2}$ PI-values requires some space, we can reuse the same space for each simulation run during the randomization tests. While checking the subsets of size 2, if one is found for which the lemma applies, we will stop checking the remaining subsets of size 2. If after checking all subsets of size 2, the lemma could not be applied, we compute $PI_0^{R_i}(\mathcal{C})$ and compare with $PI_{\text{obs}}(\mathcal{C})$. $R^{\geq PI_{\text{obs}}}$ of $\mathcal{C}$ is then incremented if $PI_0^{R_i}(\mathcal{C}')$ is not less than $PI_{\text{obs}}(\mathcal{C})$, otherwise not.

Here we clarify the pruning strategy using four features $A, B, C,$ and D. First we compute PI_{obs} for each candidate interaction pattern. In a single simulation run R_i, we start with computing the $PI_0^{R_i}$ of each 2-size pattern and increment $R^{\geq PI_{\text{obs}}}$ or $R^{\leq PI_{\text{obs}}}$ of a pattern by 1. Lets consider a 3-size pattern $\{A, B, C\}$. Assume $PI_0^{R_i}\{A, B\} < PI_{\text{obs}}\{A, B, C\}$, then $PI_0^{R_i}\{A, B, C\} < PI_{\text{obs}}\{A, B, C\}$. Hence we can just increment the counter $R^{\leq PI_{\text{obs}}}$ of $\{A, B, C\}$ without even checking the $PI_0^{R_i}\{A, B, C\}$-value and thus the computation of $PI_0^{R_i}\{A, B, C\}$ is no longer required. Similarly the decision for the 4-size pattern $\{A, B, C, D\}$ is also done by checking its 2-size subsets. Note that we can not prune its PI-value computation based on the fact that we could prune the PI-value computation for $\{A, B, C\}$ because $PI_{\text{obs}}\{A, B, C, D\}$ will, in general, be different from $PI_{\text{obs}}\{A, B, C\}$ and it could still be possible that $PI_0^{R_i}\{A, B, C, D\} \geq PI_{\text{obs}}\{A, B, C, D\}$. The PI-value decreases with the increase of the pattern size. Hence, if the number of features increases, we will see more pruning effect in smaller size interaction patterns than in larger size interaction patterns.

Pseudo-code

Algorithm 1 and 2 shows the pseudo-code for the complete procedure.

Complexity Analysis

In the worst case, there is no pruning in each simulation R_i and we compute the $PI_0^{R_i}$-value of each candidate interaction pattern $\mathcal{C}$. Before computing the $PI_0^{R_i}$-value of $\mathcal{C}$, we lookup the stored $PI_0^{R_i}$-values of its subsets of size 2. Hence the cost for $\mathcal{C}$ is the

Algorithm 1: SSCSP: Mining Statistically Significant Co-location and Segregation Patterns

```
Input: A Spatial data set $\mathcal{SD}$ with $N$ spatial features $\mathcal{S} = \{f_1, \ldots, f_N\}$ (each $f_i$ has
    $n_{f_i}$ instances).
    Level of significance $\alpha$, and total simulation runs $R$.
Output: Set of statistically significant co-locations $\mathbb{C}$ and segregation patterns $\mathbb{C}'$.
Variables:
    $k$: pattern size. $R_d$: interaction radius.
    $\mathcal{C}^k_{\text{obs}}$: Set of all $k$-size interaction patterns. Each patterns is stored along with its
    $PI_{\text{obs}}$-value, $R^{\geq PI_{\text{obs}}}$-value, and $R^{\leq PI_{\text{obs}}}$-value.
    $\mathcal{C}^2_{\text{null}}$: Set of all 2-size interaction patterns in a simulation. Each pattern is stored
    along with its $PI^{R_j}_{\text{null}}$-value from a simulation run $R_j$.
Method:
$\mathbb{C} \leftarrow \{\}$ and $\mathbb{C}' \leftarrow \{\}$
// Compute $PI_{\text{obs}}$-value of all interaction patterns from $\mathcal{SD}$
for $k = 2$ to $N$ and $i = 1$ to $\binom{N}{k}$ do
    Generate $k$-size $i$-th interaction pattern and store it in $\mathcal{C}^k_{\text{obs}}[i]$.pattern
    Compute its $PI_{\text{obs}}$-value
    $\mathcal{C}^k_{\text{obs}}[i].PI \leftarrow PI_{\text{obs}}$; $\mathcal{C}^k_{\text{obs}}[i].R^{\geq PI_{\text{obs}}} \leftarrow 0$; $\mathcal{C}^k_{\text{obs}}[i].R^{\leq PI_{\text{obs}}} \leftarrow 0$
// Computing $p_{\text{pos}}$-value and $p_{\text{neg}}$-value for all interaction patterns
for $j = 1$ to $R$ do
    Generate a simulated data set $R_j$ under the null model
    for $i = 1$ to $\binom{N}{2}$ do
        Compute its $PI^{R_j}_{\text{null}}$-value and $\mathcal{C}^2_{\text{null}}[i].PI \leftarrow PI^{R_j}_{\text{null}}$
        if $\mathcal{C}^2_{\text{null}}[i].PI \geq \mathcal{C}^2_{\text{obs}}[i].PI_{\text{obs}}$ then
            Increment $\mathcal{C}^2_{\text{obs}}[i].R^{\geq PI_{\text{obs}}}$
        if $\mathcal{C}^2_{\text{null}}[i].PI \leq \mathcal{C}^2_{\text{obs}}[i].PI_{\text{obs}}$ then
            Increment $\mathcal{C}^2_{\text{obs}}[i].R^{\leq PI_{\text{obs}}}$
    for $k = 3$ to $N$ and $i = 1$ to $\binom{N}{k}$ do
        if (isPrunedCand($\mathcal{C}^k_{\text{obs}}[i]$.pattern, $\mathcal{C}^k_{\text{obs}}[i].PI$, $\mathcal{C}^2_{\text{null}}$, $k$)) then
            Increment $\mathcal{C}^k_{\text{obs}}[i].R^{\leq PI_{\text{obs}}}$
            continue // Skip computation of $PI^{R_j}_{\text{null}}$-value
        Compute the $PI^{R_j}_{\text{null}}$-value of an interaction pattern $\mathcal{C}^k_{\text{obs}}[i]$.pattern
        if $PI^{R_j}_{\text{null}} \geq \mathcal{C}^k_{\text{obs}}[i].PI$ then
            Increment $\mathcal{C}^k_{\text{obs}}[i].R^{\geq PI_{\text{obs}}}$
        if $PI^{R_j}_{\text{null}} \leq \mathcal{C}^k_{\text{obs}}[i].PI$ then
            Increment $\mathcal{C}^k_{\text{obs}}[i].R^{\leq PI_{\text{obs}}}$
for $k = 2$ to $N$ and $i = 1$ to $\binom{N}{k}$ do
    Compute $p_{\text{pos}}$-value and $p_{\text{neg}}$-value of $\mathcal{C}^k_{\text{obs}}[i]$.pattern
    if $p_{\text{pos}} \leq \alpha$ then
        $\mathbb{C} \leftarrow \mathbb{C} \bigcup \mathcal{C}^k_{\text{obs}}[i]$.pattern
    else
        if $p_{\text{neg}} \leq \alpha$ then
            $\mathbb{C}' \leftarrow \mathbb{C}' \bigcup \mathcal{C}^k_{\text{obs}}[i]$.pattern
return $\mathcal{C}$
```

Algorithm 2: isPrunedCand(CandPattern, PI_{obs}, $\mathcal{C}^2_{\text{null}}$, k)

```
for each 2-size subset $l$ of CandPattern do
    Look for position $x$ of $l$ in $\mathcal{C}^2_{\text{obs}}$.pattern.
    if $\mathcal{C}^2_{\text{null}}[x].PI < PI_{\text{obs}}$ then
        return TRUE
return FALSE
```

sum of the lookup cost and the cost for computing its $PI_0^{R_i}$-value. Assume that a lookup costs β units of computation. For a pattern $\mathcal{C}$ of size k, the lookup cost for its $\binom{k}{2}$ pairs is $P_1^k = \binom{k}{2}\beta$. For computing $PI_0^{R_i}(\mathcal{C})$, we lookup the neighborhoods of all instances of each feature in $\mathcal{C}$ and determine if at least one instance of each feature in $\mathcal{C}$ is present in a neighborhood. Hence the cost of PI-value computation for $\mathcal{C}$ of size k is $P_2^k = k \times \max_k\{\text{\# of instances of feature } f_k\} \times \beta = k\delta\beta$ [assume $\delta = \max_{i=1}^{n}\{\text{\# of instances of feature } f_i\}$]. With n total features, there are $\binom{n}{k}$ different k-size interaction patterns. Hence the total cost for all different k-size interaction patterns is $\binom{n}{2}P_2^2 + \sum_{k=3}^{n}\binom{n}{k}\left(P_1^k + P_2^k\right)$. Using the equalities of $\sum_{k=q}^{n}\binom{n}{k}\binom{k}{q} = 2^{n-q}\binom{n}{q}$ and $\sum_{k=2}^{n} k\binom{n}{k} = n(2^{n-1}-1)$, the above cost is equal to $\binom{n}{2}(2^{n-2}-1)\beta + n(2^{n-1}-1)\delta\beta$ which is of $O(2^n)$ in the worst case.

While the worst case is expensive, in many important, real applications (e.g. in ecology), the largest pattern size that typically exists in the data is much smaller than total number of features n, since a finite interaction neighborhood can typically not accommodate instances of n different features when n is large. In such applications, the actual cost in practice is much lower than the worst case. While checking neighborhoods of feature instances, we can determine the size of the largest interaction pattern, and then restrict our search to only patterns up to this size (instead of all sizes).

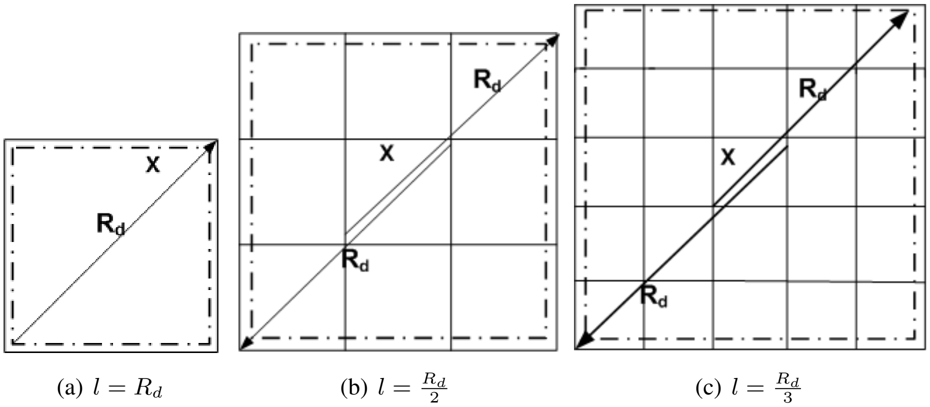
4.1.3 A Sampling Based Approach

In SSCSP the PI-value of a pattern $\mathcal{C}$ is computed, as a test statistics, using all the instances of $\mathcal{C}$. Here we propose a prevalence measure PI^* as a test statistics that is computed efficiently from a subset of the instances of $\mathcal{C}$. The new prevalence measure PI^* can be seen as an approximation of the original PI-value, which leads, in most cases, to the same statistical inferences. We propose a grid based partitioning approach to identify the instances of $\mathcal{C}$ that are considered to compute the approximate prevalence measure, efficiently. This gives a computational advantage over the *all-instance-based* SSCSP as the cost of identifying pattern instances and computing the PI-value is now less.

If a true co-location or segregation relationship exists among a group of features $\mathcal{C}$, this should be reflected even in a subset of the total instances of $\mathcal{C}$, and a statistical test should be able to capture this dependency from such a subset. Instead of looking at the full neighborhood S_o of a feature instance I, we consider only a sub-region S of S_o. By considering a larger sub-region which covers more area of S_o, the computed PI^*-value will be more similar to the original PI-value. Lets consider two features A and B that could potentially be involved in an interaction. Instead of looking in the neighborhood S_o of an instance I_A of feature A for instances of feature B, we consider a sub-region S around I_A. An instance I_B found in S then counts towards a spatial interaction with I_A. Only an I_A found in S together with I_B is considered for the PI-value computation. By checking the sub-regions around all the instances of A and B, we compute participation ratios PR^* of A and B based on these reduced neighborhoods, and obtain the PI^*-value of $\{A, B\}$ by taking the minimum.

In our randomization tests, we obtain the distribution of the PI^*-values under the null model. Thus for two features A and B, we compute PI^*-values in a simulation using similar sub-regions for the instances A and B. From all the PI^*-values of $\{A, B\}$ computed from the simulations, we get a distribution under the null model. Such distribution can be viewed as an approximation of the distribution of the actual PI-values. We argue that if two features A and B are truly associated or segregated, the observed PI^*-value in a given data set should also be statistically significant when compared to the distribution of the PI^*-values under the null model. If A and B are independent of each other, this should also be reflected by the observed PI^*-value being closer to the expected PI^*-value under the null model.

Our experimental results show that this approach works extremely well, in general. However, this approach might miss reporting a true co-location only in cases where co-located features have very few instances. For instance, let A and B have very few instances and have a true spatial dependency. Hence the number of instances of $\{A, B\}$ will also be few. If these instances do not appear in the sub-regions, the sampling approach will fail to report the spatial dependency between A and B. To improve the chances of finding pattern instances in such a case, we can either (1) increase the area of a sub-region so that the sub-region can match better with the full neighborhood of a feature or (2) use *all-instance-based* SSCSP approach as here the cost of identifying even all the pattern instances will not be high. A more detailed reasoning regarding the accuracy of this approach is given at the end



(a) $l = R_d$ (b) $l = \frac{R_d}{2}$ (c) $l = \frac{R_d}{3}$

Figure 4.3: Dashed bordered region is a sampled neighborhood for a feature instance present anywhere of cell X.

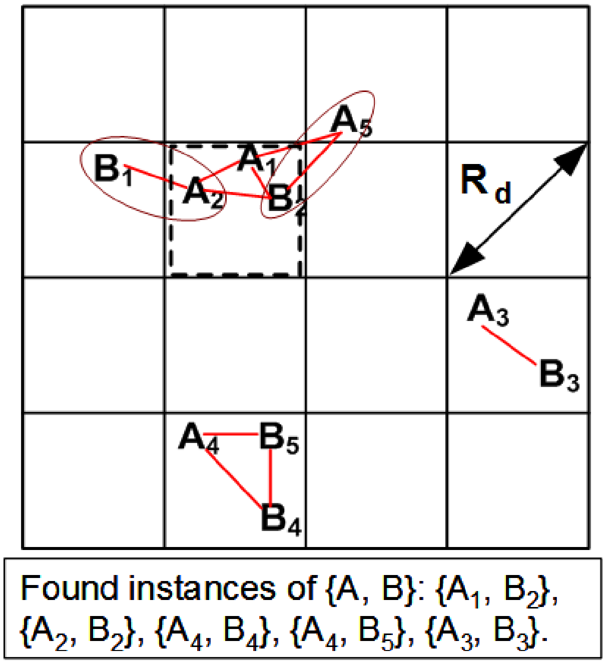
of this chapter.

A Neighborhood Sampling Approach Using a Grid Based Space Partitioning

To select sub-regions of actual neighborhoods, a grid is placed over the whole study area. Each grid cell is a square with a diagonal length l being equal to $\frac{R_d}{w}$, where R_d is the interaction neighborhood radius and $w \geq 1$ is an integer.

If $l = R_d$, the selected sub-region represents a sampled neighborhood for a feature instance I that consists of a single cell X that contains I. If $l = \frac{R_d}{2}$, the sampled neighborhood consists of the cell X that contains I, plus the 8 cells surrounding X. In general, if $l = \frac{R_d}{w}$, the sampled neighborhood of I consists of $(2w-1)^2$ cells including X. We denote the corresponding neighborhood by $S_{(2w-1)^2}$. Fig. 4.3 illustrates the sampled neighborhoods for w equal to 1, 2, and 3, i.e., S_1, S_9, and S_{25}. Note that any other feature instance located in a sampled neighborhood of I is necessarily involved in an interaction with I by construction.

For instance, if $l = R_d$, feature instances present in the same cell are all involved in an interaction. We look for the instances of an interaction pattern $\mathcal{C}$ in cells where instances of a participating feature f_i of $\mathcal{C}$ are present. However we can check all cells instead when the total number of instances of f_i is greater than the total number of cells of the grid. An instance of $\mathcal{C}$ will be counted for the computation of PI^* only if all the participating feature instances are present in a single sub-region, i.e. a single cell when $l = R_d$. Only the instances found from the sub-regions are considered to compute the $PI^*(\mathcal{C})$. Hence a valid

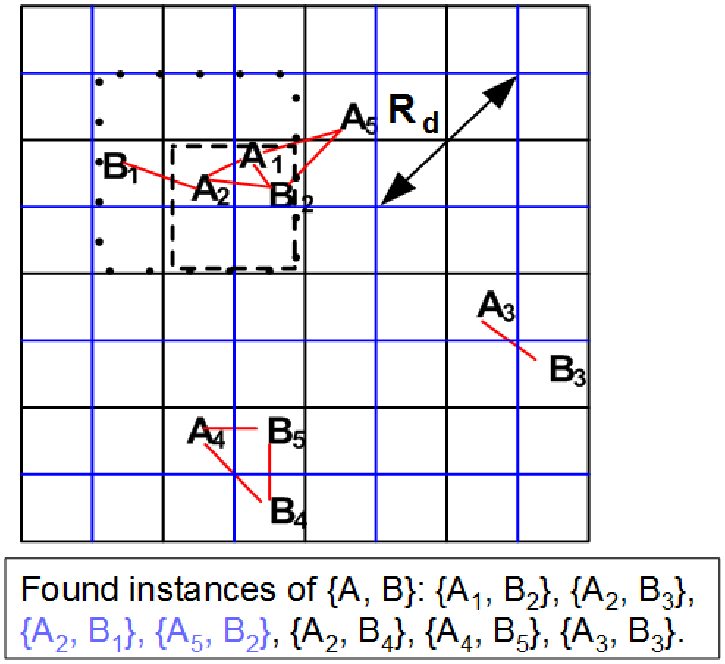


Found instances of {A, B}: {A_1, B_2}, {A_2, B_2}, {A_4, B_4}, {A_4, B_5}, {A_3, B_3}.

Figure 4.4: Interaction instances identified using a grid where $l = R_d$.

instance of $\mathcal{C}$ will not be considered in computing the PI^*-value, if the participating feature instances are split across different cells. For instance in Fig 4.4 all feature instances of A and B are are involved in spatial interaction. However, the sampling approach with $l = R_d$ will not count the patten instances $\{A_2, B_1\}$ and $\{A_5, B_2\}$ since their participating feature instances are located across different cells. Considering only the instances of $\{A, B\}$ that are found in single cells, the PI^*-value is computed as $\min\{\frac{4}{5}, \frac{4}{5}\} = \frac{4}{5}$, whereas the actual PI-value is 1. However, when doing randomization tests, we will also miss similarly pattern instances in the simulations that are not contained in a single cell, and hence compare the observed PI-value to a distribution of the PI-values that are computed in the same way.

Now we show that by using a finer grid, we can increase the count of interaction pattern instances used to compute PI^*. Fig. 4.4 depicts the sampling approach where $l = R_d$. A sampled sub-region with a feature instance A_2 is shown as a dashed bordered region. Fig. 4.5 depicts the sampling approach where $l = \frac{R_d}{2}$. Here the sub-region for A_2 is shown using a dotted bordered region. We find that this sub-region (dotted bordered region) includes the dashed bordered sub-region shown in Fig 4.4 and some additional space. Due to the increased area, instance $\{A_2, B_1\}$ is not missed now.



Found instances of {A, B}: {A1, B2}, {A2, B3}, {A2, B1}, {A5, B2}, {A2, B4}, {A4, B5}, {A3, B3}.

Figure 4.5: Interaction instances identified using a grid where $l = \frac{R_d}{2}$.

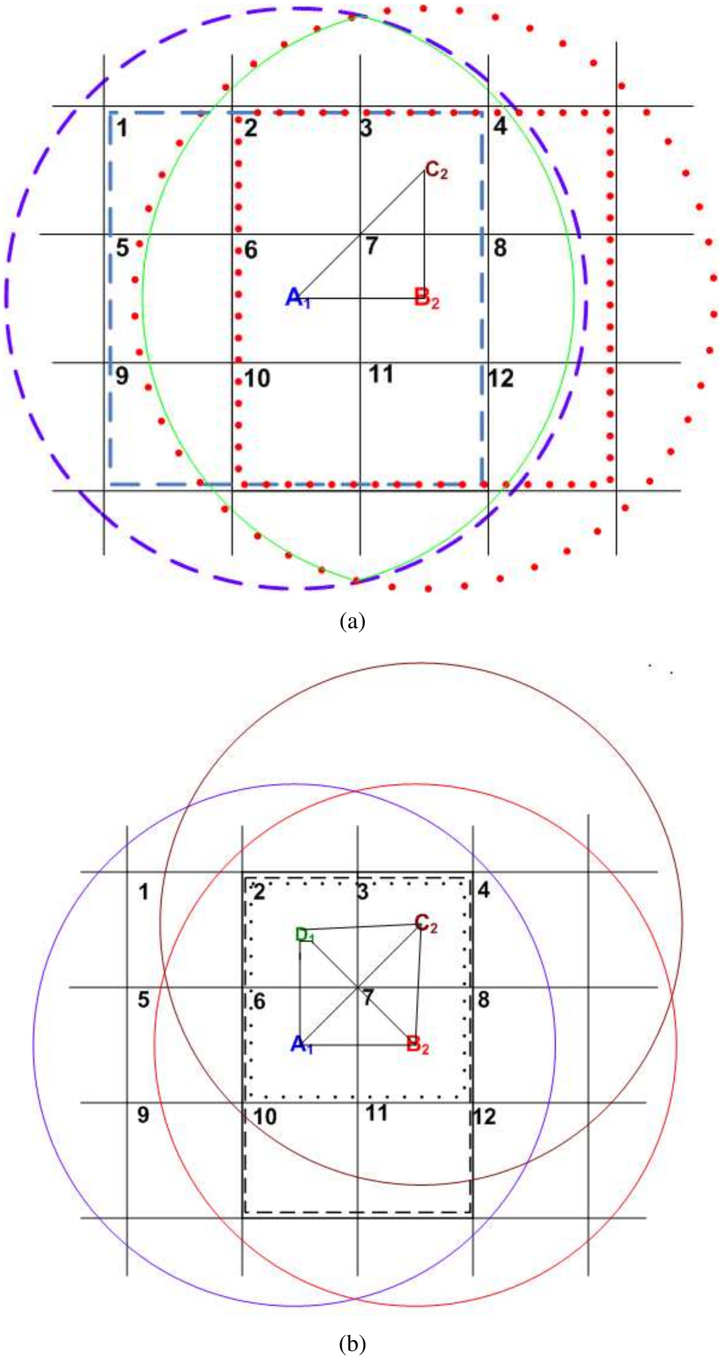
The actual neighborhood of a feature instance f_i is a circular region S_o centered at f_i and the area is πR_d^2. The area of a sampled neighborhood S_1 (Fig. 4.3(a)) is $\frac{R_d^2}{2}$. Hence S_1 covers $\frac{1}{2\pi}$ of S_o for any feature instance I appearing in S_1. When $w = 2$, the area of the sampled neighborhood S_9 (Fig. 4.3(b)) is $\frac{9R_d^2}{8}$, and it covers $\frac{9}{8\pi}$ of S_o for any I in X. The sampled neighborhood S_9 is 2.25 times larger than S_1. When $w = 3$, the area of S_{25} (Fig. 4.3(c)) becomes 2.78 times larger than S_1. In general the area of $S_{(2w-1)^2}$ is $\frac{R_d^2(2-\frac{1}{w})^2}{2}$ and it covers $\frac{(2-\frac{1}{w})^2}{2\pi}$ of the actual neighborhood S_o.

The area of $S_{(2w-1)^2}$ slowly increases with increasing w, but is limited while the number of cells that have to be checked increases fast with increasing w. When $w \to \infty$, a sampled neighborhood will cover 0.64 of S_o ($\lim_{w\to\infty} \frac{(2-\frac{1}{w})^2}{2\pi} * S_o = 0.64 * S_o$), while the sampled neighborhood with increasing w is comprised of $(2w-1)^2 \to \infty$ number of cells, all of which have to be checked for patterns of size 2. For instance, when the value of w equals 1, 2, 3 or 4 the sampled neighborhood respectively covers 0.16, 0.36, 0.44, and 0.48 of the circular region S_o and is respectively comprised of 1, 9, 25 or 49 cells, all of which have to be checked for patterns of size 2. When $w = 3$, the sampled neighborhood gives 23% more coverage on S_o compared to the sampled neighborhood with $w = 2$. However the number of cells in the sampled neighborhood with $w = 3$ is increased by 177%. Similarly, when $w = 4$ the sampled neighborhood gives 34% more coverage on S_o compared to the sampled neighborhood with $w = 2$ and the number of cells of the sampled neighborhood is increased by 444%. However the number of cells checked in a sampled neighborhood decreases with increasing pattern size. To find a k-size interaction pattern instance, we check the overlapping region of the neighborhoods of the $k - 1$ participating feature instances for the presence of an instance of the k-th feature. For instance, to find $\{A_1, B_2, C_2\}$ in a grid where $l = \frac{R_d}{2}$ (Fig. 4.6(a)), we check the overlapping region of the sampled neighborhoods of A_1 and B_2. In Fig. 4.6(a), the actual neighborhood of a feature instance is shown by a circle; whereas the sampled neighborhoods for A_1 and B_2 are shown by dashed and dotted squares, respectively. Here the overlapping region of the two sub-regions includes 6 cells $(2, 3, 6, 7, 10,$ and $11)$ and we can restrict the search for an instance of $\mathcal{C}$ to these 6 cells. Similarly, to find $\{A_1, B_2, C_2, D_1\}$ the overlapping region of the sampled neighborhoods of $A_1, B_2,$ and C_2 must be checked when looking for an instance of D. In the example shown in Fig. 4.6(b), the overlapping region includes only 4 cells (2, 3, 6, and 7) indicated by a dotted line.

Clearly, there is a trade-off between the quality of the PI^*-value as a test statistic to determine spatial associations and the resolution of the grid that induces the sampled neighborhoods based on which the PI^*-values are determined. Note, however, that achieving the best accuracy of the neighborhood approximation is not necessary. Since we are making the same kind of error when computing PI^*-values both for the observed data set, as well as in all the simulations, what matters is whether the distribution of the PI^*-values will lead to the same statistical inference about which patterns are statistically significant. Fig. 4.7 shows 4 empirical distributions of approximate (for $w = 1$, $w = 2$, $w = 3$ and $w = 4$) PI-values and Fig. 4.7 shows the empirical distribution of the actual PI-values (computed from all the instances) of a pattern $\{A, B\}$ estimated under the null model (i.e., for two independent features A and B). We can see that using a finer cell resolution for the grid, the distribution of PI^*-values of $\{A, B\}$ becomes more similar to the the distribution of actual PI-values. In our experiments, we will demonstrate that values for w equal to 4, 3, or even as low as 2 work well in all cases, except for patterns that involve features with an extremely low number of instances.

Accuracy of the Sampling Approach

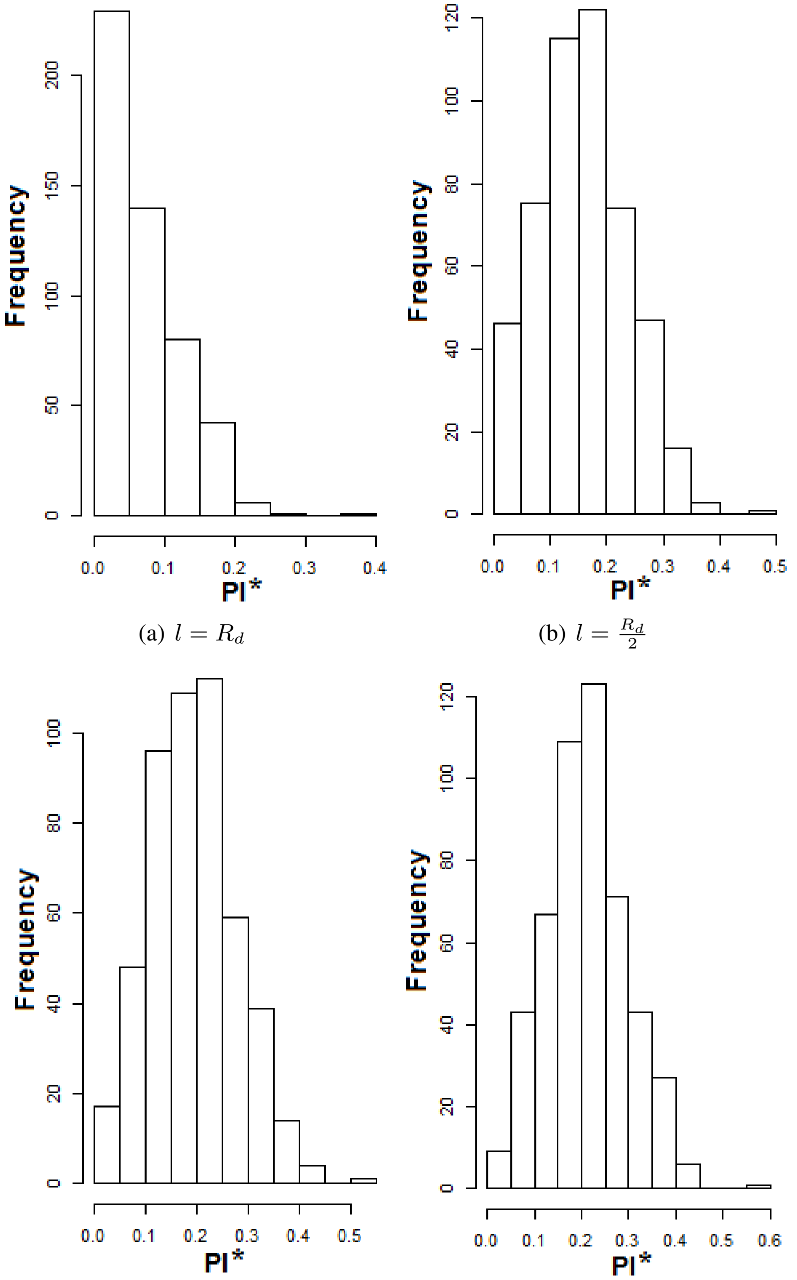
The accuracy of our sampling approach in making a statistical inference on a pattern's significance depends on the area of the sampling sub-region, and the number of interacting



(a)

(b)

Figure 4.6: Finding an interaction pattern instance a) of size 3. b) of size 4.



(a) $l = R_d$

(b) $l = \frac{R_d}{2}$

(c) $l = \frac{R_d}{3}$

(d) $l = \frac{R_d}{4}$

Figure 4.7: Distribution of the PI^*-values computed under the null model.

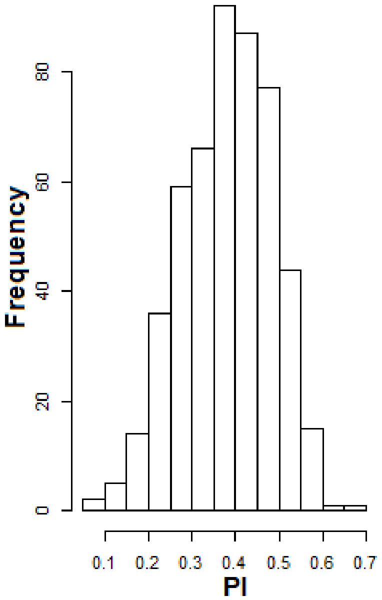


Figure 4.8: Distribution of the PI-values computed under the null model.

instances of the participating features of a pattern. We show these relationships, more formally, for the special case of an interaction pattern $\mathcal{C} = \{f_1, f_2\}$ with two features. For larger pattern sizes, it is intuitively clear that the same relationships hold, but a formal analysis will be much more complex.

Let us assume that there are $n_{\mathcal{C}}$ instances of $\mathcal{C}$, and that f_1 and f_2 have $n_{f_1}^{\mathcal{C}}$ and $n_{f_2}^{\mathcal{C}}$ number of instances, respectively, that are participating in interaction type $\mathcal{C}$. Instances of f_1 and f_2 that are involved in interaction type $\mathcal{C}$ are denoted by $I_{f_1}^{\mathcal{C}}$ and $I_{f_2}^{\mathcal{C}}$, respectively. The average number of instances of f_2 that are found in the non-approximated interaction neighborhood S_o of an instance $I_{f_1}^{\mathcal{C}}$ is denoted by $\bar{n}_{f_2}^{S_o}$. Let us assume that these instances are uniformly distributed in S_o. Then, $\bar{n}_{f_2}^{S_o} = \frac{n_{f_2}^{\mathcal{C}}}{n_{f_1}^{\mathcal{C}}}$. The average number of instances of f_1 that are found in the non-approximated interaction neighborhood of an instance $I_{f_2}^{\mathcal{C}}$ can be defined analogously. Without loss of generality, we analyse neighborhoods with feature instances of f_1 at their centers (the analysis for feature instances of f_2 is completely analogous).

According to the construction, a sampled sub-region S is always inside of S_o. Hence the probability $\mathbb{P}$ that an instance present in S_o will also be present in S is given as $\mathbb{P} = \lambda_2(S)/\lambda_2(S_o)$, where $\lambda_2(S)$, and $\lambda_2(S_o)$ denote the area of S and S_o, respectively.

Consider an instance $I_{f_1}^{\mathcal{C}}$ which is interacting with $\bar{n}_{f_2}^{S_o}$ instances of f_2, on average, in S_o. $I_{f_1}^{\mathcal{C}}$ will also be interacting with instances of f_2 in S if at least one of these instances is also present in S. The probability ($P_{f_1}^{S}$) that $I_{f_1}^{\mathcal{C}}$ is interacting with at least one instance of f_2 in

S is given as $P_{f_1}^{S} = \Big(1 - (1-\mathbb{P})^{\bar{n}_{f_2}^{S_o}}\Big)$. This probability converges to 1 with the increase of $\lambda_2(S)$ and with the average number of interacting feature instances of f_2. In other words, the chance of *missing* the feature instance $I_{f_1}^{\mathcal{C}}$ as interacting with feature f_2, when using a sampled sub-region S, is equal to $(1-\mathbb{P})^{\bar{n}_{f_2}^{S_o}}$, and this chance decreases with increasing size of the sampled sub-region.

So far we have analysed the dependency on the size of the sampled sub-region; now we turn to the dependency on the number of interacting instances of the participating features of a pattern.

Let X_i be a random variable so that $X_i = 1$ if the i^{th} ($i \leq n_{f_1}^{\mathcal{C}}$) instance $I_{f_1}^{\mathcal{C}}$ is identified as interacting with f_2 in the sampled sub-region S, 0 otherwise. $X_i \sim$ Bernoulli($P_{f_1}^{S}$), where $X_1, \cdots, X_{n_{f_1}^{\mathcal{C}}}$ are independent and $E[X_i] = P_{f_1}^{S}$. Let $X = \sum X_i$ be the number of instances of f_1 that are identified as interacting with f_2 in S. $X \sim$ Binomial($n_{f_1}^{\mathcal{C}}, P_{f_1}^{S}$) and the expected number of instances of f_1 that will be identified as interacting with f_2 in S is equal to $\mu_{f_1}^{S} = E\Big[\sum X_i\Big] = \sum E[X_i] = P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}$. Dividing $\mu_{f_1}^{S}$ by the total number of instances of f_1, yields the expected participation ratio of f_1 using the sampling approach. The empirical fraction $\bar{X} = X/n_{f_1}^{\mathcal{C}}$ gives us an estimate of $P_{f_1}^{S}$.

The next question is how large $n_{f_1}^{\mathcal{C}}$ has to be in order to obtain a good estimate of $P_{f_1}^{S}$ for a given accuracy and a given confidence.

Chernoff in [32] gives an exponentially decreasing bound on tail distributions of sums of independent random variables. For any $\epsilon \geq 0$, a multiplicative form of the two-sided Chernoff bound with respect to X above is given as:

$$\mathbf{Pr}\Big[|\bar{X} - \mu_{f_1}^{S}| \geq \epsilon * \mu_{f_1}^{S}\Big] \leq 2 * \left[\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right]^{\mu_{f_1}^{S}} = 2 * [e^{\epsilon-(1+\epsilon)*\ln(1+\epsilon)}]^{\mu_{f_1}^{S}}$$

Using Taylor's series expansion of $\ln(1+\epsilon)$ and simplification, we obtain that $\ln(1+\epsilon) > \frac{2*\epsilon}{2+\epsilon}$. This implies that $\epsilon - (1+\epsilon)\ln(1+\epsilon) \leq \frac{-\epsilon^2}{2+\epsilon}$. Hence the inequality becomes

$$\begin{aligned} & \mathbf{Pr}\Big[|\bar{X} - \mu_{f_1}^{S}| \geq \epsilon * \mu_{f_1}^{S}\Big] \leq 2 * e^{\frac{-\epsilon^2}{2+\epsilon} * \mu_{f_1}^{S}} \\ \Leftrightarrow\ & \mathbf{Pr}\Big[|\bar{X} - P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}| \geq \epsilon * P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}\Big] \leq 2 * e^{\frac{-\epsilon^2}{2+\epsilon} * P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}} \\ \Leftrightarrow\ & \mathbf{Pr}\Big[|\bar{X} - P_{f_1}^{S}| \geq \epsilon * P_{f_1}^{S}\Big] \leq 2 * e^{\frac{-\epsilon^2}{2+\epsilon} * P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}}. \end{aligned}$$

Here for a fixed ϵ, the larger the term $P_{f_1}^{S} * n_{f_1}^{\mathcal{C}}$ is, the smaller the bound on the right of the inequality is. The smaller the value of $P_{f_1}^{S}$, the larger the number of interacting instances $n_{f_1}^{\mathcal{C}}$ of f_1 has to be in order to obtain a good estimate of $P_{f_1}^{S}$. For instance, if the participation ratio of f_1 is 0.4, a data set where 8 out of 20 total instances of f_1 interacting with f_2 gives a better estimate of $P^s_{f_1}$ than a data set where 4 out of 10 total instances of f_1 interacting with f_2 gives.

Pseudo-code

Algorithm 3 is the pseudo-code of our sampling approach.

Algorithm 3: Sampling Algorithm: Mining Statistically Significant Co-location and Segregation Patterns Using PI^*-value

```
Input: A Spatial data set $\mathcal{SD}$ with $N$ spatial features $\mathcal{S} = \{f_1, \ldots, f_N\}$ (each $f_i$ has
    $n_{f_i}$ instances).
    Level of significance $\alpha$, and total simulation runs $R$.
Output: Set of statistically significant co-locations $\mathbb{C}$ and segregation patterns $\mathbb{C}'$.
Variables:
    $k$: pattern size; $2 \leq k \leq N$.
    $\mathcal{C}$: an interaction pattern of size $k$. $R_d$: interaction radius.
    $w$: a factor (integer) which determines the cell size of the grid placed over the
    study area; $w \geq 1$.
Method:
$\mathbb{C} \leftarrow \{\}$ and $\mathbb{C}' \leftarrow \{\}$
Place a grid where the diagonal length of each cell is $\frac{R_d}{w}$.
for each pattern $\mathcal{C}$ of size $k$ do
    Find the instances of $\mathcal{C}$ from the sampled sub-regions.
    Compute the $PI^*_{\text{obs}}$ of $\mathcal{C}$
for $j = 1$ to $R$ do
    Generate a simulated data set $R_j$ under the null model
    Place a grid where the diagonal length of each cell is $\frac{R_d}{w}$.
    for each pattern $\mathcal{C}$ of size $k$ do
        Find the instances of $\mathcal{C}$ from the sampled sub-regions.
        if $\mathcal{C}$ is not pruned using lemma 2 then
            Compute the $PI^*_0$ of $\mathcal{C}$
            Increment $R^{\geq PI^*_{\text{obs}}}$ or $R^{\leq PI^*_{\text{obs}}}$
        else
            Increment $R^{\leq PI^*_{\text{obs}}}$
Compute $p_{\text{pos}}$ and $p_{\text{neg}}$
if $p_{\text{pos}} \leq \alpha$ then
    $\mathbb{C} \leftarrow \mathbb{C} \bigcup \mathcal{C}$
else
    if $p_{\text{neg}} \leq \alpha$ then
        $\mathbb{C}' \leftarrow \mathbb{C}' \bigcup \mathcal{C}$
```

Complexity

The complexity in the worst case is $O(2^n)$, where n is the total number of features. However, the lookup cost β of P_2^k (see Section 4.1.2) in a grid based sampling approach is smaller than the lookup cost of the approach that computes the actual PI-value. Here the lookup cost involves only checking the cells of all sampled sub-regions. The computation of inter-distance among features which is done in the *all-instance-based* approach is not required.

4.2 Experimental Evaluation

To validate our approaches, experiments are conducted using synthetic and real data sets. In this section we discuss our experimental procedures and discuss our results. For the experiments, we compare the *all-instance-based* approach with our sampling approach for four grid cell resolutions, given by $w = 1, 2, 3, 4$, as well as with a standard co-location mining approach.

4.2.1 Synthetic Data Sets

In this section, we conduct experiments with a set of synthetic data sets to demonstrate that our approaches do not miss any true patterns in the presence of different spatial relationships such as auto-correlation, inhibition, or mixed spatial interaction.

Inhibition

Here we show that a set of negatively associated features can be wrongly reported as a prevalent co-location pattern by the existing co-location mining algorithms, using typical threshold values. We also show that our algorithm does not report such a pattern as a co-location pattern, but rather reports it as a segregation pattern.

Model to generate an inhibition type interaction: Points exhibiting pairwise interaction among themselves are modeled in the spatial point process. For a *pairwise interaction model* [6], the probability density of a point process in a bounded area W is a function $f(x)$ defined for each finite configuration $x = \{x_1, \ldots, x_n\}$ of points $x_i \in W$ for any $n \geq 0$. $f(x)$ is defined in the following form $f(x) = \alpha\Big[\prod_{i=1}^{n} b(x_i)\Big] \times \Big[\prod_{i<j} c(x_i, x_j)\Big]$

where α is a normalizing constant, $b(x_i), x_i \in W$ is the intensity (first order term), and $c(x_i, x_j), x_i, x_j \in W$ is the pair-wise interaction term (second order term) [7]. For inhibition, the second order term $0 \leq c \leq 1$.

Experimental setup 1: We generate a data set with 40 instances of each of two features $\circ$, and $\triangle$ that inhibit each other. A pair-wise inhibition type can be modeled by a *Strauss process* [37], which has three parameters ($\beta, \gamma,$ and r). The probability density $f(x)$ of a Strauss process x is $\alpha\beta^{n(x)}\gamma^{s(x)}$ [6], where α is a normalizing constant, $n(x)$ is the total number of points, and β is the contributing factor of each point to the density. $s(x)$ is the number of pairs in x which are closer than r units from each other, r is the interaction distance, and γ controls the strength of the interaction between points. A Strauss process is defined for parameters $0 \leq \gamma \leq 1$, $\beta > 0$, and $r > 0$. When $\gamma = 1$, the overall density becomes the density of a Poisson process ($f(x) = \alpha\beta^{n(x)}$). With $\gamma > 1$, the point process exhibits clustering, with $\gamma = 0$, points exhibit no interaction within a distance r, termed as a *hardcore process* [37], and with $0 < \gamma < 1$, two points that are closer than r units from each other exhibit a soft inhibition or negative association. In the Strauss process, any pair of points that are lying more than r units apart does not exhibit any interdependency and the interaction term γ for such a pair equals 1. In a *multi-type* Strauss process [37], interaction has to be defined for a pair of points of similar types and for a pair of points of different types. Our data is generated from a *multi-type* Strauss process where the interaction parameter (γ) among similar type of feature instances ($\gamma_{\circ,\circ}$ and $\gamma_{\triangle,\triangle}$) is 0.43, the interaction term among different types of feature instances ($\gamma_{\circ,\triangle}$) is 0.4, and the interaction radius (r) is set to 0.1. β is 2. The study area is a unit square and the interaction distance (R_d) is 0.1. Even when imposing a soft inhibition between $\circ$ and $\triangle$, we still see some instances of pattern $\{\circ, \triangle\}$ within the interaction distance of 0.1. Fig. 4.9 shows the data set.

Result: The actual $PI_{\text{obs}}(\{\circ, \triangle\})$-value is 0.55. The p_{pos}-value of $PI_{\text{obs}} = 0.55$ according to equation 3.1 is $\frac{99+1}{99+1} = 1$, which means that seeing a PI-value of at least 0.55 under the null model is quite certain. Hence our method will not report $\{\circ, \triangle\}$ as a significant co-location pattern. Our grid based sampling approach also does not report $\{\circ, \triangle\}$ as a co-location pattern as the p_{pos}-value is always greater than $\alpha = 0.05$. The p_{neg}-value of $PI_{\text{obs}} = 0.55$ according to equation (3.2) is $\frac{1}{99+1} = 0.01 < \alpha$ which means that under the null model the probability of seeing a PI-value of 0.55 or less is quite unlikely. In our sampling approach, we find that the p_{neg}-value is always less than $\alpha = 0.05$. Hence

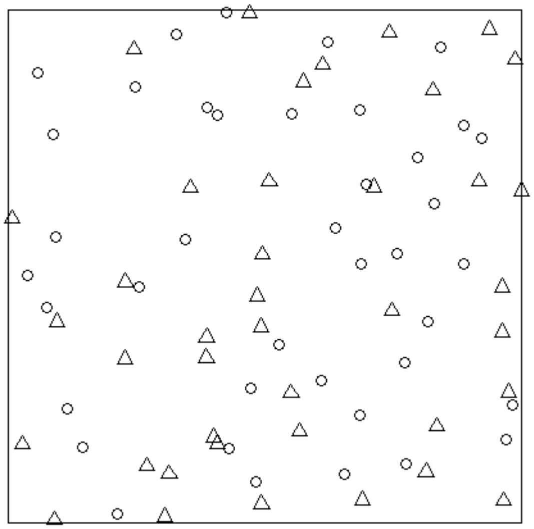


Figure 4.9: A data set where $\circ$ and $\triangle$ are negatively associated.

$\{\circ, \triangle\}$ is reported as a segregation pattern. The complete results are shown in Table A.1 of Appendix A. The reported segregation relationship can be validated by the estimation of Ripley's cross-K function. In Fig. 4.10, we see that the estimation of $K_{\circ,\triangle}(r)$ using Ripley's isotropic edge correction (solid line) is always below the theoretical curve (dashed line), which means that the average number of $\triangle$ found in a neighborhood of radius r of a $\circ$ is always less than the expected value (πr^2) indicating a negative association. The precision and recall our methods are both equal to 1, while the standard method should not report segregation patterns. However, it reports $\{\circ, \triangle\}$ as a prevalent *co-location* if a rather typical value of 0.55 or less is set as the PI threshold, which is highly misleading.

Experimental setup 2: Geyer extended the Strauss process to model an inhibition among a group of 3 close points (called a *triplet*), each pair of which is located closer than r units from each other. This inhibition model is known as *Geyers triplet process* [25]. Its probability density function $f(x)$ is similar to that of the Strauss process except in the interaction term $\gamma^{s(x)}$, $s(x)$ is defined as the number of unordered triples of points that are located closer than r units from each other. For inhibition, the model requires $0 < \gamma < 1$. Using *Geyers triplet process* [25], an extension of the Strauss process, we generate an inhibition data set (Fig. in 4.11) of 3 features $\circ$, $\triangle$, and $+$. Each feature has 50 instances and an inhibition relationship is imposed among all 3 features. The study area is a unit square and the interaction distance (R_d) is set to 0.1. Using the Metropolis-Hastings algorithm [37], first we generate a realization of the triplet process with $\beta = 2$, $\gamma = 0.45$, and $r = 0.1$. The realization is a data set of 150 un-marked points. Then we randomly pick

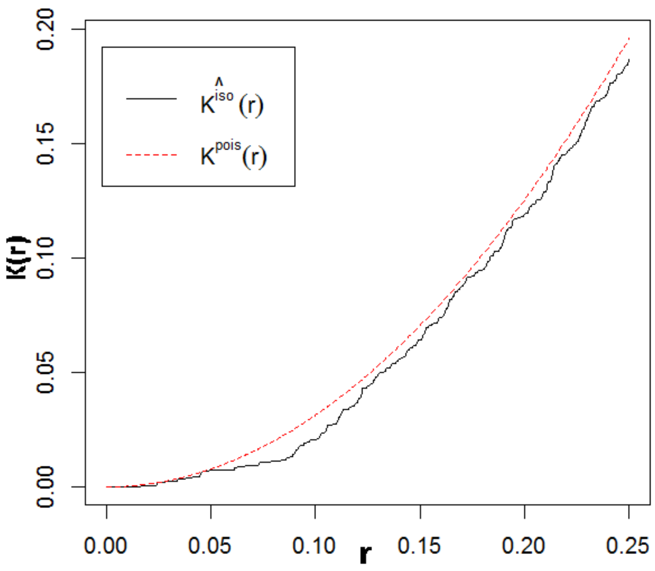


Figure 4.10: Inhibition: estimation of Ripley's K- function $K_{\circ,\triangle}(r)$.

50 un-marked points and set their marks to $\circ$. Similarly another 50 un-marked points are set to mark $\triangle$ and the rest are set to mark $+$. Note that even when imposing an inhibition relationship, we can still see some instances of pattern $\{\circ, \triangle, +\}$ (as $\gamma \neq 0$). The data set is shown in Fig. 4.11.

Result: The $PI_{\text{obs}}(\{\circ, \triangle, +\})$-value is 0.42. The p_{pos}-value is 0.99, which is greater than $\alpha = 0.05$, and hence our method does not report $\{\circ, \triangle, +\}$ as a significant co-location pattern. Our grid based sampling approach also does not report $\{\circ, \triangle, +\}$ as a co-location pattern as the p_{pos}-value is always greater than $\alpha = 0.05$. The p_{neg}-value is 0.03. The p_{neg}-values using our sampling approach are also smaller than $\alpha = 0.05$. Hence in all our approaches, $\{\circ, \triangle, +\}$ is reported as a segregation pattern. The complete results are shown in Table A.2 of Appendix A. The reported segregation relationship can also be validated by estimating the third order summary statistics $T(r)$ [58]. In Fig. 4.12, we see that the estimation of $T_{\circ,\triangle,+}(r)$ with border correction (solid line) is always below the theoretical curve (dashed line), which means that in an r-neighborhood of a typical point o, the average number of r-close triples including o is always smaller than the expected value, indicating a segregation among features $\circ$, $\triangle$, and $+$. Again, the precision and recall of all our methods are 1, while the standard method should not report the segregation pattern. However, they will wrongly report the pattern $\{\circ, \triangle, +\}$ as a prevalent *co-location* if a value of 0.42 or less as the PI threshold is used, which again, is not uncommon.

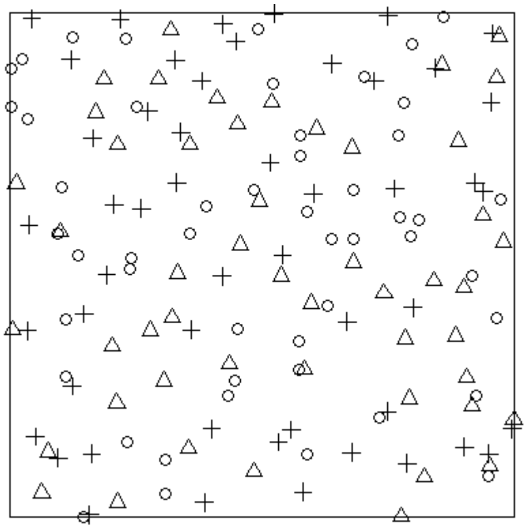


Figure 4.11: A data set with an inhibition relationship between $\circ$, $\triangle$, and $+$.

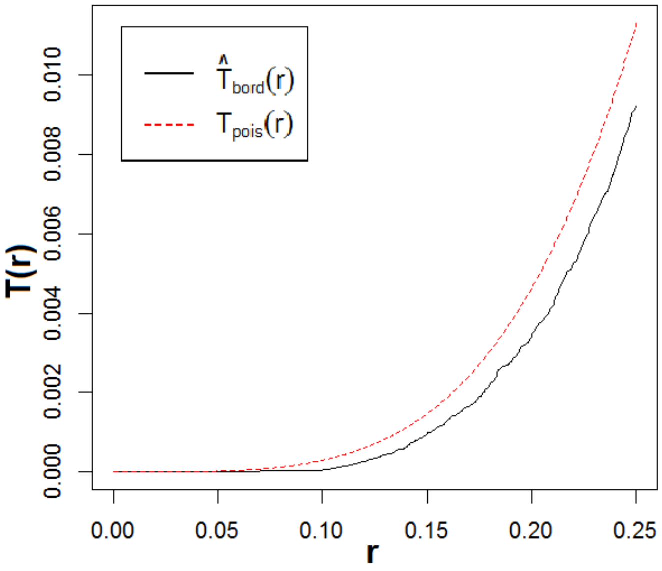


Figure 4.12: Inhibition: estimation of the 3rd order summary statistics $T(r)$.

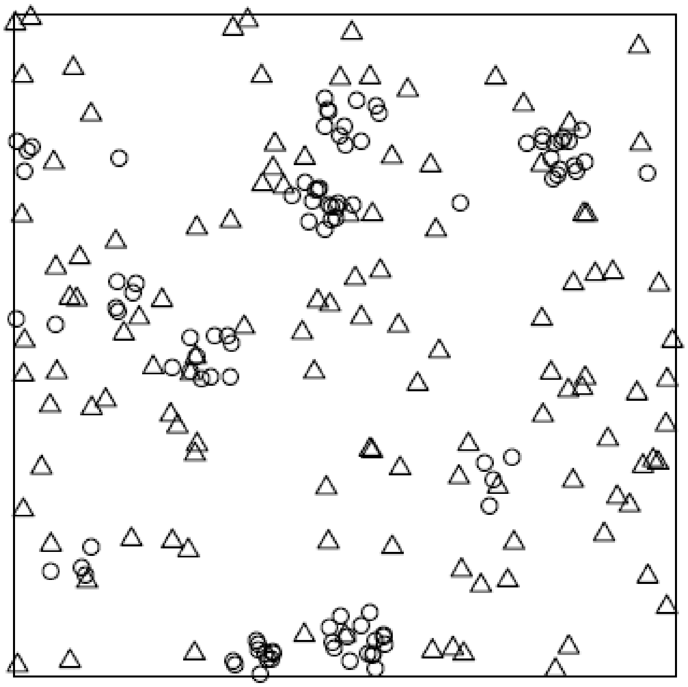


Figure 4.13: A data set where $\circ$ is auto-correlated and $\triangle$ is randomly distributed.

Auto-correlation

In this experiment, we show that even though participating features of a pattern are independent of each other, their spatial auto-correlation properties can generate a PI-value higher than a typical threshold. Our algorithms do not report such patterns as a true co-locations.

Experimental setup: We generate a synthetic data set (shown in Fig. 4.13) with 2 different features $\circ$, and $\triangle$. Feature $\triangle$ has 120 instances which are independently and uniformly distributed. Feature $\circ$ has 100 instances which are spatially auto-correlated. The spatial distribution of $\circ$ follows the model of Matérn's cluster process [37]. The study area is a unit square and the spatial interaction neighborhood radius (R_d) is 0.1.The summary statistics of $\circ$ are $\kappa = 40$, $\mu = 5$, and $r = 0.05$.

Result: The $PI_{\text{obs}}(\{\circ, \triangle\})$-value is 0.46. The p_{pos}-value is 0.75 and the p_{neg}-value is 0.31 which are greater than α, and hence pattern $\{\circ, \triangle\}$ is not reported as a co-location or segregation pattern. Our grid based sampling approach also does not report $\{\circ, \triangle\}$ as a co-location or a segregation pattern. Table A.3 of Appendix A shows the complete results from our different approaches. The standard co-location approaches, on the other hand, will mistakenly report the pattern $\{\circ, \triangle\}$ as prevalent since its PI-value of 0.46 is higher than typical thresholds.

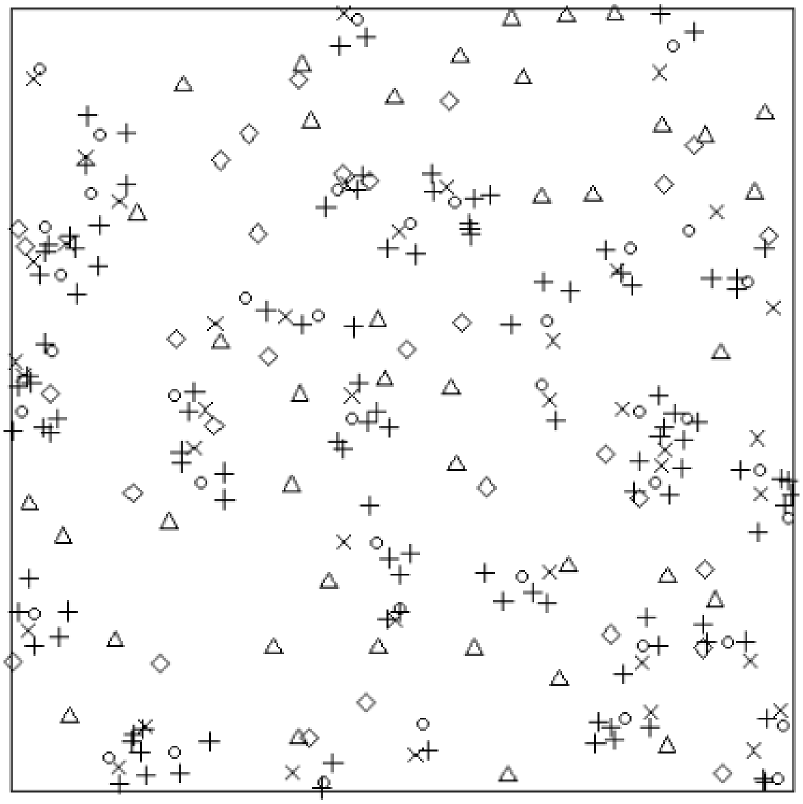


Figure 4.14: A data set with 5 features.

Mixed Spatial Interaction

Here, we generated a synthetic data set with 5 different feature types $\circ$, $\triangle$, $+$, $\times$, and $\Diamond$ (Fig. 4.14). Among these features, we impose different spatial relationships such as positive association, auto-correlation, inhibition, and randomness. We show that our algorithms are able to detect co-location and segregation patterns occurring due to positive and negative associations, and that we do not report "false" patterns even if they may have high PI-values.

Experimental setup: Features $\circ$, $\triangle$ and $\times$ have 40 instances each. Feature $+$ has 118 instances, and feature $\Diamond$ has 30 instances. Our study area is a unit square and the interaction neighborhood radius (R_d) is set to 0.1. Features $\circ$ and $\triangle$ have a negative association and instances of these two types are generated from an inhibition process (a Multi-Strauss hardcore process [37], with parameter $\beta = 300$ for both features), where no two feature instances (either the same feature types or different feature types) are seen within a predefined distance threshold (called hardcore distance h, here $h = 0.05$); and an inhibition (negative association) is present at a distance $0.05 < r < 0.1$ where the inhibition parameter γ is 0.3 between $\circ$ and $\triangle$, and 0.43 between feature instances of the same type. Feature $\circ$ and feature $\times$ are positively associated, so that an instance of feature $\times$ will be found within the R_d distance of an instance of feature $\circ$. Feature $+$ is spatially auto-correlated, modelled as a Matérn's cluster process (with parameter $\kappa = 40$, $\mu = 3$, and $r = 0.05$) and positively associated with feature $\circ$ and feature $\times$. Hence, we observe a group of $+$ around each instance of $\circ$ and $\times$. Feature $\Diamond$ is randomly distributed. Table 4.1 shows the spatial relationship that are implanted in the synthetic data.

Table 4.1: Relationships implanted in the synthetic data

Type	Relationships among the features
1	Positive associations: $\{\circ, +\}$, $\{\circ, \times\}$, $\{+, \times\}$, $\{\circ, +, \times\}$
2	Negative associations: $\{\circ, \triangle\}$, $\{\triangle, +\}$, $\{\triangle, \times\}$
3	$\Diamond$ is independent of rest of the features.

Result: Our algorithms detects patterns generated due the spatial associations that are implanted in the synthetic data. Table A.4, A.5, and A.6 of Appendix A show the complete results for the computed PI-values, the PI^*-values for different grid sizes, the p_{pos}-values, and the p_{neg}-values of all possible subsets. The results for each possible subset of features are discussed in detail, by increasing pattern size.

Size-2 subsets (Table A.4): $\{\circ, \triangle\}$ is not reported as a significant co-location pattern ($p_{\text{pos}} > 0.05$) due to their inhibitive interaction. Rather it is reported as a significant segregation pattern ($p_{\text{neg}} < 0.05$). Feature $\circ$, feature $+$, and feature $\times$ are strongly associated in the synthetic data and that is captured in the result: $\{\circ, +\}$, $\{\circ, \times\}$, and $\{+, \times\}$ all have a p_{pos}-value of 0.01 and are thus reported as significant co-location patterns. $\{\circ, \Diamond\}$ is not reported which is correct since both features are independent of each other. The same applies to $\{\triangle, \Diamond\}$, $\{+, \Diamond\}$, and $\{\times, \Diamond\}$. Since an inhibition relationship exists between feature $\circ$ and feature $\triangle$ and a positive association exists among $\circ$, $+$, and $\times$; $\triangle$ also shows an inhibition relationship with $+$ and $\times$. In our result $\{\triangle, +\}$, and $\{\triangle, \times\}$ are, accordingly, not reported as significant co-location patterns, but they are reported as significant segregation patterns. Note that some patterns (such as $\{\circ, \Diamond\}$, $\{\triangle, +\}$, $\{\times, \Diamond\}$) which are not significant co-location patterns will be reported by existing algorithms when using typical thresholds (such as 0.55), since their actual PI-values are all higher than 0.55.

Size-3 subsets (Table A.5): $\{\circ, \triangle, +\}$, $\{\circ, \triangle, \times\}$, and $\{\circ, \triangle, \Diamond\}$ are not reported due the inhibition of $\circ$ and $\triangle$. $\{\circ, \triangle, +\}$, $\{\circ, \triangle, \times\}$ are not even reported as a statistically significant segregation pattern due to the positive association of $\{\circ, +\}$, $\{\circ, \times\}$ respectively.

$\{\circ, \triangle, \diamondsuit\}$ is also not reported as segregation due to the independence of $\diamondsuit$ from $\circ$ and $\triangle$. $\{\circ, +, \times\}$ is reported as significant co-location which is correct due to the first type of relationship of Table 4.1. $\{\triangle, +, \times\}$ can not be a true co-location due the second type of relationship of Table 4.1. This subset is also not reported as significant in our result. This also can not be a true inhibition due the strong association relationship between feature $+$ and feature $\times$. Feature $\diamondsuit$ together with a strongly associated pair of features (such as feature $\circ$, and feature $+$) could also appear as a positive association which is the case in our result where we find $\{\circ, +, \diamondsuit\}$, $\{\circ, \times, \diamondsuit\}$, and $\{+, \times, \diamondsuit\}$ as significant.

Size-4 subsets (Table A.6): among all subsets of size 4, only $\{\circ, +, \times, \diamondsuit\}$ is reported as significant co-location pattern due to the positive association among $\circ$, $+$, and $\times$. To report this pattern (actual $PI = 0.55$) using the existing algorithms, the global threshold value would have to be set to at least 0.55, which, however, would also result in reporting non-significant patterns (e.g., $\{\triangle, \diamondsuit\}$, with actual $PI = 0.575$). $\{\circ, \triangle, +, \times\}$, $\{\circ, \triangle, +, \diamondsuit\}$, and $\{\circ, \triangle, \times, \diamondsuit\}$ are not reported. This is due to the fact that two negatively associated features $\circ$ and $\triangle$ are present in those 4-size subsets. They are not even reported as true segregation patterns. This is due to the fact that relationship types 1 (inhibition) and 2 (association) of Table 4.1) are both present in these 4-size subsets. similarly, $\{\triangle, +, \times, \diamondsuit\}$ is also not reported.

Size-5 subsets (Table A.6): the only subset of size 5, $\{\circ, \triangle, +, \times, \diamondsuit\}$ is not reported. The combined effect of relationship type 1, 2 and 3 of Table 4.1 does not result in either a true positive association or a true negative association among these 5 features.

As can be seen in Table A.4, A.5, and A.6, the sampling based approach does not miss any co-location or segregation pattern of size 2, 3, and 4.

The SSCSP algorithm reports all 4 implanted co-location and all 3 segregation patterns. It also reports the 4 additional co-location patterns $\{\circ, +, \diamondsuit\}$, $\{\circ, \times, \diamondsuit\}$, $\{+, \times, \diamondsuit\}$, and $\{\circ, +, \times, \diamondsuit\}$, which correspond to the 4 implanted patterns but include the additional, independently distributed feature $\diamondsuit$. Due to the positive association among $\circ$, $+$, and $\times$, the amount of pattern instances found in the observation is significantly higher than the amount found under the independence assumption. These additional four patterns are in a sense redundant patterns since they reflect the implanted, strong association between three of the involved features. Such patterns can, in principle, be pruned by using an independence assumption, conditioned on already found sub-patterns. However this is beyond the current scope of this chapter. But in the following chapter, we propose a solution to remove such type of redundant pattern by using a null hypothesis conditioned on a constraint set. As can be seen in Table A.4, A.5, and A.6: the sampling based approach find exactly the same patterns as the all-instances-based approach. The result of a standard co-location algorithm depends on the chosen threshold. For this experiment we report the performance for three different thresholds: 0.2, 0.4, and 0.5. Table 4.2 shows precision, recall, and F-measure (harmonic mean of precision and recall), for the standard co-location algorithm using these thresholds, as well as for our method.

Table 4.2: Existing methods vs. our method

PI	A standard co-location approach			Our
threshold →	0.2	0.4	0.5	method
Precision	0.15	0.21	0.27	0.64
Recall	1	1	1	1
F-measure	0.26	0.35	0.43	0.78

For instance, if the PI threshold for the standard co-location mining algorithm is set to 0.4, 19 patterns will be reported as prevalent; among those, 4 patterns are true co-locations, 4 patterns are the same redundant patterns that our algorithm reports as well, 3 patterns are the wrongly reported *segregation* patterns and the rest are meaningless patterns. In this case, precision, recall, and F-measure, not counting the segregation patterns (since the co-location mining algorithm should, according to its semantics, not find those) are $\frac{4}{19} = 0.21$, $\frac{4}{4} = 1$, and 0.35, respectively.[1]

Runtime Comparison

For an auto-correlated feature, we do not generate all of its instances and we can also prune candidate patterns which can not contribute to the p-value computation under certain circumstances (see Sect. 4.1.2). In a naïve approach, we do not apply any of these techniques.

All experiments are conducted on an Intel Core i3 processor machine with a cpu speed of 2.10 Ghz. The main memory size is 2 GB and the OS is Windows 7. For runtime comparison, we generate a data set with 4 different features $\circ$, $\triangle$, $+$, and $\times$. Features $\circ, \triangle$, and $+$ are auto-correlated features. They also show an inhibition relationship with feature $\times$. The

[1] If we are "generous" and count the wrongly reported segregation patterns as correct, the precision, recall, and F-measure would be $\frac{7}{19} = 0.37$, $\frac{7}{7} = 1$, and 0.54, respectively, which is still substantially worse than our method.

study area is a square with an area of 100 sq. units and the interaction distance R_d is set to 0.1. We impose a positive association among $\circ$, $\triangle$, and $+$ where each instance of a feature is in co-location with an instance of the other two types of features. Feature $\circ$, feature $\triangle$, and feature $+$ have 400 instances each and feature $\times$ has 20 instances. The naïve approach, the All-instances-based SSCSP, as well as all the sampling approaches find the same significant co-location patterns ($\{\circ, \triangle\}$, $\{\circ, +\}$, $\{\triangle, +\}$, and $\{\circ, \triangle, +\}$) and the same significant segregation patterns ($\{\circ, \times\}$, $\{\triangle, \times\}$, and $\{+, \times\}$). We conduct four more, similar experiments, and in each experiment we keep the total cluster number of each auto-correlated feature the same but increased the total number of instances per cluster by a factor k for all clusters. For all these experiments the same co-location and segregation patterns are reported as significant by all different approaches. Figure 4.15 shows the runtime of a naïve approach, the all-instances-based SSCSP algorithm, and the grid based sampling approach with 4 different cell resolutions. Figure 4.16 shows that with the increase of the number of instances, we obtain an increasing speedup growing from 1.9 to 5.31 for the All-instances-based SSCSP algorithm. We obtain further increasing speedup using grid based sampling. With increasing number of feature instances, the speedup increases from 4.7 to 12.8 when $l = R_d$, from 4 to 11.9 when $l = \frac{R_d}{2}$, from 3.12 to 10.9 when $l = \frac{R_d}{3}$, and from 2.3 to 9.3 when $l = \frac{R_d}{4}$. A cell resolution of $l = R_d$ gives the best speedup but may not be a safe choice for mining a true co-location or segregation which has very few instances. Our experiments overall suggest that a cell resolution of either $l = \frac{R_d}{2}$ or $l = \frac{R_d}{3}$ is a good choice since in all conducted experiments with synthetic and real data sets (to be discussed next), it only missed one true co-location pattern in a case very one of the involved features has a very low number of instances.

For an auto-correlated feature, if the number of clusters increases, the chance of a cluster being close to other features will be higher. Hence the data generation step might have to generate more instances of each auto-correlated feature in such cases. In another 5 experiments, we increased the number of instances of feature $\times$ and the number of clusters of each auto-correlated feature ($\circ$, $\triangle$, and $+$) by the same factor but keep the number of instances per cluster the same. Fig. 4.17 shows the runtime and Fig. 4.18 shows the speedup obtained by the different approaches in the 5 experiments. We see that with increasing number of clusters, after increasing first, the speedup eventually goes down. This happens when more and more instances actually have to be generated, eventually leaving only the speedup due to candidate pruning.

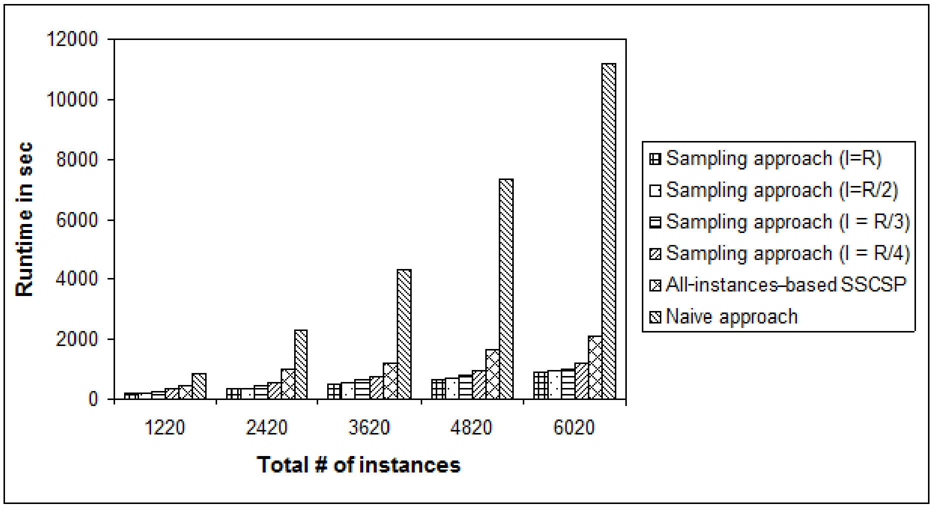


Figure 4.15: Runtime comparison.

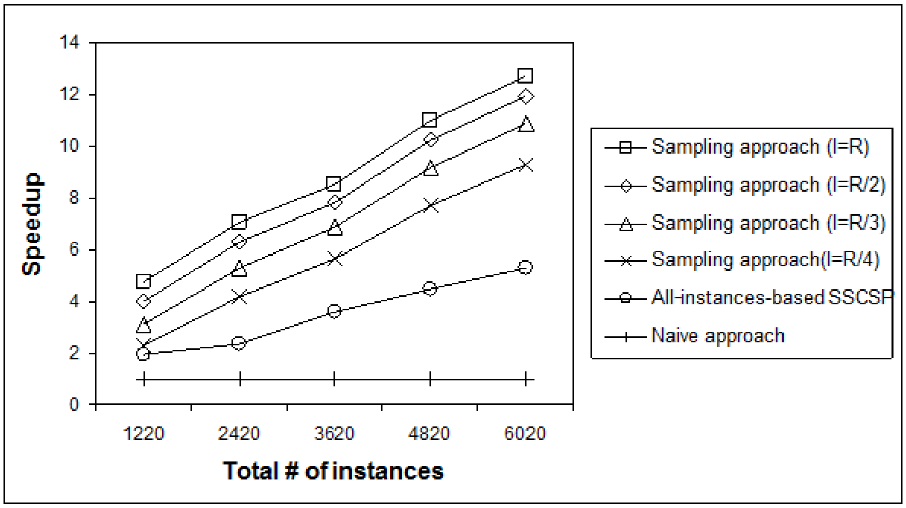


Figure 4.16: Speedup.

These experiments also demonstrate that the runtimes of our approaches are acceptable for many real world application where the quality of the results matters more than speed.

Comparison with the existing algorithms: The computational time of the existing algorithms depends on the selection of the PI_{thre}-value. A low PI_{thre}-value is computationally more expensive than a high PI_{thre}-value. A low PI_{thre}-value allows fewer pruning and thus results in more candidate patterns as being prevalent. Hence there is no fair way to compare our algorithm with the existing algorithms. To show a range of possible results, we use 0.2 and 0.5 as PI_{thre}-value to measure the computational time of the algorithm in [72] and compare it with our algorithm (shown in Table A.7 of Appendix A) using the data (Fig. 4.14) of

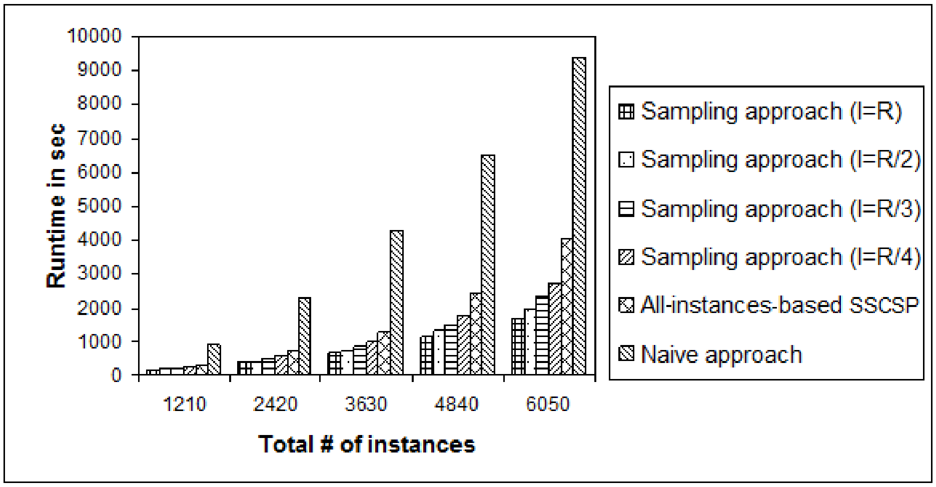


Figure 4.17: Runtime comparison.

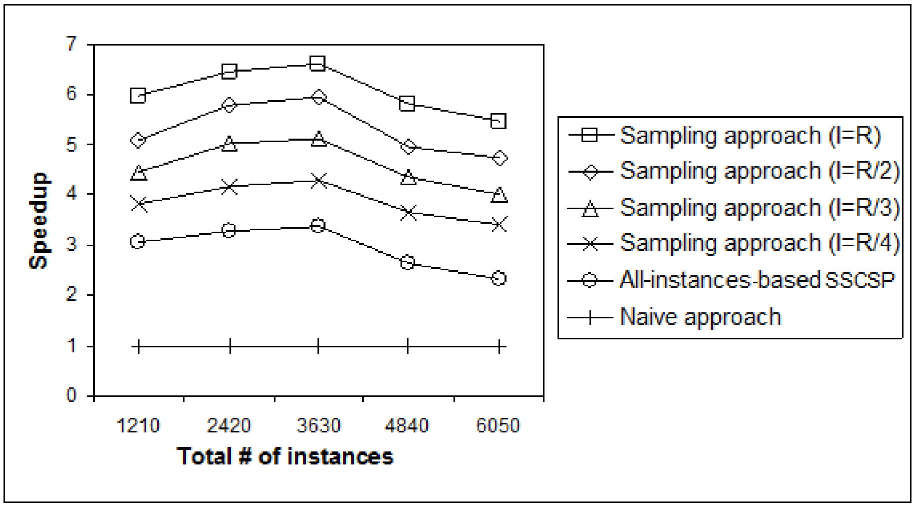


Figure 4.18: Speedup.

the mixed spatial interaction experiment of section 4.2.1. Due to the randomization tests required for the significance tests, our algorithm is clearly slower than the existing algorithm. However with $PI_{\text{thre}} = 0.2$, the join-less algorithm reports all subsets as prevalent, which is meaningless, and when when $PI_{\text{thre}} = 0.5$ the algorithm reports the four true co-location patterns, but also eleven additional patterns, five of which are meaningless, four of which are redundant patterns, two of which are in fact segregation patterns and not co-location patters, which is a particularly severe mistake.

4.2.2 Real Data Sets

We conduct experiments with five real data sets to validate our proposed approaches. Some of these data sets are also used by ecologists. We compare our findings with their results.

Ants Data

The nesting behavior of two species of ants (*Cataglyphis bicolor* and *Messor wasman*) is investigated to check if they have any dependency on biological grounds. The Messor ants live on seeds while the *Cataglyphis* ants collect dead insects for foods which are for the most part dead *Messor* ants. *Zodarium frenatum*, a hunting spider, kills *Messor* ants. The question is if there is any possible connection we can determine between these two ant species based on their nest locations. The full data set gives the spatial locations of nests recorded by Professor R.D. Harkness [31]. It comprises 97 nests (68 *Messor* and 29 *Cataglyphis*) inside an irregular convex polygon (Fig. 4.19).

We run our algorithm on the ants data and compute the PI-value based on all instances and based on the grid based sampling approach. Each of the 24 *Cataglyphis* ant nests is close to at least one *Messor* ant's nest, not more than 50 unit away, and the participation ratio of *Cataglyphis* ant is $\frac{24}{29} = 0.83$. For *Messor* ants, the participation ratio is $\frac{30}{68} = 0.44$. Thus the actual PI_{obs}-value of interaction pattern $\{$*Cataglyphis* ,*Messor*$\}$ is 0.44. In the randomization test, we generate 99 simulation runs and find that in 18 simulation runs, the $PI_0^{R_i}$-value is greater than or equal to the PI_{obs}-value. The p_{pos}-value is equal to $\frac{18+1}{99+1} = 0.19$, which is greater than 0.05 and thus not statistically significant. Hence we can not conclude that there is a positive dependency between these two types of ants. The p_{neg}-value is calculated as $\frac{81+1}{99+1} = 0.82$, which is greater than 0.05 and thus the interaction pattern can not be a statistically significant segregation pattern. Table B.1 in the Appendix B shows the computed PI-values, p_{pos}-values, and p_{neg}-values using the grid based sampling method with different cell resolutions. Again, for all grid sizes, the result is the same, i.e. the spatial interaction of *Cataglyphis* and *Messor* is neither a statistically significant co-location nor a statistically significant segregation. In fact, clear evidence of a spatial association between these two species is also not found in [31]. Existing co-location mining algorithms would report $\{$*Cataglyphis* ,*Messor*$\}$ as a prevalent co-location if a value of 0.44 or less is set as PI threshold.

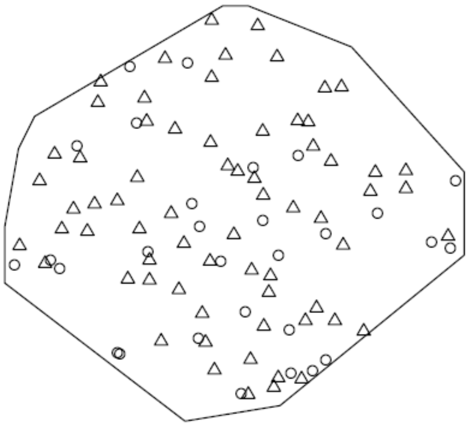
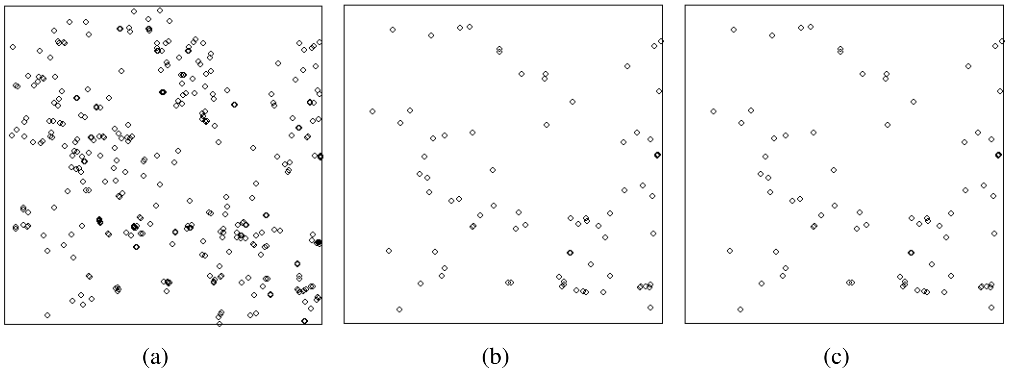


Figure 4.19: Ants data: ○ = Cataglyphis ants and △ = Messor ants. [31]

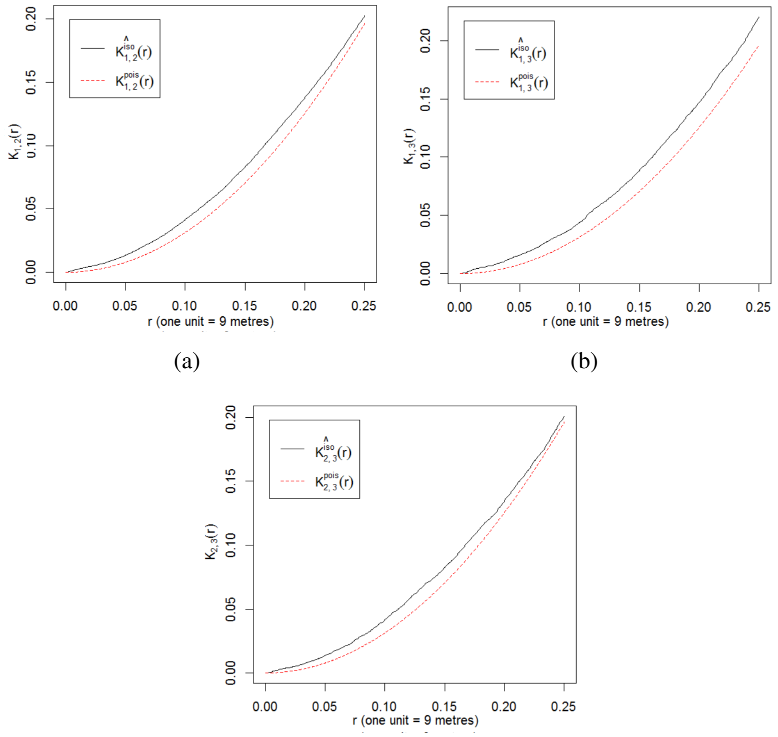
Bramble Canes Data

Hutchings recorded and analyzed the cane distribution of *Rubus fruticosus* (blackberry). The blackberry bush is known as *Bramble*. Bramble canes data (published in [20]) records the locations (x, y) and ages of bramble canes in a field of a 9m square plot. The canes were classified according to age as either winter buds breaking the soil surface, un-branched and non-flowering first year stems, or branched and flower bearing second year stems [36]. These three classes are encoded as marks 1, 2, and 3 respectively in the data set. There are 359 canes with mark 1, 385 with mark 2, and 79 with mark 3. Hutchings' investigation finds an aggregated pattern in all cohorts of canes [36]. This indicates the presence of auto-correlation for each mark. Diggle also analyses the bivariate pattern formed by canes with mark 1 and 2 and finds a positive dependency between these two types [20] (Section 6.3.2).



(a) (b) (c)

Figure 4.20: Bramble canes data: distribution of a) newly emergent (mark 1), b) 1 year old (mark 2), and c) 2 years old (mark 3) canes [20].



(a)

(b)

(c)

Figure 4.21: Bramble canes data: Ripley's a) $K_{1,2}$, b) $K_{1,3}$, and c) $K_{2,3}$ curves.

For our experiments, we re-scaled the 9 m square plot area to the unit square and set the co-location radius to 0.1. Using the location information of the Bramble canes from [20] the spatial distribution of each type of cane can be plotted, which is shown in Fig. 4.20. The PI-values, p_{pos}-values, and p_{neg}-values are shown in Table B.2 of the Appendix B. In the result, all possible subsets are reported as significant co-location patterns. This also conforms with Diggle's investigation where a pair-wise positive dependency among different types of canes is also reported. The aggregation tendency of the three types of canes that is reported (as $\{1, 2, 3\}$) in our approach can also be predicted from the estimated Ripleys cross-K function curves (Fig. 4.21) of all possible pairs. In all 3 cross-K function curves of Fig. 4.21, we see that the estimated K-value from the data at the co-location distance ($R_d = 0.1$) is always greater than the theoretical K-value (estimated from a Poisson distribution) indicating an pairwise aggregation tendency. A positive association among all pairs and similar spatial distribution of each type of cane results in a positive association among all three types of canes.

Lansing Woods Data

D.J. Gerrard prepared this famous multi-type point data set from a plot of 19.6 acre in Lansing Woods, Clinton County, Michigan, USA. This data set records the location of 2251 trees of 6 different species (135 black oaks, 703 hickories, 514 maples, 105 red oaks, 346 white oaks, and 448 miscellaneous trees) [24]. For our experiments, we set the interaction distance to 92.4 feet and re-scaled the original plot size (924×924 feet) to the unit square in order to mine significant positive and negative interactions. The individual distribution of each tree species is shown in Fig. 4.22. We estimate the pair-wise correlation function ($g(d)$) value for each tree species. At distance $d = 92.4$ feet, we find each tree species as spatially auto-correlated as $g(d) > 1$.

In the Appendix B, Table B.3 shows the PI-values and p_{pos}-values of the significant co-location patterns found by our algorithms. In the Appendix B, Table B.4 shows the PI-values and p_{neg}-values of the segregation patterns found by our algorithms. Diggle [20] and Perry et al. [49] analyze the Lansing Woods data to find bivariate patterns only. Some of the 2-size patterns that are reported in our method are also found in their work. In their result, hickory and maple are reported to deviate from randomness and exhibit segregation. Our findings can also be validated by estimating Ripley's cross-K function at the interaction distance 92.4 feet for all pairs. From the estimated Ripleys cross-K function values, we find the following pair-wise spatial relationships (Table 4.3):

Table 4.3: Pari-wise spatial association: Auto: auto-correlation, Ind: independency, +: positive, -: negative.

	Black oak	Hickory	Maple	Misc.	Red oak	White oak
Black oak	Auto	+	-	-	Ind	Ind
Hickory		Auto	-	-	Ind	Ind
Maple			Auto	+	Ind	Ind
Misc.				Auto	Ind	Ind
Red oak					Auto	Ind
White oak						Auto

To report co-locations of Table B.3 using the existing co-location algorithms, the PI threshold can not be greater than 0.702. Such a threshold would, however, also select interaction patterns $\{\textit{Black oak}, \textit{Maple}\}$ and $\{\textit{Hickory}, \textit{Maple}\}$ of Table B.4 as co-location patterns, which are actually segregation patterns.

In the Appendix B, Table B.5 shows some of the interaction patterns that have high PI-

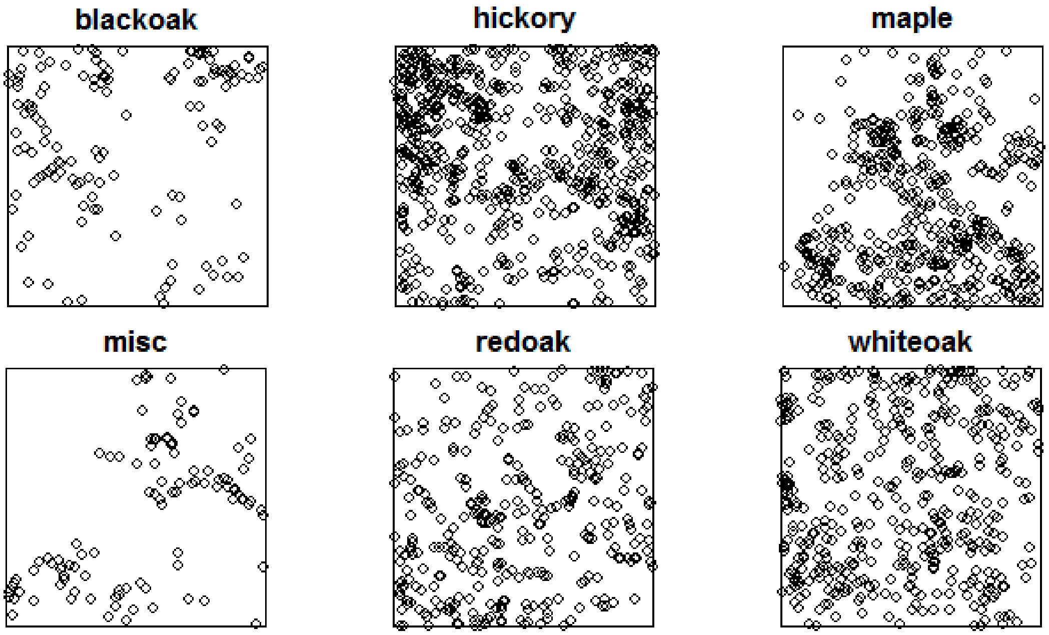


Figure 4.22: Spatial distribution of each tree species of Lansing woods data [7].

values (actual and approximated) but not reported as significant by our method. Existing co-location mining algorithms will report them as prevalent co-location patterns as their actual PI-values are 1. Ripley's K-function values for $\{$*Hickory*, *Red oak*$\}$, $\{$*Hickory*, *White oak*$\}$, and $\{$*Red oak*, *White oak*$\}$ at distance 92.4 feet also indicate pair-wise independence among the participating features involved in these patterns. The investigation of Diggle and Perry et al. and the result of cross-K function is used as the ground truth. These approaches analyze only bivariate spatial relationship. Using their findings as the ground truth we compute the precision, recall, and F-measure of our method for patterns of size 2 only. In terms of mining strategy, existing co-location algorithms follow the same standard approach of using a PI threshold. These methods only vary in terms of performance efficiency by adopting different techniques for identifying instances of candidate patterns. We also compute the precision, recall, and F-measure of a standard co-location mining approach using patterns of size 2 and compare our method with a standard co-location mining approach in terms of detection accuracy. Table 4.4 shows the precision, recall, and F-measure of our method and a standard co-location approach for 3 different PI threshold values, 0.2, 0.4, and 0.5, for patterns of size 2. There are 2 true co-location patterns of size 2 and 4 true segregation patterns of size 2. For instance, with a PI threshold value of 0.4, a standard co-location approach reports 15 pairs as prevalent co-locations. All the true co-location patterns are included among those reported patterns. Counting segregation patterns as mistake, the pre-

Table 4.4: Existing methods vs. our method

PI	A standard co-location approach			Our
threshold →	0.2	0.4	0.5	method
Precision	0.13	0.13	0.14	1
Recall	1	1	1	1
F-measure	0.23	0.23	0.25	1

cision, recall, and F-measure are $\frac{2}{15} = 0.13$, $\frac{2}{2} = 1$, and 0.23, respectively.[2]

Toronto Address Repository Data

The Toronto Open Data provides a data set with over 500000 addresses within the City of Toronto enclosed in a polygonal area. Each address point has a series of attributes including a feature class with 65 features and coordinates. After removing entries with missing data and removing features with very high frequency (e.g. high density residential), we consider 10 features for our experiment: low density residential (66 instances), nursing home (31 instances), public primary school (510 instances), separate primary school (166 instances), college (32 instances), university (91 instances), fire station (63 instances), police station (19 instances), other emergency service (21 instances), and fire/ambulance station (16 instances). Due to space limitations, only some of the feature distributions are shown in Fig. 4.23. To determine if a feature shows clustering (spatial auto-correlation), regularity (inhibition), or randomness (Poisson), we compute the *pair correlation function* $g(d)$ [37]. Police stations, fire stations, fire/ambulance stations, and separate primary schools show regular distributions, since $g(d) < 1$ at smaller d values. The remaining features are auto-correlated since their $g(d) > 1$ for smaller values of d. The interaction neighborhood radius is set to 500.

In Table B.6 of Appendix B, we show statistically significant 2-size, 3-size, and 4-size co-locations and their PI_{obs}-values computed by the All-instances-based SSCSP algorithm. Note that the PI_{obs}-values are so low that existing co-location mining algorithms would return almost every feature combination as a co-location if their global threshold would be set so that the reported statistically significant co-locations can be returned. Our grid based sampling approach also finds all the statistically significant co-locations for all grid sizes,

[2] If we are "generous" again, and count the wrongly reported segregation pattern as pattern as correct, the precision, recall, and F-measure would be $\frac{6}{15} = 0.4$, $\frac{6}{6} = 1$, and 0.57, respectively, which is still substantially worse than our method.

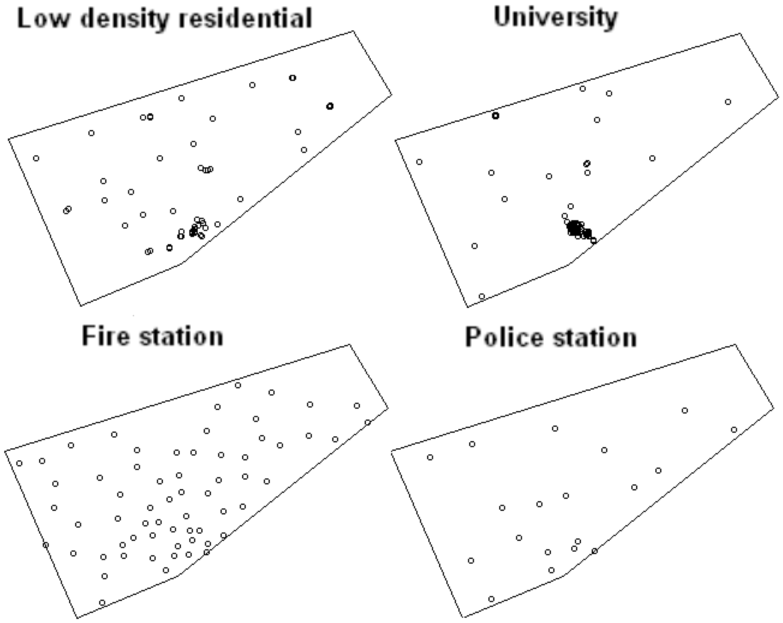


Figure 4.23: Spatial distribution of 4 features from the Toronto address data.

with the exception of co-location $\{\text{Low density resid.}, \text{Univ.}, \text{Fire station}, \text{Police station}\}$, which is missed when a grid with only $l = R_d$ (i.e. $w = 1$) is used for sampling. In the Appendix B, Table B.7 shows the actual and approximate PI_{obs}-values and p_{pos}-values of all the reported co-locations.

4.3 Summary

In this chapter, we propose a new definition of co-location and segregation patterns and a method to detect them. Existing approaches in the literature find prevalent patterns based on a predefined threshold value which can lead to missing meaningful patterns or reporting meaningless patterns. Our method uses a statistical test. Such statistical test is computationally expensive and we introduce two approaches to improve the runtime. In our first approach, we reduce the runtime by generating a reduced number of instances for an auto-correlated feature in a simulated data generation step and by pruning unnecessary candidate patterns in the PI-value computation step. In the second approach, we show that a PI-value of a pattern computed from a subset of the total instances is, in general, sufficient to test the significance of a pattern. We introduce a grid based sampling approach to identify the instances of a pattern for the significance test at a reduced computational cost. As a result, the speedup is further improved compared to our *all-instance-based* SSCSP approach.

We evaluate our algorithm using synthetic and real data sets. Our experimental results show that our sampling approach never misses any true patterns when the number of feature instances is not extremely low. However for a pattern with a very few instances of a participating feature, we recommend to use a finer grid instead of a coarser grid. Both the *all-instance-based* SSCSP and grid based sampling algorithms find all the true patterns from the synthetic data sets. Using real data sets, we show that our algorithms do not miss any pattern of size 2 found in others work found in ecology. The pattern finding approach proposed in ecology can not detect patterns of size greater than 2. We show that our methods also finds meaningful patterns of larger sizes. We find that our approaches may also report redundant patterns. A redundant pattern could occur in a statistical test due to presence of true patterns. In the next chapter, we propose a solution to prune redundant patterns from the result.

Chapter 5

Co-location Pattern Mining at Multiple Interaction Distances

In spatial domains, spatial interaction among Binary spatial features results in co-location or segregation patterns. A co-location pattern occurs due to the positive interaction of a subset of spatial features. Besides the PI-threshold parameter, existing algorithms [35, 60, 69, 71, 72] also require another parameter, the distance threshold R_d, to define the co-location neighborhood. The interaction distance of any two features participating in a co-location pattern $\mathcal{C}$ can not be greater than R_d. Existing algorithms claim to find all prevalent co-location patterns for a given R_d-value. In Chapter 3, we discuss the limitations of the existing approaches in mining true patterns for a given interaction neighborhood. Our statistical model SSCSP proposes a solution for limitations of the existing approaches. Like the existing approaches, SSCSP requires a distance threshold R_d. Then true patterns that occur at R_d are searched. However knowing the right R_d-value of each true co-location is not easy in many areas such as forestry, and ecology. In these areas spatial interactions among Binary spatial features occur at different distances resulting in co-locations of different types at different distances. The following scenario gives an example of multiple interactions occurring at different distances.

5.1 Motivation

5.1.1 A Motivating Example

In the ecosystem, living organisms at different levels of the food chain exhibit interaction among themselves. Wild boars prefer to live in open damp woodlands that offer a variety of foods. They like to live near water or muddy areas as they do not perspire. If there is no food shortage, wild boars stay in their home area all their lives and do not travel beyond a few square miles. Tigers and wild boars have a predator-prey relationship and wild boar is a favorite food source of tigers [67]. Tigers also like to live near damp areas and near vegetation to hide themselves. The tiger has a larger territory (7.7 sq. miles for females and $23-39$ sq. miles for males) than wild boars [67]. Tigers, wild boars, and damp land generate a co-location pattern. Pair-wise they also form co-locations at different spatial distances. There the co-location distance of the pattern {wild boar, damp land} is smaller than the co-location distance of the pattern {wild boar, tiger}. To identify all these co-locations, existing algorithms require each co-location distance which is sometimes difficult to predetermine. In such cases, by using a large value as the interaction distance threshold, an existing algorithm may find all of the above mentioned patterns but may also, as a consequence, report other subsets of features (from the same domain) which are not true patterns.

5.1.2 Limitations of the Current Approaches

Using one single distance threshold to capture all true patterns interacting at multiple distances is typically not feasible and could result in missing some of them. Such a case requires using more than one distance threshold. In order to determine these distance thresholds, we need to know all the interaction distances. In many application domains determining all distances is difficult due to the presence of a large variety of spatial features and their intra (between instances of a feature) and inter (between instances of features of different types) types of spatial interactions.

Even by doing repetitive trials with different distance thresholds, the existing algorithms can not fix just one single distance threshold to find all true co-locations without reporting random patterns. Let us assume that $\mathcal{C}$ is a co-location that occurs at a large distance. To capture $\mathcal{C}$ and other co-locations occurring at a distance smaller than that of $\mathcal{C}$, the existing algorithms can try a large value for the distance threshold so that the distance is sufficient to find $\mathcal{C}$. By using such a large value, we will not miss $\mathcal{C}$ and other prevalent co-locations occurring at smaller distances. Another observation is that the larger the distance threshold is, the higher the number of instances of a feature participating in some co-location types will be. A random pattern can attain a high prevalence measure value, when a large distance threshold is used by the existing algorithms and get reported as prevalent.

5.1.3 Our Contributions

To the best of our knowledge, mining meaningful co-location patterns without any prior information of their interaction distance is a relatively new problem in spatial data mining research. Finding a statistically sound solution for such a problem can be challenging in the presence of spatial auto-correlation and feature abundance, which are not uncommon in the spatial domain. The contributions of this chapter are as follows:

- We propose a model to mine all true co-location patterns occurring at different distances in a given data set. Our approach neither requires a prevalence measure threshold nor a distance threshold, which are the essential parameters of the existing co-location mining algorithms. Our algorithm determines the co-location distance of a true co-location.
- To ensure that our mined patterns are not occurring just by chance, we propose a model for a statistical significance test. This model considers the effects of spatial auto-correlation and feature abundance, which often mislead even a standard statistical test on spatial data, as well as the existing algorithms, in reporting true patterns.
- We further propose a post-processing step which prunes redundant patterns and finally keeps a minimal set of patterns that is sufficient to explain all positive interactions in the data.
- We validate our approach with synthetic and real data sets.

5.2 Statistically Significant Co-locations at Multiple Interaction Distances - Definitions and Concepts

For a given data set, our objective is to determine if a subset of features is exhibiting a true co-location. If a true co-location exists among a group of features, we determine the co-location distance. For a group of features $\mathcal{S}$ exhibiting a true co-location at d, the observed co-location property will rarely be seen under a null hypothesis based on an independence assumption. To test if $\mathcal{S}$ exhibits a true co-location, we compute the probability (p-value) of obtaining a co-location property of $\mathcal{S}$ at d (under a null hypothesis) at least as extreme as the one that was actually observed in the given data. If $\mathcal{S}$ is a true co-location, this computed probability will not be greater than a given level of significance (α). For the test described above, we need a measure which can compute the co-location (i.e. aggregation) property of $\mathcal{S}$ at a distance d. Let $\mathcal{T}$ be such a measure. From the given data, we identify all unique instances of $\mathcal{S}$ and also the distances at which these instances are identified. For each identified distance d, we compute the $\mathcal{T}$-value and then test the significance of the $\mathcal{T}$-value of $\mathcal{S}$ observed at d. If the observed $\mathcal{T}$-value at d is significant, we report subset $\mathcal{S}$ as a significant co-location pattern at d.

5.2.1 Definitions

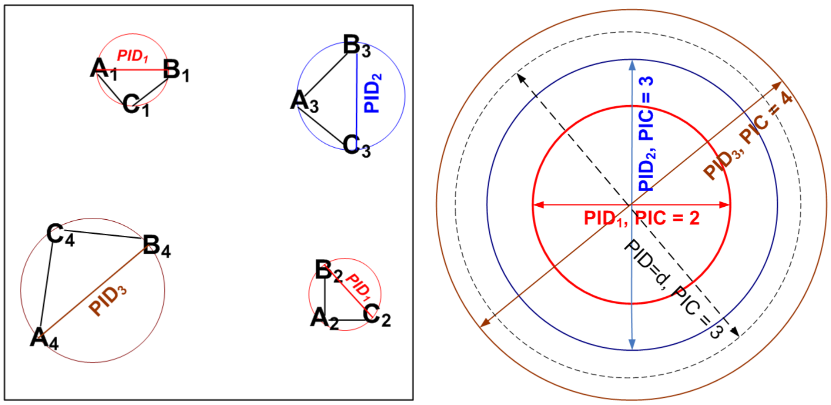
To formulate our problem in a better way, we provide some definitions.

Definition 6. *A co-location pattern is a subset of k different features $f_1, f_2, \ldots, f_k$ which are more likely to been seen together due to a spatial relationship R interacting at distance R_d. The interaction distance R_d is called the "co-location distance." Feature instances that are involved in a co-location instance are considered neighbors of each other. Two feature instances are neighbors of each other if their Euclidian distance is not larger than R_d. Let $\mathcal{C} = \{f_1, f_2, \ldots, f_k\}$ be a co-location pattern. In an instance of $\mathcal{C}$, one instance from each of the k features will be present and all these feature instances are neighbors of each other.*

Definition 7. *The Pattern Instance Distance (PID) of a pattern instance I is the maximum pair-wise distance from all the participating members of I. More formally, let us assume a pattern $\mathcal{C}$ is a group of k features $\{f_1, f_2, \ldots, f_k\}$ and $I_{\mathcal{C}} = \{I_{f_1}, I_{f_2}, \ldots, I_{f_k}\}$ be an instance of $\mathcal{C}$ where I_{f_i}, a member of $I_{\mathcal{C}}$, is an instance of the participating feature f_i of $\mathcal{C}$. The PID of $I_{\mathcal{C}}$, $PID^{I_{\mathcal{C}}}$, is computed as $PID^{I_{\mathcal{C}}} = \max\{\text{dist}(I_{f_i}, I_{f_j}) | I_{f_i}, I_{f_j} \in I_{\mathcal{C}} \wedge i \neq j\}$ where 'dist' is the Euclidian distance function.*

Let $\mathcal{C} = \{A, B, C\}$. Assume feature A has 4 instances (A_1, A_2, A_3 and A_4), feature B has 4 instances (B_1, B_2, B_3 and B_4) and feature C has 4 instances (C_1, C_2, C_3 and C_4). At most 64 instances of $\mathcal{C}$ are possible. Among those, 4 instances such as $\{A_1, B_1, C_1\}$, $\{A_2, B_2, C_2\}$, $\{A_3, B_3, C_3\}$, and $\{A_4, B_4, C_4\}$ and their co-location neighborhoods are shown in Fig. 5.1(a). Both $\{A_1, B_1, C_1\}$ and $\{A_2, B_2, C_2\}$ are identified at the same PID distance PID_1. $\{A_3, B_3, C_3\}$, and $\{A_4, B_4, C_4\}$ are identified at their PID distance of PID_2 and PID_3 respectively. Let us assume $PID_1 < PID_2 < PID_3$. For instance, the PID distance of $\{A_1, B_1, C_1\}$, $PID_1 = max\{dist(A_1, B_1), dist(A_1, C_1), dist(B_1, C_1)\}$.

Definition 8. *The Pattern Instance Count (PIC) of a pattern $\mathcal{C}$ at a distance d is the total number of instances of $\mathcal{C}$ whose PID is at most d. More formally, let $\mathcal{C}$ be a pattern and $I_{\mathcal{C}}$ be an instance of $\mathcal{C}$. Then, the PIC-value of $\mathcal{C}$ at a distance d is computed as $PIC^{\mathcal{C}_d} = |\{I_{\mathcal{C}} | PID^{I_{\mathcal{C}}} \leq d \wedge I_{\mathcal{C}} \textit{ is an instance of } \mathcal{C}\}|$.*



(a) A data set with 3 features A, B, and C. 3 smallest PID distances are also shown

(b) PIC-values of different PID distances; a circle represents a co-location neighborhood

Figure 5.1: a) 4 instances of $\{A, B, C\}$ and their co-location neighborhoods. b) Relationship between PIC and PID.

For the above mentioned pattern $\mathcal{C} = \{A, B, C\}$, the PIC-value of $\mathcal{C}$ at $d = PID_3$ will be 4 and at a d where $PID_2 < d < PID_3$ will be 3. This is shown in Fig. 5.1(b).

Lemma 3. *The PIC-value of a pattern is monotonically non-decreasing with the increasing value of the PID distance.*

The PIC-value of the above mentioned pattern $\mathcal{C} = \{A, B, C\}$ at PID_1 and PID_2 (where $PID_1 < PID_2$) are respectively 2 and 3.

5.2.2 *PIC* as a Test Statistic for the Statistical Significance Test

The PI is mostly used as the prevalence measure in the current literature. SSCSP also uses the PI as the test statistic for the statistical test. However, any measure capturing the spatial dependency of a subset $\mathcal{S}$ can be used as a statistic for testing the significance of $\mathcal{S}$. If the member features of $\mathcal{S}$ are truly associated at a distance d, the number of instances of $\mathcal{S}$ observed at d will be much higher than the expected number of instances found at d under a null model. The total instances of $\mathcal{S}$ identified at d gives the PIC-value of $\mathcal{S}$ at d.

We use PIC as a test statistic for the statistical significance test and show its ability to find all true co-locations at their "correct" co-location distances. Furthermore, PIC, as a test statistic, can give computational advantages over the PI. PI-value computation of a pattern $\mathcal{C}$ at a given distance d requires first identifying all the instances of $\mathcal{C}$ considering a PID-value no greater than d and then enumerating the identified instances in a table instance of $\mathcal{C}$. Using the table instance, we then compute the participation ratio of each participating feature of $\mathcal{C}$. By using PIC as a test statistic, we can avoid the computation of the participation ratios.

Existing algorithms use the PI as a prevalence measure and report $\mathcal{C}$ if its PI-value is greater than a given PI-threshold. SSCSP also uses the PI to find statistically significant co-locations. However it finds co-locations only for a given distance threshold. Here we generalize the problem and propose a solution by which we will be able to find all statistically significant co-locations occurring at different distances. In other words, the proposed approach does not require the distance threshold parameter. Instead of PI, our approach uses PIC as the test statistic. We call this new approach ***"Co-location Pattern Mining with No Distance Constraint (CPMNDC)."***

We define a statistically significant co-location pattern $\mathcal{C}$ occurring at distance R_d (due to a spatial interaction R) using the PIC-value of $\mathcal{C}$ computed at R_d, $PIC^{\mathcal{C}_{R_d}}$, as the test statistic.

Definition 9. *A pattern $\mathcal{C} = \{f_1, f_2, \ldots, f_k\}$ at a distance R_d is called a **statistically significant co-location pattern** at level α, if the probability (p-value) of seeing, in a data set conforming to our null hypothesis, a $PIC^{\mathcal{C}_{R_d}}$-value larger than or equal to the observed $PIC^{\mathcal{C}_{R_d}}$-value is not greater than α.*

A pattern $\mathcal{C}$ is reported as a statistically significant co-location pattern at distance R_d, if the number of instances of $\mathcal{C}$ identified at R_d ($PIC^{\mathcal{C}_{R_d}}$) in the given data set is so high that seeing an equal or higher value at distance R_d will be very unlikely (the probability will not be greater than α) under the null model. Let us denote the observed $PIC^{\mathcal{C}_{R_d}}$-value (computed from the given data set) as $PIC_{\text{obs}}^{\mathcal{C}_{R_d}}$ and the $PIC^{\mathcal{C}_{R_d}}$-value computed under the null model as $PIC_0^{\mathcal{C}_{R_d}}$. We compute the probability (known as p-value) of obtaining a $PIC_0^{\mathcal{C}_{R_d}}$-value under the null model being equal or higher than the $PIC_{\text{obs}}^{\mathcal{C}_{R_d}}$, i.e. p-value $= Pr(PIC_0^{\mathcal{C}_{R_d}} \geq PIC_{\text{obs}}^{\mathcal{C}_{R_d}} | \text{null model})$. The null model assumes that features of different types do not exhibit any spatial interaction with each other but maintain their individual distributional properties as in the observed data.

To compute a p-value, we do randomization tests where we simulate the null model. A detailed discussion on how to simulate the null model is given in Section 3.3 of Chapter 3. By running a sufficient number of simulations in the randomization tests, we compute the

p-value for $\mathcal{C}$ at distance R_d, $p^{\mathcal{C}_{R_d}}$, as:

$$p^{\mathcal{C}_{R_d}} = \frac{S^{\geq PIC_{\text{obs}}^{\mathcal{C}_{R_d}}} + 1}{S + 1} \tag{5.1}$$

Here $S^{\geq PIC_{\text{obs}}^{\mathcal{C}_{R_d}}}$ of equation (5.1) represents the number of simulations where the computed $PIC_0^{\mathcal{C}_{R_d}}$-value is greater than or equal to the $PIC_{\text{obs}}^{\mathcal{C}_{R_d}}$-value. S represents the total number of simulations. In both the numerator and the denominator, one is added to account for the observed data. If $p^{\mathcal{C}_{R_d}} \leq \alpha$, we report pattern $\mathcal{C}$ as a statistically significant co-location occurring at distance R_d. α is the probability of committing a type I error which is rejecting the null hypothesis when it is in fact true.

For each pattern, we are conducting the hypothesis test at each unique PID-value. When m hypothesis tests are performed, the probability of committing at least one error in m tests will be $1 - (1 - \alpha)^m$ and this probability increases as m increases. Hence, the significance level for a single test is adjusted with the number of tests and is required to be much smaller to ensure the same overall rate of type I errors. Several methods such as [11, 30, 33] are proposed to adjust the significance level α for a single test. The classic *Bonferroni correction* adjusts the α-value of a single test by dividing it by the number of performed tests [30]. For a pattern, the required number of tests will be equal to the total number of unique PID-values. In our case, the Bonferroni correction will give a very low α-value for a single test. To many researchers, the Bonferroni correction is too conservative. It also leads to a high probability of a type II error, i.e. not rejecting the null hypothesis at each unique PID distance when significant patterns at a PID distance, in fact, exist. Additionally, allowing a certain number of false positives is accepted in many applications such as genomics.

Our approach mines true patterns in two steps. The first step performs a statistical test based on a *simple* null hypothesis assuming the independence of all spatial features. This results in a set of statistically significant patterns which are again refined by performing another statistical test in the second step. The statistical test of the second step uses a *constrained* null hypothesis that assumes the independence of spatial features together with a given set of constraints (i.e. rules). Due to the multiple hypothesis tests done at multiple PID distances, a random subset could be reported as a true pattern in the first step. If such a pattern exists, it is pruned in the second step. Thus, the second step works as a filtering step and also removes the necessity of using a very low α-value that may lead to missing a true pattern. We use $\alpha = 0.01$ for a single test. For that we conduct 499 simulations according to the recommendation of Besag and Diggle in [12]. If a pattern is found significant at more than one distance, we report the minimum distance at which the pattern attains the highest PI-value.

Definition 10. *For a pattern $\mathcal{C}$ being significant at more than one PID-value, the smallest PID-value that gives the highest PI-value is considered as the co-location distance R_d of $\mathcal{C}$. Hence ${R_d}^{\mathcal{C}} = min\{d | p^{\mathcal{C}_d} \leq \alpha \wedge \forall d' : PI^{\mathcal{C}_{d'}} \leq PI^{\mathcal{C}_d}\}$.*

Let $\mathcal{C}$ be found significant at 3 PID-values, d_1, d_2, and d_3, where $d_1 < d_2 < d_3$. Let us assume that the PI-values at those 3 distances are $PI^{\mathcal{C}_{d_1}}$, $PI^{\mathcal{C}_{d_2}}$, and $PI^{\mathcal{C}_{d_2}}$ respectively. $PI^{\mathcal{C}_{d_1}} < PI^{\mathcal{C}_{d_2}}$ and $PI^{\mathcal{C}_{d_2}} = PI^{\mathcal{C}_{d_3}}$. In this example, d_2 is reported as the co-location distance R_d of $\mathcal{C}$.

5.3 Algorithms

For a given spatial data set, our objective is to mine all statistically significant and non-redundant co-location patterns and their co-location distances. For a candidate pattern $\mathcal{C}$, a statistical test is performed to check the significance of its observed co-location behavior at a distance. Here the pattern instance count is used as a measure for the co-location behavior of a pattern. The next question is which distances are going to be tested for $\mathcal{C}$. We identify all possible instances of $\mathcal{C}$. To identify them, we first compute all possible pair-wise distances from the instances of different features found from the given data set and sort them in an increasing order. Then we try each pair-wise distance from the sorted list one by one and check if a considered distance gives at least one instance of $\mathcal{C}$ which has not been identified before at any of the pair-wise distances smaller than the currently considered one. By doing so, we obtain a sorted list of distances; each distance d of that list gives at least one instances of $\mathcal{C}$ which can not be identified by considering a distance smaller than d. Finally, all unique distances are considered for the significance test of $\mathcal{C}$. At each unique distance d', we now compute the pattern instance count of $\mathcal{C}$ as the sum of all instances identified at distances smaller than and equal to d'. Then we perform the significance test of $\mathcal{C}$ at each d' using the computed pattern instance count at d' as the test statistic.

Algorithm 4 and 5 present the pseudo codes of our approach. Features are ordered either arbitrarily or using domain related information. For each candidate pattern, Algorithm 4 first finds its instances. Each pattern instance occurs at a PID distance. The algorithm then records all unique PID distances and tests if the observed PIC-value at each of these PID distances is statistically significant or not. If the observed PIC-value at a PID distance is found significant, the pattern is recorded in the result. Finally, we determine the co-location distance of each pattern recorded in the result. Algorithm 4 may report redundant patterns that are not true patterns but occur due to the presence of true patterns. Such redundant patterns are pruned by Algorithm 5, which works as a filtering step. Algorithm 5 reports a set of minimal co-location patterns that explains all the reported patterns of Algorithm 4. The reported patterns become statistically significant and non-redundant.

5.3.1 Statistically Significant Co-location Pattern Mining with No Distance Constraint

In the following, we will describe Algorithm 4 in more technical detail.

For each candidate pattern $\mathcal{C}$ (a subset of features), all unique PID-values are identified from the given data set and stored along with two additional values, the $PIC^{\mathcal{C}_{PID}}$-values and the $S^{\geq PIC^{\mathcal{C}_{PID}}}$-counters. The $PIC^{\mathcal{C}_{PID}}$-value gives the total number of instances of $\mathcal{C}$ identified at the PID distance. The counter $S^{\geq PIC^{\mathcal{C}_{PID}}}$ is used to compute the p-value of $\mathcal{C}$ at the PID distance. It is initially set to zero and incremented by one in a simulation during the randomization tests. We store this information for all patterns of size k in a record $\mathcal{C}^k_{\text{obs}}$.

Line $1-2$**:** From the given data set, we first find all the pair-wise distances, sort them in increasing order, and finally store them in $\mathcal{D}$. Pair-wise distances are only computed for two features of *different* types. All distance based measures introduced in spatial statistics use distance values less than either one-half of the shortest dimension of an approximately rectangular shaped study region or $(A/2)^{\frac{1}{2}}$ where A is the area of the study region. The same strategy is used in our case to set an upper limit for the considered distance values to find the meaningful co-location distances of "true" co-locations. Having an upper bound on the meaningful pair-wise distances also allows to speed up the pair-wise distance calculations using simple spatial index structures.

Line $3-5$**:** Each pair-wise distance $d \in \mathcal{D}$ is the PID-value of at least one pattern instance of size 2. It could also be the PID-value of pattern instances of sizes larger than 2. We identify all these pattern instances. By definition, a pattern instance is a clique. All the feature instances can be seen as the vertices of an undirected graph $\mathcal{G}$. Initially $\mathcal{G}$ is without edges. By considering a distance d in $\mathcal{D}$, we are in fact adding an edge e between a pair of

Algorithm 4: CPMNDC: Co-location Pattern Mining with No Distance Constraint

```
Input: A spatial data set $\mathcal{SD}$ with $N$ spatial features $\mathcal{F} = f_1, f_2, \cdots f_N$ (each $f_i$ has
    $n_{f_i}$ instances). Level of significance $\alpha$, and number of simulation runs $S$.
Output: A set of co-location patterns $\mathcal{SC}$, each with their co-location distances $R_d$.
Variables:
    $\mathcal{C}^k_{\text{obs}}$: Stores the set of all $k$-size candidate patterns from the given data set. Each
    pattern is stored along with a list of $PID_{\text{obs}}$-distances, $PIC_{\text{obs}}$-values, and
    $S^{\geq PIC_{\text{obs}}}$-values.
    $\mathcal{C}^k_0$: Stores the set of all $k$-size candidate patterns from a simulation. Each pattern
    is stored along with a list of $PID_0$-distances and $PIC_0$-values found from a
    simulation $S_i$.
Method:
Compute pair-wise distances between instances of different feature types in $\mathcal{SD}$
Sort the distances in increasing order and store them in $\mathcal{D}$
for each $d \in \mathcal{D}$ in increasing order do
    Find pattern instances for which the $PID$-value is exactly $d$
    For each identified instance of a pattern, we store the $PID_{\text{obs}}$ and $PIC_{\text{obs}}$ in $\mathcal{C}^k_{\text{obs}}$
for $i = 1$ to $\mathcal{S}$ do
    Generate a simulated data set $\mathcal{SD}_0$ under the null model
    Compute pair-wise distances among instances of different feature types in $\mathcal{SD}_0$
    Sort the distances in increasing order and store them in $\mathcal{D}_0$
    for all $d \in \mathcal{D}_0$ do
        Find pattern instances which have the $PID$-value of exactly $d$
        For each identified instance of a pattern, we store the $PID_0$ and $PIC_0$ in $\mathcal{C}^k_0$
    for each candidate pattern $\mathcal{C}$ of $k$-size; $2 \leq k \leq N$ do
        for each $PID_{\text{obs}}$-value of $\mathcal{C}$ do
            Set the $PIC_{\text{obs}}$ of $PID_{\text{obs}}$ to $T_1$
            From $\mathcal{C}^k_0$, find the $PID_0$-value that is equal to the $PID_{\text{obs}}$-value. If no
            such value exists, find a maximal value that $\ngtr PID_{\text{obs}}$-value and set the
            $PIC_0$ of $PID_0$ to $T_2$
            if $T_2 \geq T_1$ then
                Increment the $S^{\geq PIC_{\text{obs}}}$-value of $\mathcal{C}$ by 1
// $p$-value computation of each $\mathcal{C}$ at all its $PID_{\text{obs}}$-values
for each candidate pattern $\mathcal{C}$ of $k$-size; $2 \leq k \leq N$ do
    for each $PID_{\text{obs}}$-value of $\mathcal{C}$ do
        if $\frac{S^{\geq PIC_{\text{obs}}}+1}{S+1} \leq \alpha$ then
            Include $\mathcal{C}$ and the $PID_{\text{obs}}$-value in $\mathcal{SC}$
for each candidate pattern $\mathcal{C}$ in $\mathcal{SC}$ do
    $\mathcal{SC} \leftarrow$ ReportColocationDistance($\mathcal{SC}$)
return $\mathcal{SC}$
```

vertices of $\mathcal{G}$. Then we find the cliques that occur due to the inclusion of e in graph $\mathcal{G}$. e will be the largest edge of these found cliques. Let us assume edge e of length d connects two instances I_{f_i} of feature type f_i and I_{f_j} of feature type f_j. To efficiently identify all cliques that occur due to the inclusion of edge e, we only check a subset of vertices in $\mathcal{G}$, instead of checking all vertices of $\mathcal{G}$. Only the vertices that are already connected to both I_{f_i} and I_{f_j} by edges in $\mathcal{G}$ are included in such a subset. This subset can be found efficiently by identifying the members (feature instances) of the star neighborhoods [72] of I_{f_i} and I_{f_j} and then finding a subset of feature instances, Υ, that are present in both star neighborhoods. We maintain an adjacency matrix $\mathcal{M}$ which helps to construct the star neighborhood of a feature instance. Matrix $\mathcal{M}$ is of size $F_N \times F_N$ (F_N is the total number of feature instances) and all the entries are initially set to 0. For each pair-wise distance d in $\mathcal{D}$, $\mathcal{M}$ is updated by setting both (I_{f_i}, I_{f_j}) and (I_{f_j}, I_{f_i}) entries of $\mathcal{M}$ to 1.

If Υ is empty, only one clique that is the pattern instance $\{I_{f_i}, I_{f_j}\}$ is generated due to the inclusion of edge e of length d. If $|\Upsilon| = 1$, two cliques are generated. Let the member of Υ be $\{I_{f_k}\}$, then the generated cliques are the pattern instances $\{I_{f_i}, I_{f_j}\}$ and $\{I_{f_i}, I_{f_j}, I_{f_k}\}$. If $|\Upsilon| > 1$, we have to find cliques from the members (feature instances) of Υ. We use the adjacency information of the members of $|\Upsilon|$ stored in $\mathcal{M}$ to find cliques from Υ. The run-time of most clique finding algorithms is exponential. For instance, the popular BronKerbosch algorithm [14] has a run-time of $O(3^{n/3})$ in the worst case. In finding cliques from Υ, we use the algorithm of Tsukiyama *et al.* [64]. Tsukiyama showed that all maximal cliques can be enumerated in a polynomial time per output. The algorithm finds the independent vertex sets. An independent set is a set of vertices in a graph, no two of which are connected by an edge. A clique of a graph $\mathcal{G}$ is an independent vertex set of the complement (or inverse) graph of $\mathcal{G}$. Hence, the clique finding problem and independent vertex set problem are complementary. Let $\mathcal{CL}$ be a set of cliques that are identified from the members of Υ. Each clique $c \in \mathcal{CL}$ generates a pattern instance which is comprised of I_{f_i}, I_{f_j}, and all the members (feature instances) of c. When $|\Upsilon| > 1$, the identified pattern instances are the instances generated from $\mathcal{CL}$ and $\{I_{f_i}, I_{f_j}\}$. Hence, the total number of pattern instances that are identified by adding an edge e of length d will be $|\mathcal{CL}| + 1$. The PID-value of all these identified pattern instances due to the inclusion of edge e is exactly d since d is the largest edge in each pattern by construction (we consider distances in increasing order of length). Finally, for each identified pattern instance we update the record of its corresponding pattern type $\mathcal{C}$ by adding d as the PID-value and a $PIC^{\mathcal{C}_{PID}}$-value. The $PIC^{\mathcal{C}_{PID}}$-value is the sum of the new instances of $\mathcal{C}$ identified at d and the preceding

PIC-value (at the previous PID-value) that is already stored in the record of $\mathcal{C}$.

Line $6-12$**:** We do randomization tests in which we generate a simulated data set according to the null hypothesis (line 7). From the generated data set, first we compute all the pair-wise distances, sort them, and store them (Line 8). Then we identify all pattern instances from this generated data set by applying the same strategy that is used for identifying pattern instances from the observed data. We use a record $\mathcal{C}_0^k$ to store the PID distance and the PIC-value of all pattern instances identified at each pair-wise distance (line $10-12$). We use the same record $\mathcal{C}_0^k$ in all simulations.

Line $13-18$**:** At the end of a simulation, we update the $S^{\geq PIC_{\text{obs}}}$ counter for each unique observed PID distance (PID_{obs}) of a pattern $\mathcal{C}$. For each PID_{obs}-value, we compare its PIC-value (PIC_{obs}) with the PIC-value computed at a distance of PID_{obs} from the generated data of a simulation. From the PID_0 distances stored in $\mathcal{C}_0^k$, we look for a distance equal to PID_{obs}. If such a value is found, we compare the corresponding PIC_0 with PIC_{obs}. If no such value exists, from the stored PID_0-values we select the one that is maximal and is not greater than PID_{obs}. After finding the PID_0-value, we get the corresponding PIC_0-value stored in $\mathcal{C}_0^k$ and compare it with the PIC-value (PIC_{obs}-value) of the PID_{obs}-value. As in the generated data, no instance of $\mathcal{C}$ with a PID-value of exactly PID_{obs} exists; the PIC_0-value at a PID distance of exactly PID_{obs} will be the same as the PIC_0 value of a stored PID_0 distance which is maximal and is not greater than PID_{obs}. This can also be explained using the example shown in Fig. 5.1(b). In this figure, let us assume d is our observed PID distance and PID_1, PID_2, and PID_3 are the only PID distances found in the generated data of a simulation and stored in $\mathcal{C}_0^k$. In this case, the PIC-value at a PID distance of d will be same as the PIC-value at PID_2, which is equal to 3. Finally, if $PIC_0 \geq PIC_{\text{obs}}$, we increment $S^{\geq PIC_{\text{obs}}}$ of $\mathcal{C}$ by 1.

Line $19-23$**:** Here we compute the p-values for all the candidate patterns at their PID_{obs}-values using equation (5.1). A pattern along with the PID_{obs}-value is stored in the reported pattern list if the computed p-value at PID_{obs} is less than the α-value.

Line $23-24$**:** Function *ReportColocationDistance* decides the co-location distance of $\mathcal{C}$, when $\mathcal{C}$ is found significant at more than one PID_{obs}-value. In such a case, we first compute the PI-values of $\mathcal{C}$ at those distances (PID_{obs}-values) where $\mathcal{C}$ is found as statistically significant. The smallest PID_{obs}-value that first yields the highest PI-value among all computed PI-values is reported as the co-location distance R_d of $\mathcal{C}$. For instance, let $\mathcal{C}$ be found significant at 4 distances, PID^1_{obs}, PID^2_{obs}, PID^3_{obs}, and PID^4_{obs}, and the computed PI-values at those distances are PI^1, PI^2, PI^3, and PI^4, respectively. Let us assume $PI^1 < PI^2 < PI^3$, and $PI^3 = PI^4$. We report PID^3_{obs} as the co-location distance of $\mathcal{C}$.

Complexity

The run-time cost of finding all pair-wise distances is $\sum n_{f_i} \times \sum n_{f_i}$ where $\sum n_{f_i}$ is total feature instances present in the data set. The cost of finding all cliques from a vertex set Υ (with n vertices, m edges, and r maximal cliques) using the algorithm of Tsukiyama *et al.* is $O(n * m * r)$ [64]. Please note that initially $n \ll \sum n_{f_i}$ and increases with increasing values of the distance d. However, by limiting the maximum value of d (as mentioned in the description of step $1 - 2$ of section 5.3.1), we can keep this cost at a feasible level. Let us assume the expected number of vertices, edges, and maximal cliques found in Υ are $\hat{n}$, $\hat{m}$, and $\hat{r}$, respectively. The total run-time cost will be of $O(\sum n_{f_i} * \sum n_{f_i} * \hat{n} * \hat{m} * \hat{r})$. As the cost of computing pair-wise distances is the major computational cost, the total run-time cost can be approximated to $O((\sum n_{f_i})^2)$.

5.3.2 Redundant Pattern Pruning

The pattern set $\mathcal{SC}$ reported by Algorithm 4 could have "redundant" patterns. A redundant pattern is a random pattern that is mistakenly reported as significant due to the presence of true patterns. Our null model assumes that features are independent of each other. A statistical test using such an assumption may result in reporting redundant patterns whose subsets or supersets happen to be true co-locations. Instances of randomly distributed features when appearing close to the instances of a true co-location $\mathcal{C}_{\mathcal{T}}$ get co-located with the participating features of $\mathcal{C}_{\mathcal{T}}$ and generate some new co-locations. A new co-location $\mathcal{C}_{\mathcal{R}}$ generated in such a scenario should not be considered as a true co-location as the participating random features do not have a true interaction with the participating features of $\mathcal{C}_{\mathcal{T}}$. The amount of instances of $\mathcal{C}_{\mathcal{R}}$ we see in such a scenario can be higher than the amount of instances we expect to see in a data set if generated based on our null hypothesis assumption (i.e. all participating features are distributed independently of each other). Hence, a statistical test based on our null hypothesis may result in a p-value lower than α for $\mathcal{C}_{\mathcal{R}}$ which results in reporting $\mathcal{C}_{\mathcal{R}}$ as statistically significant. However, in a generated data set where the presence of the true co-location $\mathcal{C}_{\mathcal{T}}$ is taken into account, seeing the same number of instances of $\mathcal{C}_{\mathcal{R}}$ found from the given data will not be unusual but rather quite common. To avoid reporting redundant patterns from our statistical test, we now refine our null hypothesis and propose a new null hypothesis where true co-locations (if present) are also taken into account in addition to the spatial distribution of each individual feature, and then we perform a statistical test based on this new null hypothesis.

Let us consider a data set with three features A, B, and C and $\{A, B\}$ as the only true pattern. The presence of $\{A, B\}$ could induce reporting $\{A, B, C\}$ as significant even though instances of feature C are randomly distributed. The number of instances of $\{A, B, C\}$ that are generated due to the presence of instances of C in the neighborhoods $\{A, B\}$ could appear as a high number compared to the expected number of instances of $\{A, B, C\}$ observed in a null model. The null model simply assumes the independence of A, B, and C. Hence a low p-value of $\{A, B, C\}$, even lower than the α-value, could be possible. To avoid $\{A, B, C\}$ being reported, the p-value of $\{A, B, C\}$ should rather be computed from a null model that takes into account the association of feature A and feature B, instead of simply considering them independent of each other. We call this a null model with a constraint set $\{\{A, B\}\}$. The p-value of $\{A, B, C\}$ in a null model with a constraint set $\{\{A, B\}\}$ will be $Pr(PIC_0^{\{A,B,C\}_{R_d}} \geq PIC_{\text{obs}}^{\{A,B,C\}_{R_d}} | \{\{A, B\}\})$ and if $p > \alpha$, we say $\{A, B, C\}$ is *explained* by $\{A, B\}$. The observed co-location tendency of feature A, feature B, and feature C (although not involved in a true association) appeared as significant in our statistical test due to the existence of the true co-location of feature A and feature B. In another example, a subset can also be reported as significant if a superset is a true co-location. For instance, $\{A, B\}$, although not a true co-location by itself, could still be reported as significant if a superset of $\{A, B\}$ such as $\{A, B, C\}$ is a true co-location. Here the observed number of instances of $\{A, B\}$ can be unusually higher than the expected number of instances seen in our null model, but most of the appearances of $\{A, B\}$ are from the instances of $\{A, B, C\}$. A p-value of $\{A, B\}$ is computed from a null model with a constraint set $\{\{A, B, C\}\}$ as $Pr(PIC_0^{\{A,B\}_{R_d}} \geq PIC_{\text{obs}}^{\{A,B\}_{R_d}} | \{\{A, B, C\}\})$ and compared with the α-value. If $p > \alpha$, $\{A, B\}$ is considered as a redundant pattern and we say $\{A, B\}$ is *explained* by $\{A, B, C\}$. However, both $\{A, B, C\}$ and $\{A, B\}$ will be true patterns if neither of them is *explained* by the other, i.e. their p-values are both at most α.

Definition 11. *Let $PIC_{obs}^{\mathcal{C}_d}$ be the observed PIC-value of a co-location $\mathcal{C}$ at distance d and $PIC_{0_{\mathcal{E}}}^{\mathcal{C}_d}$ be the PIC-value at d computed under a null model with a constraint set $\mathcal{E}$. The p-value of $PIC_{obs}^{\mathcal{C}_d}$ at distance d under a null model with a constraint set $\mathcal{E}$ is the probability $p = Pr(PIC_{0_{\mathcal{E}}}^{\mathcal{C}_d} \geq PIC_{obs}^{\mathcal{C}_d} | \mathcal{E})$. A pattern set $\mathcal{E}$* ***explains*** *a pattern $\mathcal{C}$ with $PIC_{obs}^{\mathcal{C}_d}$, formally $\mathcal{E}$ explains $\mathcal{C}$, if the p-value of $\mathcal{C}$ with $PIC_{obs}^{\mathcal{C}_d}$ computed under the null model with the* *constraint set $\mathcal{E}$ is larger than the given level of significance α. If $\mathcal{E}$* ***explains*** *every pattern of a set $\mathcal{SC}$, we say $\mathcal{E}$ explains $\mathcal{SC}$.*

Using the *explain* relationship, we can now define a set of non-redundant and statistically significant co-location patterns $\mathcal{E}$ that explains all the significant patterns returned by Algorithm 4. $\mathcal{E}$ explains itself, and several sets of patterns that explain $\mathcal{SC}$ may exist. We prefer one that has the minimum cardinality among all such sets.

Definition 12. *A minimal "explaining" co-location pattern set is a set $\mathcal{E}$ that explains all statistically significant patterns of $\mathcal{SC}$ and that has the minimum cardinality among all such sets.*

To find a smallest subset of $\mathcal{SC}$ that explains all the significant patterns of $\mathcal{SC}$, we need to test all possible subsets of $\mathcal{SC}$. Finding such a smallest subset of $\mathcal{SC}$ has the complexity of $2^{|\mathcal{SC}|}$ that becomes expensive with the increasing size of $\mathcal{SC}$. To solve this problem efficiently, we propose a greedy strategy which can give an approximate solution.

Design of a Null Model with a Constraint Set

In a randomization test based on a null model with no constraint set, we generate data sets where each feature maintains the same spatial distribution seen in the observed data. If a null model has a constraint set, we implant the co-locations of the constraint set into the generated data of simulations. These implanted co-locations should have co-location properties similar to the one seen in the observed data. To the best of our knowledge, no model exists that can generate instances for given co-location properties. In spatial statistics some models have been introduced that can simulate spatial interactions among a pair of spatial features [6, 7]. However, these interactions are of the inhibition type. Models to simulate a positive interaction (aggregation or clustering) among instances of the *same* feature type are also proposed. However, there is no model that can simulate a positive interaction between instances of *different* feature types. A strategy to simulate an observed co-location $\mathcal{C}$ can be developed using the concept of a *reconstruction algorithm* [37, 63]. We first estimate some interaction measures (such as the J-function) of the participating features of $\mathcal{C}$ from the observed data. Then we start with a data set where points (feature instances) are uniformly distributed and conduct an iterative process. In an iteration, we translate points in space and compute the same interaction measures from the translated data set. We keep a translated data set to use it for the next interaction if the interaction measure values from the translated data set converge to the one seen in the observed data.

We continue iterations until the difference between the computed values from the translated data and that from the observed data is greater than a given level of error. This method works well for a small data set but becomes computationally expensive as the number and size of the patterns to simulate increases.

In order to solve this problem, we propose the following heuristic to simulate a co-location $\mathcal{C}$ from a given data set. If $\mathcal{C}$ is in a constraint set, in a simulated data set the instances of $\mathcal{C}$ as observed in the given data are maintained. This is achieved by maintaining the locations of the participating feature instances of $\mathcal{C}$ found in the given data set. Instances of the participating features of $\mathcal{C}$ which are not involved in co-location type $\mathcal{C}$ are distributed according to their own spatial distribution. Instances of non-participating features of $\mathcal{C}$ are also distributed according to their own spatial distribution. To justify if the significance of pattern $\{A, B\}$ is explained by the significance of pattern $\{A, B, C\}$, we compute the p-value of $\{A, B\}$ from a null model with $\{A, B, C\}$ in the constraint set. To generate a data set during simulation using such a null model, we maintain the same location from the given data only for those instances of A, B, and C that are involved in co-location type $\{A, B, C\}$. The remaining instances of A, B, and C and instances of features other than A, B, and C (that exist in the given data set) are distributed independently of each other but maintain the same individual spatial distribution observed in the given data. Thus we can generate a data set that maintains the co-location properties of the constraint set as well as the spatial distribution property of individual feature seen in the given data set.

An Approximation Algorithm to Find a Minimal Explaining Pattern Set

To find an approximate solution for a minimal *explaining* pattern set $\mathcal{P}^{\text{sol}}$, our approximation algorithm follows a greedy forward selection strategy. Algorithm 5 shows the pseudo code of our approach. To find a minimal explaining set of $\mathcal{SC}$ reported from Algorithm 4, we use a greedy approach in selecting a pattern from $\mathcal{SC}$. Our greedy approach chooses a pattern that explains the highest number of patterns from $\mathcal{SC}$. Let $\mathcal{P}^{\text{rest}}$ denote the current set of patterns. The patterns of $\mathcal{P}^{\text{rest}}$ are not explained by the patterns in the current solution $\mathcal{P}^{\text{sol}}$. $\mathcal{P}^{\text{sol}}$ and $\mathcal{P}^{\text{rest}}$ are initialized with an empty set and $\mathcal{SC}$, respectively (line $1-2$). The algorithm checks each pattern $\mathcal{P}$ from $\mathcal{P}^{\text{rest}}$ and finds the pattern set explained by $\mathcal{P}^{\text{sol}} \cup \mathcal{P}$ (line $6-21$). To find the explained pattern set of $\mathcal{P}^{\text{sol}} \cup \mathcal{P}$ from $\mathcal{P}^{\text{rest}}$, we compute the p-value of each pattern of $\mathcal{P}^{\text{rest}} \backslash \{\mathcal{P}\}$ under a null model with a constraint set $\{\mathcal{P}^{\text{sol}}, \{\mathcal{P}\}\}$ and identify patterns $\mathcal{P}' \in \mathcal{P}^{\text{rest}} \backslash \{\mathcal{P}\}$ for which the p-values are greater than α (line $10-11$).

To compute such a p-value, function **SimulateConditionalNullModel** first simulates a null model with a constraint set $\{\mathcal{P}^{\text{sol}}, \{\mathcal{P}\}\}$ (according to the strategy described earlier). Function **ComputeConditionalpValues** (line 11) computes p-values using the same approach as described in Algorithm 4. A pattern $\mathcal{P}$ will be selected if $\mathcal{P}^{\text{sol}} \cup \{\mathcal{P}\}$ explains the largest number of patterns of $\mathcal{P}^{\text{rest}}$ and $\mathcal{P}$ has the smallest number of participating features (line $13 - 15$). Once such a $\mathcal{P}$ is found, it is added to the current $\mathcal{P}^{\text{sol}}$. $\mathcal{P}^{\text{rest}}$ is also updated by removing $\mathcal{P}$ and its explained pattern set from $\mathcal{P}^{\text{rest}}$ (line $20 - 21$). Finding no such $\mathcal{P}$ means that each of the existing patterns of $\mathcal{P}^{\text{rest}}$ only explains itself, and hence the whole set $\mathcal{P}^{\text{rest}}$ is added to $\mathcal{P}^{\text{sol}}$ in this case (line $17 - 18$).

Algorithm 5: Greedy Approximation of Minimal Explaining Pattern Set

```
Input: $\mathcal{SD}$: A spatial data set with $N$ spatial features $f_1, f_2, \cdots f_N$ (each $f_i$ has $n_{f_i}$
    instances). $\mathcal{SC}$: A set of statistically significant co-locations and their co-location
    distances.
Output: $\mathcal{P}^{\text{sol}}$: Approximated minimal pattern set that explains all significant patterns
    in $\mathcal{SC}$.
Method:
 1: $\mathcal{P}^{\text{sol}} \leftarrow \emptyset$
 2: $\mathcal{P}^{\text{rest}} \leftarrow \mathcal{SC}$
 3: while $\mathcal{P}^{\text{rest}} \neq \emptyset$ do
 4:   $CandP \leftarrow \emptyset$
 5:   $nExplnP \leftarrow 0$
      // Select the best $\mathcal{P}$ from $\mathcal{P}^{\text{rest}}$ that explains the highest number of patterns of
      $\mathcal{P}^{\text{rest}}$ and has the minimum number of participating features
 6:   for all $\mathcal{P} \in \mathcal{P}^{\text{rest}}$ do
 7:     if $|\mathcal{P}^{\text{rest}}| = 1$ then
 8:       $CandP \leftarrow \{\mathcal{P}\}$
 9:       break
        // Randomization tests using a conditional null model
10:     SimulateConditionalNullModel($\{\mathcal{P}^{\text{sol}}, \{\mathcal{P}\}\}$)
11:     ComputeConditionalpValues($\mathcal{P}^{\text{rest}} \backslash \{\mathcal{P}\}$)
12:     $ExplnP \leftarrow \{\mathcal{P}' \in \mathcal{P}^{\text{rest}} | (\mathcal{P}^{\text{sol}} \cup \{\mathcal{P}\})$ explains $\mathcal{P}'\}$
13:     if $(|ExplnP| > nExplnP)$ or
        $((|ExplnP| = nExplnP)$ and $(|\mathcal{P}| < |CandP|))$ then
14:       $CandP \leftarrow \{\mathcal{P}\}$
15:       $nExplnP \leftarrow |ExplnP|$
16:   if $CandP = \emptyset$ then
17:     $\mathcal{P}^{\text{sol}} \leftarrow \mathcal{P}^{\text{sol}} \cup \mathcal{P}^{\text{rest}}$
18:     $\mathcal{P}^{\text{rest}} \leftarrow \emptyset$
19:   else
20:     $\mathcal{P}^{\text{sol}} \leftarrow \mathcal{P}^{\text{sol}} \cup CandP$
21:     $\mathcal{P}^{\text{rest}} \leftarrow \mathcal{P}^{\text{rest}} \backslash (CandP \cup ExplnP)$
22: return $\mathcal{P}^{\text{sol}}$
```

Complexity

Algorithm 5 has a run-time cost similar to that of Algorithm 4. The run-time complexity of Algorithm 5 will be of $O((\sum n_{f_i})^2)$ where n_{f_i} is the number of instances of a feature f_i participating in a pattern of $\mathcal{SC}$. Due to the large number of conducted simulations, the total run-time is high and increases with increasing pattern size. However, in many application domains, patterns of large size are not of interest and may not even occur. For instance, to analyse spatial interactions in forestry applications it is sufficient to consider the 3 or 4 nearest neighbors [3, 37]. Our algorithm will also be a good choice for a domain where accuracy is an important concern. Parallelization of our algorithm is also possible due to independence of the conducted simulations. Hence implementation of our algorithm on a parallel and distributed architecture will allow us to handle large sized data and ensure accuracy at the same time.

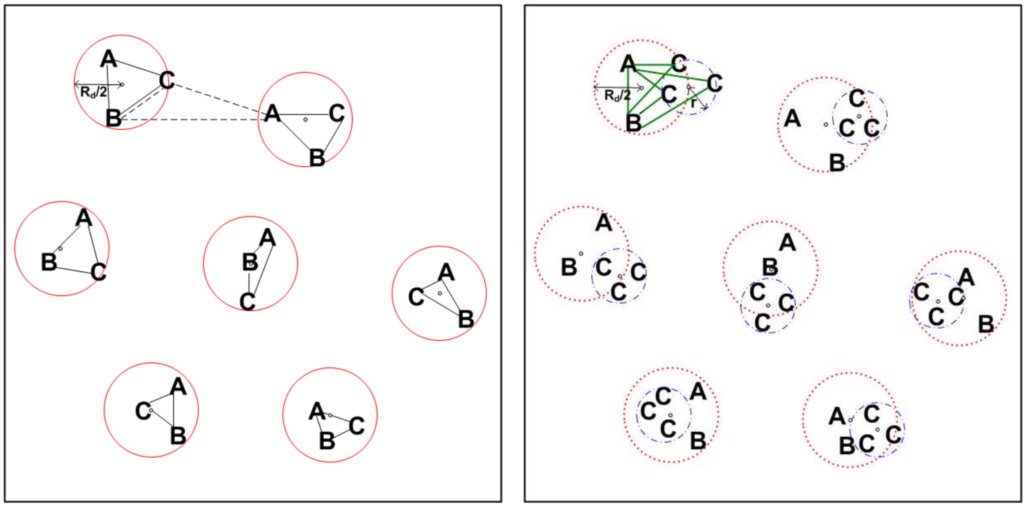
5.4 Experimental Evaluation

5.4.1 Synthetic Data Sets

For the evaluation with synthetic data, we generate data sets with different properties to investigate the effects of auto-correlation, feature abundance, and multi-type interactions. We implant co-locations at different distances in these synthetic data sets and show that in all cases our method can successfully find all the implanted co-locations without requiring any distance threshold information. In addition, our method can determine the correct co-location distance for a true pattern.

Proposed Model for the Simulation of a Co-location

Let $\mathcal{C} = \{f_1 \cdots f_n\}$ be a subset of n features exhibiting a positive interaction in a circular neighborhood of radius $\frac{R_d}{2}$. The true interaction distance for co-location type $\mathcal{C}$ will be R_d. Instances of $\mathcal{C}$ interacting at R_d can be generated using a *Multi-type cluster process* [8]. Let each feature $f_i \in \mathcal{C}$ be non-autocorrelated. To generate N instances of $\mathcal{C}$, N parent points (cluster centers) are first generated using either a Poisson process or an inhibition process. For each parent point c (cluster center), we generate a set of n offspring points, uniformly distributed in a circular neighborhood D_c of radius $\frac{R_d}{2}$ centered at c. We call D_c a *'co-location neighborhood'*. One offspring point in a co-location neighborhood corresponds to



(a) A *multi-type cluster process* used to generate co-location instances

(b) A *multi-type parent-child cluster process* used to generate co-location instances when participating features are auto-correlated

Figure 5.2: a) Generated 7 instances of $\{A, B, C\}$ and co-location neighborhood radius: $\frac{R_d}{2}$. b) Generated 21 instances of $\{A, B, C\}$; feature C is auto-correlated and co-location neighborhood radius: $\frac{R_d}{2} + r$.

an instance of one participating feature of $\mathcal{C}$, and each off-spring point is assigned a feature type using a random mechanism. Here the PID-value of an instance $I_\mathcal{C}$ of $\mathcal{C}$ will be less than or equal to R_d when all participating feature instances belong to one single cluster. Fig. 5.2(a) shows 7 generated instances of co-location type $\{A, B, C\}$ using a multi-type cluster process. Feature instances of such an instance of $\{A, B, C\}$ are shown connected using solid lines. An instance of $\mathcal{C}$ can also be generated using feature instances that belong to different clusters and their PID-values can be greater than R_d. Such an instance of $\{A, B, C\}$ is also shown in Fig. 5.2(a) where the participating feature instances (belonging to different clusters) are shown connected using dashed lines.

When a participating feature exhibits auto-correlation, we generate the instances of the co-location using a *multi-type parent-child cluster process*. First N co-location neighborhoods and their offsprings are generated using the cluster process described above. We call this a *parent cluster process*. For a non auto-correlated feature $f_i \in \mathcal{C}$, one offspring point of a co-location neighborhood D_c corresponds to an instance of f_i, whereas, if a feature $f_j \in \mathcal{C}$ is auto-correlated, we spawn another cluster process, named as a *child cluster process*. In that case one offspring point of a co-location neighborhood D_c works as a cluster center of a cluster c_{f_j} of f_j. Cluster c_{f_j} is defined using two parameters: a radius r and the number of offsprings n' generated in c_{f_j}. These n' offsprings points are uniformly distributed in

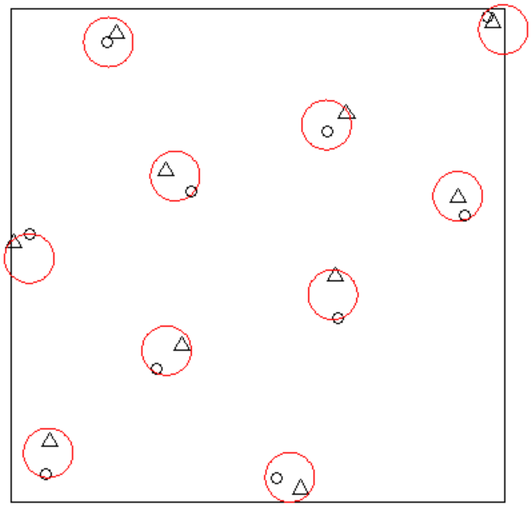


Figure 5.3: A data set with two features $\circ$ and $\triangle$ where a co-location neighborhood is shown using a circle.

c_{f_j} and they will all be assigned the same feature type f_j. All the instances of n features generated from the parent and child cluster processes finally give $N \times n'$ true instances of $\mathcal{C}$. The PID-value of a true instance $I_{\mathcal{C}}$ of $\mathcal{C}$ will be at most $R_d + r$ when the participating feature instances of $I_{\mathcal{C}}$ are generated from one single parent cluster. Fig. 5.2(b) depicts how 21 true instances of $\{A, B, C\}$ are generated from 7 co-location neighborhoods. Feature C shows spatial auto-correlation. The instances of C form clusters and each cluster has 3 instances of C. The figure also shows 3 instances of $\{A, B, C\}$ generated from a co-location neighborhood where co-located instances of A, B, and C are connected by solid lines (in green color).

Pair-wise Interaction

Using the multi-type cluster process described above, we generate a synthetic data set (Fig. 5.3) with an implanted co-location of feature $\circ$ and feature $\triangle$ in a co-location neighborhood of radius 0.05. In a co-location neighborhood, instances of $\circ$ and $\triangle$ are co-located at an average distance of 0.05. However, the maximum possible co-location distance value will be 0.1. Each feature has 10 instances. The study area is a unit square.

Our algorithm finds $\{\circ, \triangle\}$ as significant and reports 0.08 as the co-location distance R_d. At $R_d = 0.08$, the PIC and p-values are 10 and 1, respectively (shown in Table 5.1). $\{\circ, \triangle\}$ is found significant at 6 more distances (shown in the Table A.8 of Appendix A) that are less than the co-location neighborhood diameter 0.1. At those distances, only true instances

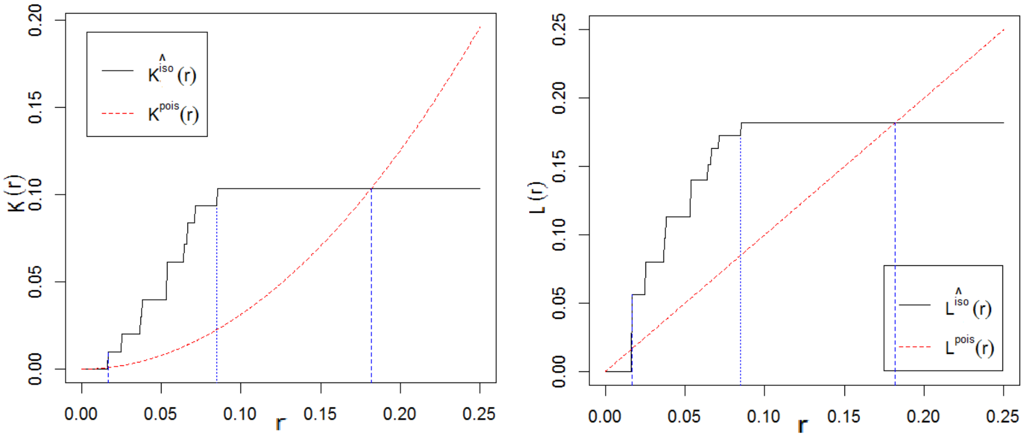
Table 5.1: Pairwise interaction experiment: a reported pattern $\{\circ, \triangle\}$

Pattern	Reported, R_d, p-value	PIC, PI
$\{\circ, \triangle\}$	Yes, 0.08, 0.002	10, 1

of $\{\circ, \triangle\}$ are identified and all participating feature instances of each identified instance of $\{\circ, \triangle\}$ belong to the same co-location neighborhood. The number of identified instances of $\{\circ, \triangle\}$ (PIC-value) at each of those 6 distances is also found statistically significant under a null model. However, only 0.08 is reported as this is the distance among 6 other distances where $\{\circ, \triangle\}$ attains the highest PI value equal to 1. An existing algorithm with a PI-threshold of 0.5 will miss $\{\circ, \triangle\}$ if a distance threshold less than 0.0533 is used (Table A.8 of the Appendix A). Existing algorithms will always report $\{\circ, \triangle\}$ if a value larger than 0.1 is used as a distance threshold. At a very large distance which is higher than 0.1, $\circ$ and $\triangle$ exhibit CSR rather than an aggregation.

We compare our finding on co-location distance with two popular summary statistics that are used in exploratory spatial data analysis. These statistics are quite independent from our method and used as a measure of the clustering tendency among two features. We estimate these statistics at different distances and find a distance interval where the estimated values are deviating positively from the theoretical value, which indicates a clustering behavior. We find that the co-location distance reported from our method falls into that interval and close to the point where the difference of the estimated and the theoretical value is maximum.

First, we estimate the multi-type (type $\circ$ to type $\triangle$) K-function with isotropic edge correction $\hat{K^{\text{iso}}}$. Fig. 5.4(a) shows the plot of the function. We compare the estimated $\hat{K^{\text{iso}}}$-value with the theoretical value at distance r. The theoretical value is computed under the independence assumption and is equal to πr^2. We find that the iso-curve (black solid line) is always above the theoretical (Poisson) curve (red broken line) for a distance interval of $[0.017 - 0.182]$. This interval is shown by two vertical dashed lines (in blue color) in Fig. 5.4(a). Our reported co-location distance 0.08 also lies in this interval. The reported value can also be found on the X co-ordinate by a dotted vertical line (in blue color) of Fig. 5.4(a). ($\hat{K^{\text{iso}}} - \hat{K^{\text{pois}}}$) becomes maximum at $r = 0.085$ and our reported distance is also close to that r-value. The lowest and highest values of the K-function in the shown distance interval are 0.01 and 0.104, respectively. Note that the measures used in our method are the count of pattern instances and the PI-value that are respectively 10 and 1 at dis-



(a) Ripley's multi-type $K_{\{\circ,\triangle\}}$ function

(b) Besag's multi-type $L_{\{\circ,\triangle\}}$ function

Figure 5.4: Pairwise interaction experiment: range of r-values shown by vertical dashed lines where $\hat{K^{\text{iso}}} - K^{\text{pois}} > 0$ and r value at the vertical dotted line is our reported distance.

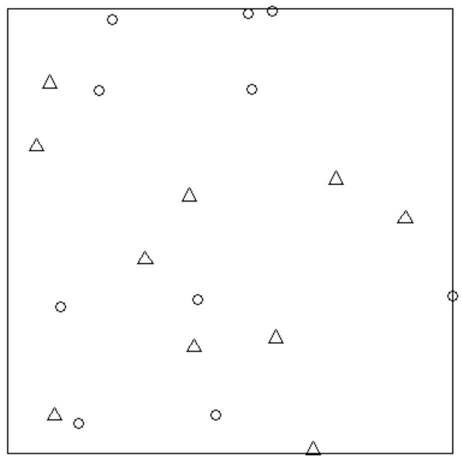
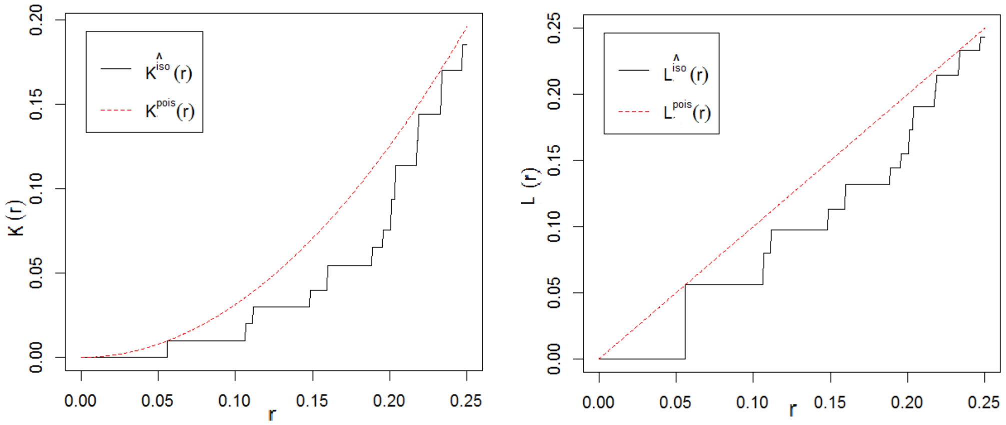


Figure 5.5: Feature $\circ$ and feature $\triangle$ are randomly distributed.

tance 0.08. As a second summary statistic we also estimate the Besag's L-function [20] (Fig. 5.4(b)). A multi-type (type $\circ$ to type $\triangle$) Besag's L-function at a distance r with the isotropic edge correction $\hat{L^{\text{iso}}}$ is computed as $\sqrt{\frac{\hat{K^{\text{iso}}(r)}}{\pi}}$ and compared with its theoretical value at r. The theoretical value at r is equal to r. We find the same distance interval as found by the K-function, where the iso-curve is always above the theoretical curve. However, the computed minimum and the maximum L-function value at that distance interval are respectively 0.056 and 0.182. $r = 0.072$ gives the maximum value of $(\hat{L^{\text{iso}}} - \hat{L^{\text{pois}}})$ and is also not far away from our reported distance.

In another experiment, we generate a data set (Fig. 5.5) where feature $\circ$ and feature $\triangle$ are randomly distributed. The computed PIC-value at any of the unique PID-values is not



(a) Ripley's multi-type $K_{\{\circ,\triangle\}}$ function

(b) Besag's multi-type $L_{\{\circ,\triangle\}}$ function

Figure 5.6: Pair-wise interaction experiment: $\hat{K}^{\text{iso}}$-curve and $\hat{L}^{\text{iso}}$-curve are either below or closely following the theoretical curve.

found as statistically significant; hence $\{\circ, \triangle\}$ is not reported by our algorithm for any of those observed PID-values. We also estimate the multi-type Ripley's K-function and the Besag's L-function. The plot of these two functions are shown in Fig. 5.6(a) and Fig. 5.6(b), respectively. In both cases, the iso-curve is either below or closely following the theoretical (Poisson) curve, indicating no association between feature $\circ$ and feature $\triangle$. In total 73 unique PID values are identified; the first 12 of those together with their PIC and p-values are shown in Table A.9 of the Appendix A. The PI-value at distance 0.201 is 0.5 and does not decrease with further increase of the distance. An existing algorithm with a PI-threshold of 0.5 or smaller will find $\{\circ, \triangle\}$ as *prevalent and report* it if a distance threshold of 0.201 or higher is set. Note that in practice a PI-threshold of 0.5 is not uncommon.

Auto-correlation

We generate a synthetic data set (Fig. 5.7) in a unit square. A positive association between a non-autocorrelated feature $\circ$ and an auto-correlated feature $\triangle$ is implanted using a multi-type parent-child cluster process. The parent cluster radius is 0.05. Feature $\circ$ and feature $\triangle$ have 20 and 60 instances, respectively. Instances of $\triangle$ appear in 20 clusters. For feature $\triangle$, the child cluster radius is 0.025 and each cluster has 3 instances of $\triangle$. Instances of $\circ$ and $\triangle$ that are generated from the same *parent* cluster, can not be more than $0.05 * 2 + 0.025 = 0.125$ units away from each other. Hence at a PID-value of 0.125 or less, we see all feature instances involved in co-locations. At this distance, the PI-value becomes 1 in the generated data set. Table 5.2 gives the distance value at which $\{\circ, \triangle\}$ is reported as

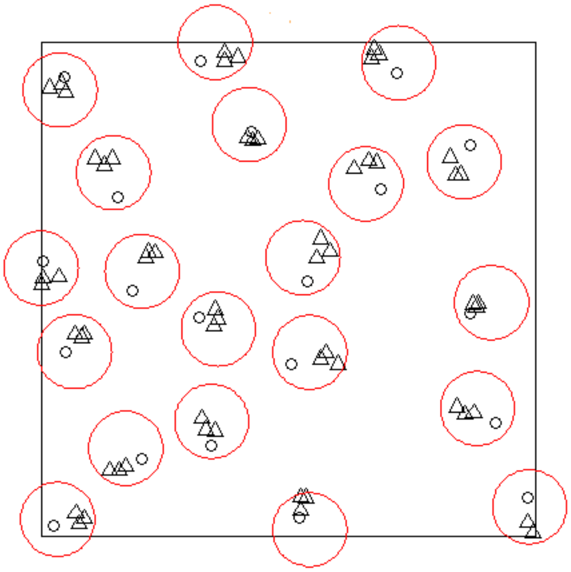


Figure 5.7: A data set with 40 $\circ$s and 50 $\triangle$s where $\circ$ and $\triangle$ are associated.

statistically significant. Table 5.3 shows a few more PID-values at which the observed PIC-values are also found as significant. From these results we can infer that an existing algorithm for a given PI-threshold can miss reporting $\{\circ, \triangle\}$ when the distance threshold is not properly chosen. For instance, for a PI-threshold of 0.55 and a distance threshold smaller than 0.053, $\{\circ, \triangle\}$ will not be found by a traditional co-location mining algorithm.

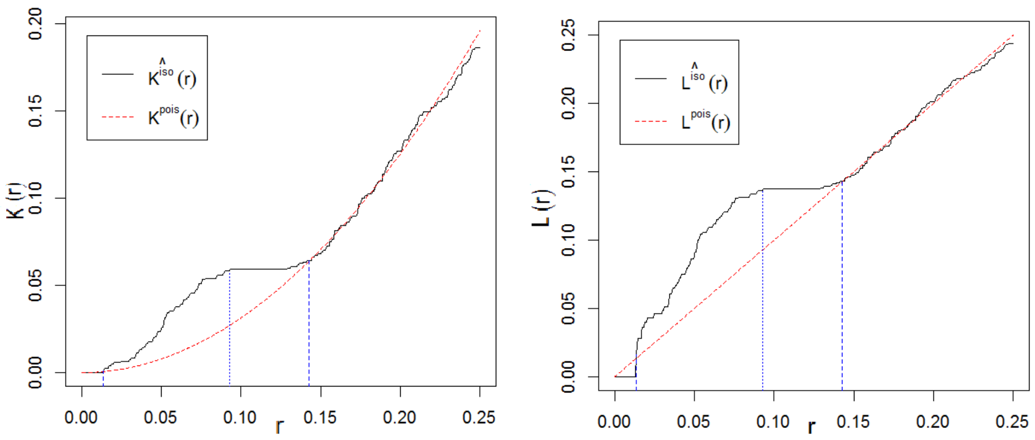
Table 5.2: Auto-correlation experiment: a reported pattern $\{\circ, \triangle\}$

Pattern	Reported, R_d, p-value	PIC, PI
$\{\circ, \triangle\}$	Yes, 0.093, 0.008	60, 1

Table 5.3: Auto-correlation experiment: $\{\circ, \triangle\}$ found significant at multiple distances

PID	0.036	0.044	0.05	0.053	0.062
PIC	15	21	27	33	39
p-value	0.01	0.008	0.004	0.002	0.002
PI-value	0.25	0.35	0.45	0.55	0.65

Fig. 5.8 shows the plot of the estimated multi-type Ripley's K-function and Besag'e L-function with the isotropic edge correction. For both functions, the iso-curve is always above the theoretical curve and monotonically increasing for a distance interval of $[0.014, 0.142]$. The interval is again shown by two vertical dashed lines (in blue color) in



(a) Ripley's multi-type $K_{\{\circ,\triangle\}}$ function

(b) Besag's multi-type $L_{\{\circ,\triangle\}}$ function

Figure 5.8: Auto-correlation experiment: range of increasing r-values shown by vertical dashed lines where $\hat{K^{\text{iso}}} - K^{\text{pois}} > 0$ and r value at the vertical dotted line is our reported distance.

both Fig. 5.8(a) and Fig. 5.8(b). Our reported co-location distance 0.093 also falls in that interval. We also find that $r = 0.08$ gives the maximum values for both $(\hat{K^{\text{iso}}} - \hat{K^{\text{pois}}})$ and $(\hat{L^{\text{iso}}} - \hat{L^{\text{pois}}})$. Our reported distance (0.093) is close to this r-value. In both figures, the reported distance can also be found on the X co-ordinate by a vertical dotted line (in blue color). The minimum and the maximum values of the K-function within that interval are respectively 0.0008 and 0.064. The minimum and the maximum values of the L-function within that interval are respectively 0.016 and 0.143. Our employed measures PIC and PI are 60 and 1, respectively, at the reported distance 0.093.

We generate another synthetic data set (Fig. 5.9) where 40 instances of feature $\circ$ are randomly distributed and feature $\triangle$ with 50 instances is auto-correlated. The auto-correlation of $\triangle$ is modeled using a *Neyman-Scott* cluster process [37]. Instances of $\triangle$ appear in 10 clusters (with a cluster radius of 0.05) and are randomly distributed in each cluster. Each cluster contains 5 $\triangle$s. Although there is no spatial association between $\circ$ and $\triangle$, due to auto-correlation, we observe instances of the pattern $\{\circ, \triangle\}$ even at smaller distances. We compute the p-value of the PIC-value computed at each unique observed PID distance. All the computed p-values are higher than α $(= 0.01)$. Hence $\{\circ, \triangle\}$ is not reported as statistically significant for any of these observed PID distances. The multi-type Ripley's K-function and Besag's L-function also show that these two features do not have any clustering tendency. The plot of these two functions are shown in Fig 5.10(a) and Fig. 5.10(b), respectively. In both cases, the iso-curve is always below the theoretical (Poisson) curve,

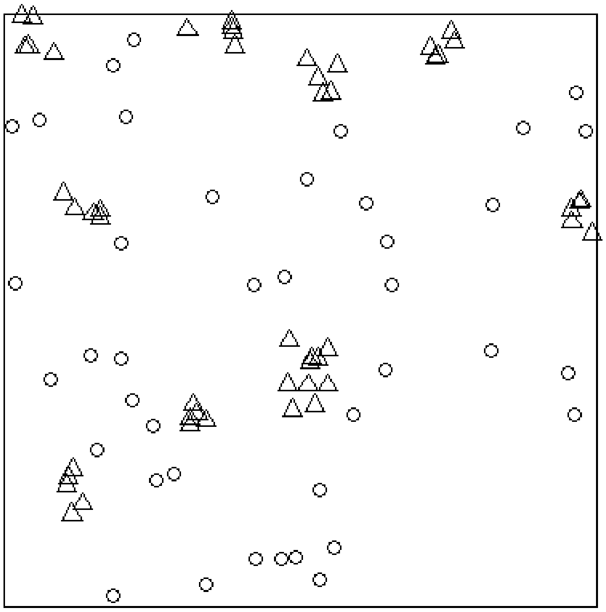


Figure 5.9: A data set with 40 $\circ$s and 50 $\triangle$s with no association between $\circ$ & $\triangle$.

Table 5.4: Auto-correlation experiment: a non-significant pattern $\{\circ, \triangle\}$

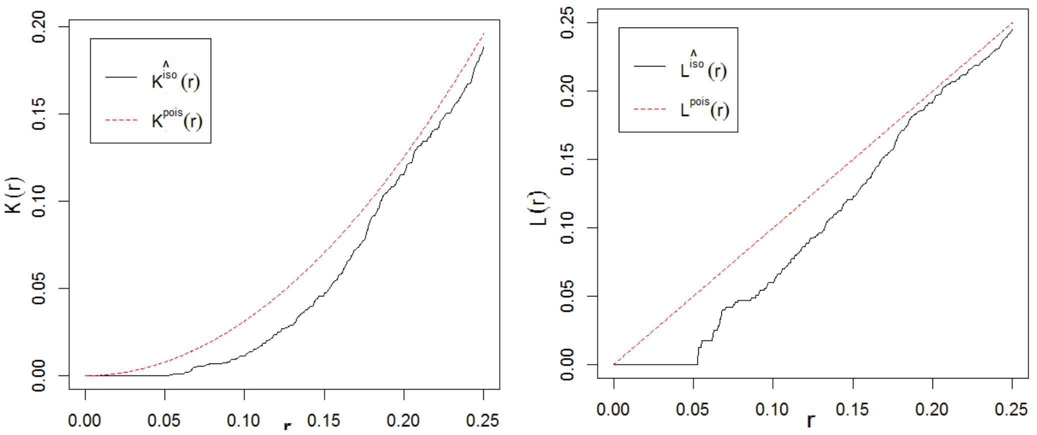
PID	0.108	0.113	0.143	0.153	0.161
PIC	31	38	72	89	104
p-value	0.99	0.988	0.982	0.956	0.948
PI-value	0.25	0.35	0.45	0.55	0.65

indicating no association between feature $\circ$, and feature $\triangle$.

In total 1673 unique PID-values are found and Table 5.4 lists a subset of those. In the table, we see the PI-value increases with the increase of the PID-value. From these values we can infer that $\{\circ, \triangle\}$ can erroneously be reported as prevalent by an existing co-location mining algorithm. Even though there is no true association between $\circ$ and $\triangle$, $\{\circ, \triangle\}$ will still be reported as a prevalent co-location for a PI-threshold of 0.55, when a distance threshold of 0.153 or higher is set.

Multi-type Interaction

A synthetic data set (Fig. 5.11) is generated with 8 different features ($\circ, \triangle, +, \times, \Diamond, \nabla, \boxtimes$, and $*$) exhibiting a variety of spatial interactions in a window of 3 sq. units. 3 different co-location patterns are implanted in the data; details of each co-location type, such as number of instances, spatial distribution of each participating feature, are shown in Table 5.5. All



(a) Ripley's multi-type $K_{\{\circ,\triangle\}}$ function (b) Besag's multi-type $L_{\{\circ,\triangle\}}$ function

Figure 5.10: Auto-correlation experiment: $\hat{K}^{\mathrm{iso}}$-curve and $\hat{L}^{\mathrm{iso}}$-curve are always below the theoretical curve.

Table 5.5: Implanted co-location patterns and distributions of participating features

No	Co-location	Co-location neighborhood radius	Feature: no of instances - individual feature distribution
1	$\{\circ, \triangle, +\}$	0.04	$\circ$: 8, $\triangle$: 8, and $+$: 8 - each non auto-correlated
2	$\{\times, \Diamond\}$	0.06	$\times$: 24 - auto-correlated, $\Diamond$: 6 - non auto-correlated
3	$\{\nabla, \boxtimes\}$	0.08	∇ : 6 and $\boxtimes$: 6 - each non auto-correlated

features, except feature $*$, are involved in some co-location types. Feature $\times$ and feature $*$ are auto-correlated. Feature $\times$ has 24 instances that appear in 6 clusters. The cluster radius is 0.03 and each cluster has 4 instances. Feature $*$ has 12 instances which appear in 4 clusters. The cluster radius is 0.05 and each cluster has 3 instances. The rest of the features are non auto-correlated.

Algorithm 4 (CPMNDC) returns 7 patterns as statistically significant. Table 5.6 shows their co-location distances R_d, their number of instances identified at R_d, their PI-values, and their p-values. The first 6 patterns listed in Table 5.6 are the 3 implanted patterns and their subsets. Their R_d-values are not greater than the diameters of the co-location neighborhood, as expected. Subset $\{\circ, \triangle, +, \nabla, \boxtimes\}$ is also reported as statistically significant due to the presence of the two true co-locations $\{\circ, \triangle, +\}$ and $\{\nabla, \boxtimes\}$. On the other hand, $\{\circ, \triangle\}$, $\{\circ, +\}$, and $\{\triangle, +\}$ are reported due to the presence of the true pattern $\{\circ, \triangle, +\}$. $\{\circ, \triangle, +, \nabla, \boxtimes\}$, $\{\circ, \triangle\}$, $\{\circ, +\}$, and $\{\triangle, +\}$ are all "redundant" patterns which Algorithm 5 is intended to find.

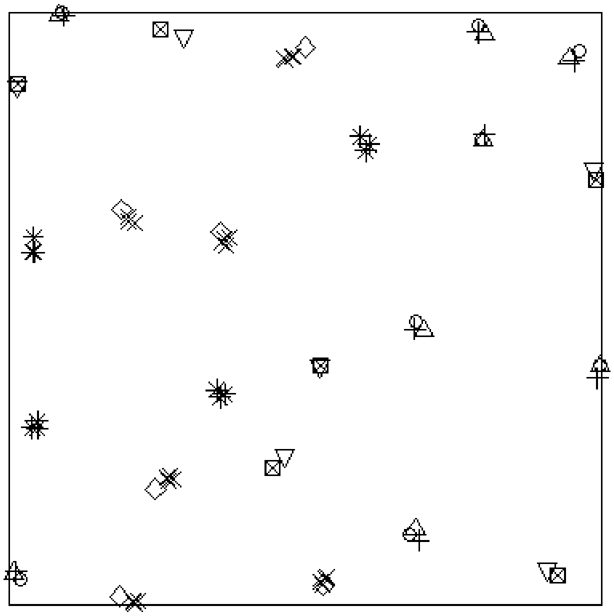


Figure 5.11: A data set of 8 features. Implanted co-locations $\{\circ, \triangle, +\}$, $\{\times, \Diamond\}$, $\{\nabla, \boxtimes\}$.

We apply Algorithm 5 to determine a minimal set of patterns that "explains" all patterns returned by Algorithm 4. In the first pass of the for loop of Algorithm 5, $\{\circ, \triangle\}$, $\{\circ, +\}$, $\{\triangle, +\}$, and $\{\nabla, \boxtimes\}$ explain themselves and one additional pattern $\{\circ, \triangle, +, \nabla, \boxtimes\}$. No additional pattern is explained by $\{\times, \Diamond\}$. However, $\{\circ, \triangle, +\}$ explains the largest number of patterns of $\mathcal{P}^{\text{sol}}$, which are $\{\circ, \triangle\}$, $\{\circ, +\}$, $\{\triangle, +\}$, and $\{\circ, \triangle, +, \nabla, \boxtimes\}$; hence $\{\circ, \triangle, +\}$ is selected for inclusion in $\mathcal{P}^{\text{sol}}$. After the first pass, $\mathcal{P}^{\text{rest}} = \{\{\times, \Diamond\}, \{\nabla, \boxtimes\}\}$. In the second pass, both patterns of the $\mathcal{P}^{\text{rest}}$ are included in the $\mathcal{P}^{\text{sol}}$ as none of them together with the current pattern in $\mathcal{P}^{\text{sol}}$, $\{\circ, \triangle, +\}$, explains any additional pattern. After the second pass, $\mathcal{P}^{\text{sol}} = \{\{\circ, \triangle, +\}, \{\times, \Diamond\}, \{\nabla, \boxtimes\}\}$ and $\mathcal{P}^{\text{rest}} = \emptyset$, and the algorithm stops. The last column of Table 5.6 is the set of patterns that is finally reported by Algorithm 5.

We compute the F-measure of a standard co-location mining algorithm and compare it with the the F-measure of our approach. As our method can detect all true pattern without generating any random pattern, the precision, recall and F-measure are equal to 1. In an attempt to compute the precision, recall, and F-measure, we set 5 different PI-thresholds. For each of these 5 PI-threshold values, we compute the precision, recall, and F measures for each unique pair-wise distance found in the given data. In Table A.10 of the Appendix A, we list the 5 distances that give the top 5 F-measure values for a selected PI-threshold value. From the table, we see that a standard co-location mining algorithm does not achieve an F-measure value as high as our method at any of these distances.

Table 5.6: Multi-type interaction experiment: 3 out of 7 statistically significant patterns are included in the reported minimal explaining co-location pattern set

Statistically significant pattern	p-value	R_d	PIC	PI-value	True pattern
$\{\circ, \triangle\}$	0.002	0.068	8	1	
$\{\circ, +\}$	0.002	0.066	8	1	
$\{\triangle, +\}$	0.002	0.061	8	1	
$\{\times, \Diamond\}$	0.002	0.12	24	1	✓
$\{\nabla, \boxtimes\}$	0.002	0.121	6	1	✓
$\{\circ, \triangle, +\}$	0.002	0.068	8	1	✓
$\{\circ, \triangle, +, \nabla, \boxtimes\}$	0.002	0.667	11	0.5	

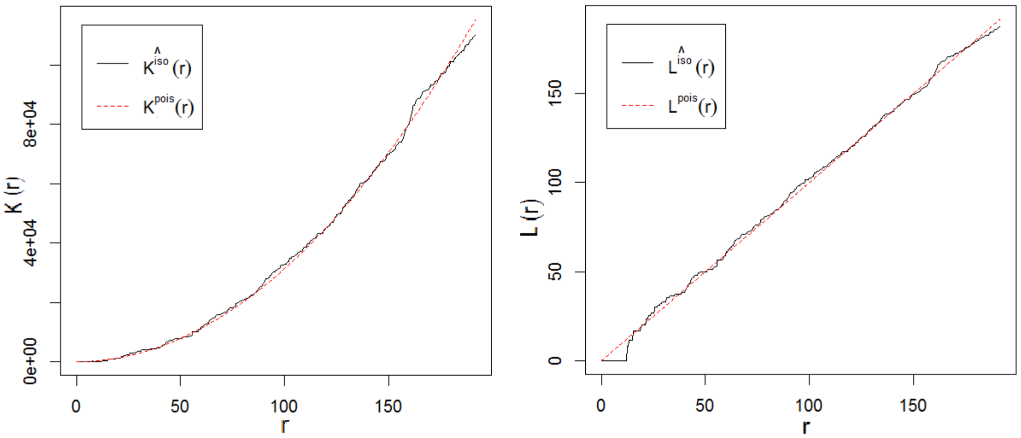
Table 5.7: Ants data: the PIC, p, and PI values at 3 distances (1 unit = $0.5'$)

Distance value→	Smallest	Average	Highest
PID	12.207	304.026	567.62
PIC	1	899	1804
p-value	0.876	0.262	0.436
PI-value	0.0147	0.426	0.426

5.4.2 Real Data Sets

Ants Data

From the nesting behavior, ecologists tried to find an association between the *Cataglyphis wasmanni* ant and the *Messor bicolor* ant, but did not find any association. Fig. 4.19 of Section 4.2.2 shows the Harkness-Isham ants' nests data [31]. Our experiment shows that the SSCSP and sampling algorithms do not find any significant association between these two species for a given distance threshold. Our CPMNDC algorithm allows us to conduct a significance test for multiple distances. We try 1764 unique pair-wise distances that are computed from the ants data and check if the association observed at each of the pair-wise distances becomes significant. None of the PIC-values computed at these distances becomes significant. The multi-type K-function and L-function values with the isometric edge correction are also estimated at different distances. The iso-curves of the K and L functions are shown in Fig 5.12(a) and Fig. 5.12(b), respectively. In both cases, the iso-curve closely follows the theoretical curve, indicating no meaningful association of the Cataglyphis ant and the Messor ant. The PIC-value, p-value, and PI-value computed at the smallest, average, and the highest pair-wise distance values are shown in Table 5.7.



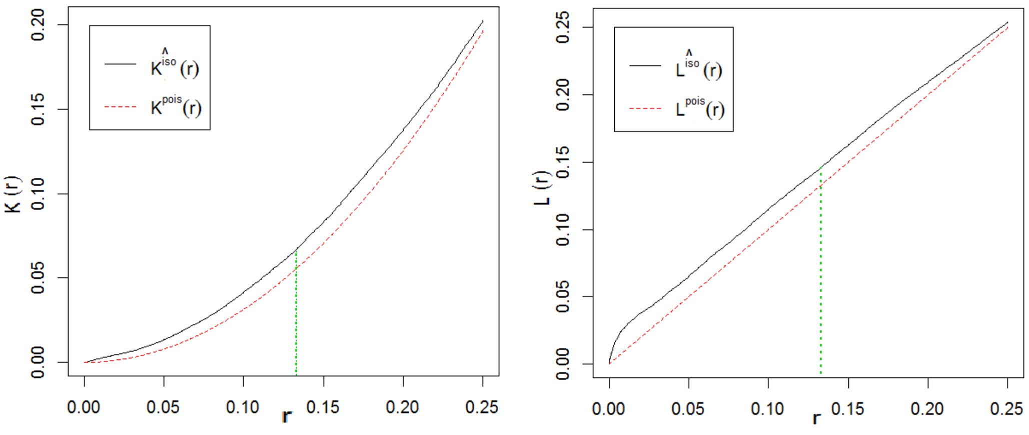
(a) Ripley's multi-type $K_{\{\text{Cataglyphis},\text{Messor}\}}$ function (b) Besag's multi-type $L_{\{\text{Cataglyphis},\text{Messor}\}}$ function

Figure 5.12: Ants data: both $\hat{K^{\text{iso}}}$-curve and $\hat{L^{\text{iso}}}$-curve are closely following their theoretical curves, r (1 unit = $0.5'$).

For instance, the largest pair-wise distance is 567.62 units (1 unit = 0.5 feet). Considering 567.62 as a distance threshold (i.e. PID-value), we find 1804 instances ($= PIC$-value) of $\{$Cataglyphis, Messor$\}$. Under the independence assumption, the probability of the observing 1804 instances at distance 567.62 is computed as 0.436, which is higher than the α-value. Hence, the observed association at distance 567.62 is not significant. The PI-value at this distance is found as 0.426. We also find that the PI-value remains the same ($= 0.426$) for any distance within the interval of $[90.26 - 567.62]$. A standard co-location algorithm using a distance threshold in the above distance range and a PI-threshold of 0.426 or smaller will report $\{$Cataglyphis, Messor$\}$ as a prevalent co-location pattern.

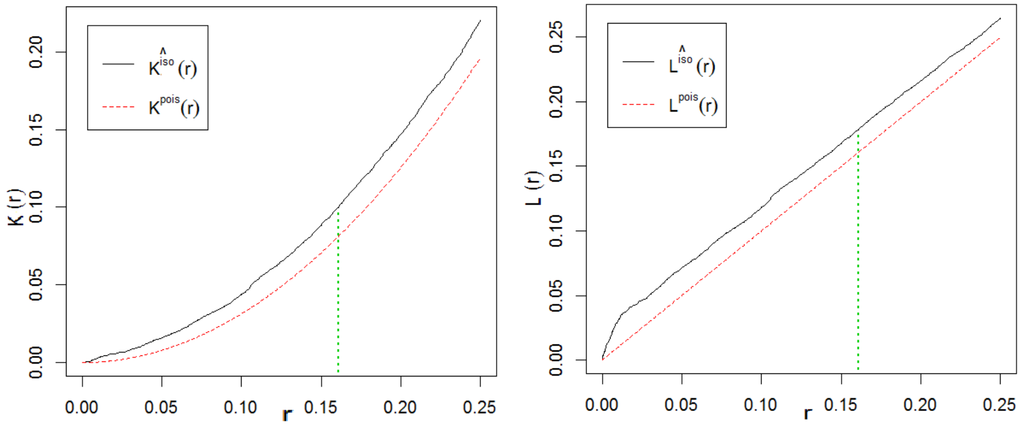
Bramble Canes Data

In the Bramble canes data, there are 3 types of canes. A cane is marked as either 1, or 2, or 3 according to its age in number of years. Mark 1 has 359 instance, mark 2 has 385 instances and mark 3 has 79 instances. The locations of the canes are recorded from a 9m square plot, which is scaled down to a unit square. From the Bramble canes data, our algorithm finds 3 significant patterns. Table 5.8 shows the reported patterns and their co-location distances. The PIC-value, p-value, and the PI-value of each reported co-location distance are also shown. For instance, for a statistically significant pattern $\{1, 2\}$, the reported co-location distance is 0.133. At that distance, 8514 instances of $\{1, 2\}$ are identified. To find the co-location distance of $\{1, 2\}$, our algorithm performs the statistical significance test at 83638



(a) Ripley's multi-type $K_{\{1,2\}}$ function (b) Besag's multi-type $L_{\{1,2\}}$ function

Figure 5.13: Bramble canes data: K and L functions of pattern $\{1, 2\}$, r (1 unit = 9 m).

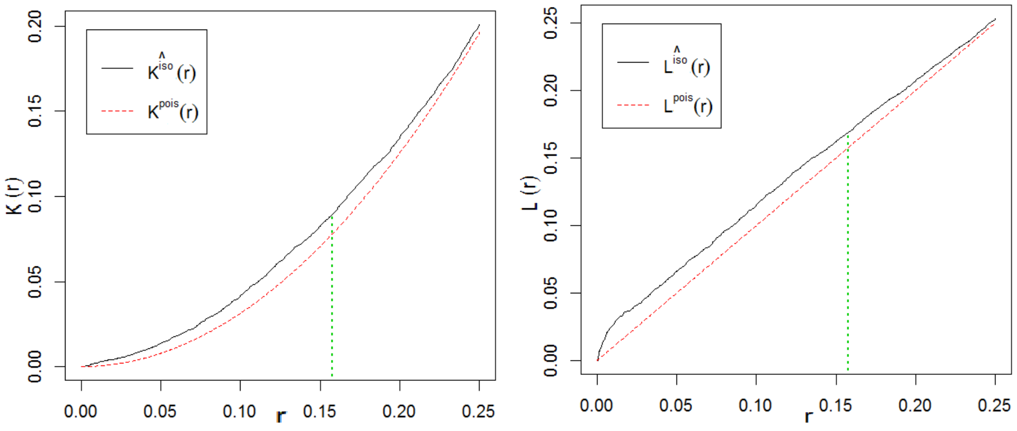


(a) Ripley's multi-type $K_{\{1,3\}}$ function (b) Besag's multi-type $L_{\{1,3\}}$ function

Figure 5.14: Bramble canes data: K and L functions of pattern $\{1, 3\}$, r (1 unit = 9 m).

unique PID distances and finds 108878 instances of $\{1, 2\}$ in total. The K-function and L-function with the isotropic edge correction are shown in Fig. 5.13(a) and Fig. 5.13(b), respectively. The vertical dotted line (in green color) in these two figures shows the reported co-location distance of pattern $\{1, 2\}$ on the X co-ordinate. Similar results of pattern $\{1, 3\}$ and pattern $\{2, 3\}$ are also shown in Fig. 5.14 and Fig. 5.15. In these figures, at our reported distances, the estimated K-function value and the estimated L-function value are higher than the theoretical values, indicating a clustering behavior of the participating features of each reported pattern.

For K-function, we find r-value at which a maximum deviation of the estimated value and theoretical value occurs. The computed r-values are shown in Table 5.9. From this table we find that the reported co-location distance for each of the detected patterns $\{1,2\}$,$\{1,3\}$, and $\{2,3\}$, is not close to the r-value giving the maximum value of $(\hat{K^{\text{iso}}} - \hat{K^{\text{pois}}})$. In the Bramble canes data, the expected number instances of mark 2 around a randomly chosen instance of mark 1 increase with the increase of r. Hence the $K_{\{1,2\}}$-value increases with the increase of r. Our approach reports the distance at which the maximum number of instances of mark 1 and mark 2 become co-located and the number of observed instances of $\{1,2\}$ at that distance becomes statistically significant. Our reported distance for $\{1,2\}$ is 0.133. At that distance each instance of mark 1 finds at least one instance of mark 2 in its circular neighborhood resulting an instance of $\{1,2\}$ and the PI-value of $\{1,2\}$ becomes 1. If the distance (radius of the circular neighborhood of mark 1) increases further, more instances of mark 2 around an instance of mark 1 are observed. In such a case, the PI-value will not increase, but the $K_{\{1,2\}}$-value increases. For this reason the r-value giving maximum value of $(\hat{K^{\text{iso}}} - \hat{K^{\text{pois}}})$ does not match with our reported distances. K-function is designed to measure the clustering behavior of the points of a point pattern. However, this function, by construction, is not the exact measure of a co-location property. It successfully identifies a co-location tendency of spatial features in cases which are observed in our experiments with synthetic data sets. However, in some cases, it mistakenly reports a co-location behavior among features even though the features do not exhibit a true co-location. The weakness of K-function as a co-location measure is also discussed in Section 2.1.2 of Chapter 2.



(a) Ripley's multi-type $K_{\{2,3\}}$ function (b) Besag's multi-type $L_{\{2,3\}}$ function

Figure 5.15: Bramble canes data: K and L functions of pattern $\{2,3\}$, r (1 unit = 9 m).

Table 5.8: Bramble canes data: reported patterns (1 unit = 9 m)

Reported	Reported co-location distances & measures				No. of	No. of
pattern	R_d	PIC	p-value	PI	PID	instances
$\{1,2\}$	0.133	8514	0.002	1	83638	108878
$\{1,3\}$	0.161	2533	0.002	1	21103	22387
$\{2,3\}$	0.158	2462	0.002	1	8187	8737
$\{1,2,3\}$	0.161	59903	0.002	1	98626	2355447

Table 5.9: Bramble canes data: r-values giving $\max(\widehat{K^{\text{iso}}} - \widehat{K^{\text{pois}}})$ (1 unit = 9 m)

Reported	Reported co-location	r-value of
pattern	distances	$\max(\widehat{K^{\text{iso}}} - \widehat{K^{\text{pois}}})$
$\{1,2\}$	0.133	0.182
$\{1,3\}$	0.161	0.249
$\{2,3\}$	0.158	0.133

Lansing Woods Data

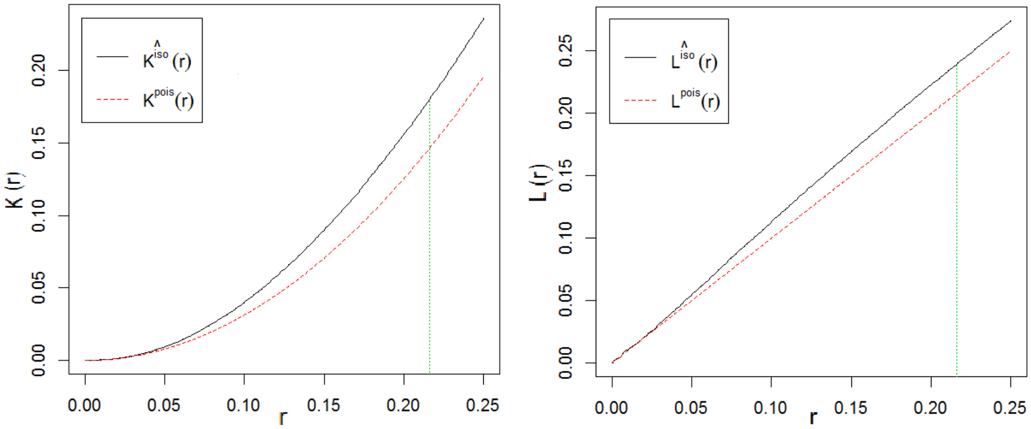
The Lansing Woods multi-type point data records the locations of 6 different tree species from an area of 924 ft × 924 ft (19.6 acres). The locations of 135 black oaks, 703 hickories, 514 maples, 105 red oaks, 346 white oaks, and 448 miscellaneous trees are stored in this data. The data set is shown in Fig. 4.22. In the previous chapter, our SSCSP algorithm has analysed this data. For an interaction distance threshold of 92.4 ft, SSCSP algorithm finds 4 statistically significant co-location patterns (Table B.3 of the Appendix B). These are {Black oak, Hickory}, {Maple, Misc.}, {Black oak, Hickory, Red oak}, and {Maple, Misc., Red oak}. Using algorithm CPMNDC, we search significant patterns occurring at different distances. Algorithm CPMNDC finds the same subsets as statistically significant co-locations. We also apply approximation algorithm 5 to find redundant patterns from the result of CPMNDC algorithm. Approximation algorithm 5 finds {Black oak, Hickory, Red oak}, and {Maple, Misc., Red oak} as redundant patterns. This findings can also be validated from the pair-wise interactions of Red oak with Black oak, hickory, and misc. trees as shown in Table 4.3. These two patterns are respectively explained by {Black oak, Hickory} and {Maple, Misc.}. Finally, {Black oak, Hickory} and {Maple, Misc.} are reported as the true co-locations from the Lansing Woods data set. Their co-location distances are also determined and a detailed result of each reported co-location is shown in Table 5.10. For instance, {Black oak, Hickory} is reported as significant at distance 0.1803802 (1 unit =

Table 5.10: Lansing Woods data: reported patterns (1 unit = 924′)

Reported pattern	Reported values				No. of	No. of
	R_d	PIC	p-value	PI	PID	instances
{Black oak, Hickory}	0.1803802	9447	0.002	1	12309	14419
{Maple, Misc.}	0.2983035	13282	0.002	1	15609	17486

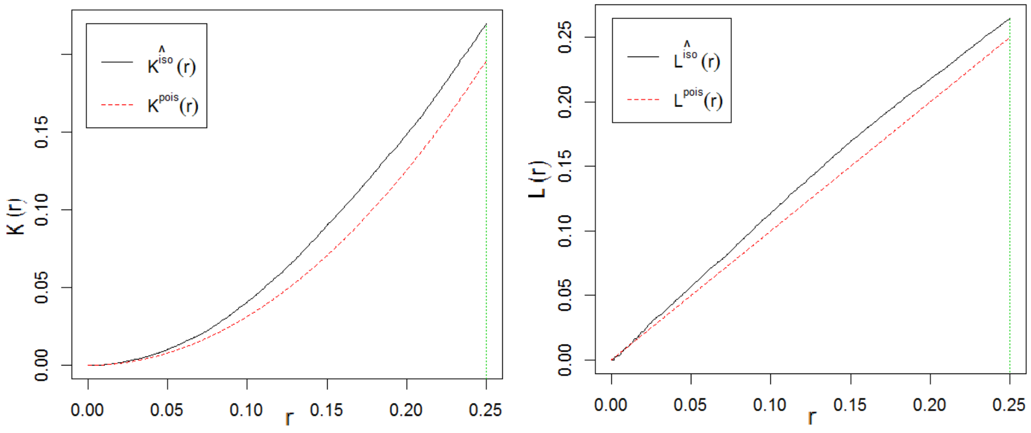
924 feet). The PIC and PI-values at that distance are 9447 and 1, respectively. In total, 12309 unique PID distances are considered and 14419 instances of {Black oak, Hickory} are identified.

The K-function and L-function with the isotropic edge correction are shown in Fig. 5.16(a) and Fig. 5.16(b), respectively. The vertical dotted line (in green color) in these two figures shows the reported co-location distance of pattern {Black oak, Hickory} on the X co-ordinate. Similar results of pattern {Maple, Misc.} are also shown in Fig. 5.17. For K-function, we find r-value at which a maximum deviation of the estimated value and theoretical value occurs. The computed r-values are shown in Table 5.11. From this table we find that the reported co-location distance for each of the patterns {Black oak, Hickory} and {Maple, Misc.} is not far from the r-value giving the maximum value of $(\hat{K}^{\text{iso}} - \hat{K}^{\text{pois}})$.



(a) Ripley's $K_{\{Black\ oak,Hickory\}}$ function (b) Besag's $L_{\{Black\ oak,Hickory\}}$ function

Figure 5.16: Lansing woods data: K and L functions of {Black oak, Hickory}, r (1 unit = 924′).



(a) Ripley's $K_{\{Maple,Misc.\}}$ function

(b) Besag's $L_{\{Maple,Misc.\}}$ function

Figure 5.17: Lansing woods data: K and L functions of {Maple, Misc.}, r (1 unit = 924′).

Table 5.11: Lansing woods data: r-values giving max$(\hat{K}^{\text{iso}} - \hat{K}^{\text{pois}})$ (1 unit = 924′)

Reported pattern	Reported co-location distances	r-value of max$(\hat{K}^{\text{iso}} - \hat{K}^{\text{pois}})$
{Black oak, Hickory}	0.1803802	0.2363503
{Maple, Misc.}	0.2983035	0.234375

5.5 Summary

The current co-location mining algorithms require a distance threshold. In this chapter, we have discussed how a co-location mining algorithm using a distance threshold may fail mining true patterns when patterns occur at multiple distances. To solve this limitation, here we have proposed a solution which can mine true co-locations without using any threshold parameters from the user end. Our mining approach is based on a statistical test. We also have proposed an approximation algorithm to prune redundant patterns that may occur in a statistical test. We have validated our approach using synthetic and real data sets. The experimental results show that our method has found all the true patterns that were implanted in the synthetic data sets. Our findings from the tested ecological data sets can be an important feedback for the domain scientists in their analysis.

Chapter 6

Conclusions and Future Directions

With the rapid growth of spatial data, efforts are being given in developing theories and techniques which can analyse massive and complex spatial data sets and acquire new insights and implicit knowledge embedded in the data. Spatial data mining aims to find interesting patterns from spatial data. In this thesis, we studied *spatial interaction patterns*, that occur due to the *spatial dependencies* of features. Spatial interaction patterns are important in many spatial domains such as ecology, forestry, urban planning, and environmental science.

In mining prevalent patterns the current approaches require two predefined threshold parameters - one is for the used prevalence measure and the other is for the interaction distance. Finding appropriate values for these thresholds is not easy in many spatial domains and one single threshold value may not work to mine true patterns of different sizes. Interaction among features occur at different spatial levels; hence the existing approach of using one single distance threshold may fail to find all true patterns at different spatial distances. An arbitrary selection of prevalence or interaction distance thresholds may report random subsets of features as prevalent. In this thesis we aimed to resolve the above limitations of the current approaches and our objective was: "*finding a threshold free approach which can mine statistically sound interaction patterns from spatial data*". To this end, we had to consider the following key issues:

- designing an appropriate null model for a statistical test,
- selecting an appropriate test statistic which will be a numerical summary of the spatial dependencies measured from the data,
- reducing the total computational cost of the statistical simulations conducted for estimating the distribution of a test statistic,

- solving a redundancy problem that may occur when using statistical tests,
- generating appropriate synthetic data sets to validate our proposed approaches. This requires developing data generation models which can simulate interactions among multi-type features and which can also simulate spatial distributions of individual features.

6.1 Key Results

For a statistical significance test, we designed an appropriate null model which preserves the observed spatial distribution of each individual feature of the data. To this end, spatial auto-correlation of a feature is taken into account and a cluster process is used to model a spatial auto-correlation. We used three different test statistics. The PI measure is used as a prevalence measure in the current literature. In our work, we use the PI as a test statistic and this worked successfully for our approach. We also proposed another test statistic which is an approximation value of the PI. This new statistic is computationally more efficient to compute and leads in general to correct statistical inferences for interaction patterns. The third test statistic is the pattern instance count, which is used to find true co-locations at different distances.

We proposed two approaches that can mine statistically significant interaction patterns for a given interaction distance. In our first approach named SSCSP, the PI is used as a test statistic. Two strategies are proposed to reduce the cost during the data generation and PI computation steps of a simulation. In the second approach, we proposed a grid based sampling approach to compute an approximated PI-value used as a test statistic. Lastly, we conducted a broad set of experiments and analysed evaluating the effectiveness of our approach in finding both co-location patterns as well as segregation patterns using a variety of synthetic data sets, generated using popular spatial point process models. We also used ecological, forestry, and urban data sets to validate our approaches. We demonstrated the efficacy of our strategies adopted for the data generation and prevalence measure computation steps. The experimental results show that the runtime of a naïve implementation can be improved significantly by adopting those strategies. Particularly, the sampling approach appears to be quite robust, as it is found in our experiments with a large variety of real and synthetic data sets. In all of our experiments, we did not miss any significant pattern even using the coarsest grid, except for a single pattern involving a feature with an extremely low number of instances. Finally, the sampling approach provides the highest runtime improvement compared to our all-instances-based and naïve approaches.

As the second contribution of this thesis, we proposed an approach to mine true co-locations without prevalence and interaction distance thresholds. This approach can find the co-location distance of a true pattern. In this context we also solved the redundancy problem that could occur when the null hypothesis simply assumes the independence of features. For redundancy checking, we proposed a *constrained null model* for the statistical test and also proposed a heuristic to simulate such a null model. Finally, we proposed models to generate synthetic data sets to evaluate our approach. To the best of our knowledge, there is no model in the literature that can simulate a co-location of more than two features. We proposed a model to simulate a co-location where a participating feature can either be auto-correlated or non auto-correlated.

6.2 Future Research Directions

There are several directions for future research:

- **Design of a better sampling approach:** We would like to investigate how the number of instances and spatial distribution of the participating features should affect the selection of the grid resolution for our sampling approach. It is possible to design a "mixed" approach in which grid cells of different sizes and even the full circular neighborhood are used for different features, depending on the number of its instances. We also plan to investigate other sampling approaches such as sampling based on randomly selected regions, Latin hypercube sampling [43], and the space filling curve technique [54] and find if these approaches can offer a better computational efficiency compared to our sampling approach.
- **Design of a null model for improved detection accuracy:** In the thesis, our first null model used in a statistical test assumes the independence of all features. This model suffered from a redundancy problem. Later we modified this null model and imposed a set of constraints on the null model. To improve the detection accuracy, we would like to investigate null models that are proposed in the literature and compare their computational efficiency in our mining framework. Roxburgh *et al.* in [53] discuss two null models, which are patch model and random shifts model and validate these null models statistically for spatial association pattern detection for pairs of fea-

tures. In a random shifts model, a set of data points are first generated in a simulation. These data points are then rotated and shifted to produce data points for subsequent simulations. We would like to investigate if such a data generation approach can be computationally more efficient than our current data generation followed in the randomization tests. Gionis *et al.* also proposed a *swap randomization* technique [26] for assessing data mining results on Binary data sets. Their approach is found more efficient than existing randomization methods and detects expected patterns from retail data sets. We would like to investigate the performance of a swap randomization technique in estimating the distribution of the test statistic used in our data mining approach. Future work may also include developing data permutation techniques for a null model which addresses auto-correlation and true interactions (to mitigate redundancy issues). Such techniques will be compared with our current approaches in terms of computational gain and pattern detection accuracy.

Our proposed statistical framework has not considered the edge effects [20] while computing the test statistic. A preliminary solution along this line is that we can consider only points whose circular neighborhoods are completely inside the given study area. These points will then be used to compute the test statistic for both the observed data and the data sets generated under our null hypothesis. If the number of data points considered to compute the test statistic is too low, our statistical model will still be able to make the correct statistical inference about an observed interaction. However our approach may suffer by the edge effects in the case where features have fewer instances. We would like to investigate the sensitivity of our approach for such a case and also improve our method by taking edge effects into account for the test statistic computation.

- **Design of new test statistics:** The current literature in association rule mining proposes several interest measures for identifying and ranking detected patterns according to their potential interest. [23] gives a list of these measures. In the current literature of co-location mining, only two prevalence measures, PI and $maxPI$, are proposed. Our future work includes investigating new prevalence measures that are suitable as test statistics and could also allow additional pruning techniques in our framework.

  We also would like to investigate the applicability of other statistical tests on the detection of spatial interaction patterns. Some would prefer estimating a confidence interval of the test statistic (such as PI) value instead of computing a p-value. A significance test based on the p-value is a Boolean test. On the other hand, a confidence interval of PI gives information on the accuracy of an estimated PI value with a certain probability. Then constructing a test hypothesis, efficiency of the interval estimation, and pattern detection accuracy may appear as the main issues.

- **Improving the computational efficiency of the CPMNDC algorithm:** Due to the statistical simulations conducted at multiple distances, the CPMNDC approach is computationally expensive. For many application domains such as cell biology and forestry, where the co-location size is not very large and accuracy is important, our approach can be a good choice. However, to extend its applicability we aim to improve the computational efficiency of CPMNDC along the following lines: (1) a simple spatial index structure can help to reduce the cost of the pair-wise distance computation of the algorithm; (2) the conducted simulations of the randomization tests are independent of each other. Hence, a simple parallelization of the randomization tests is possible. Parallelization of clique finding computation for different pair-wise distances can also be done.

- **Experimental evaluation:** Currently, we could not verify our co-location distance results found from the used real data sets because of ground truth. Ecologists identified spatial associations from these data sets but did not report any result on the actual interaction distance. In the future, we would like to evaluate CPMNDC with real data sets that have ground truth about interaction distances. Evaluation of our proposed approach using good synthetic data sets is also important.

  Our experiments with synthetic data sets overlook the effects of spatial heterogeneity. To generate a random point pattern for a feature, we use a homogenous Poisson process with a constant intensity. For an auto-correlated feature, cluster centers are similarly generated from a homogenous Poisson process with a constant intensity. Instances of a feature may show spatial heterogeneity, in which case a Poisson process with a variable intensity can be followed to generate them. Here the intensity will be described as a function of x and y. The effect of spatial heterogeneity will further be investigated in the future.

- **Interaction pattern mining from Spatiotemporal databases:** In the future, we would like to extend our statistical model to find interaction patterns in ST domains.

Methods such as [15] are proposed where the temporal dimension is discretized; then for each discretized value of time, a snapshot from the observation data is prepared for analysis. Each snapshot now becomes a spatial data set. Prevalent patterns found from each of these snapshots are tested if they are prevalent across different snapshots taken over time. Many of the proposed methods are threshold based and do not test the significance of their found patterns statistically. One major challenge is constructing the null model for a statistical test. The null model for each different snapshot of time may not be the same when features do not maintain the same spatial distribution across times and also show variation in their distribution over time. The distribution of a feature will now be more complex as it will be described in terms of three parameters (x, y, and t). Auto-correlation behavior along space as well as along time will further complicate the model in characterizing the distribution of a feature in a ST domain. Minimizing the computational cost of finding spatial interaction patterns from a ST domain will also be a challenge.

- **Mining statistically sound complex patterns:** We would like to extend our framework to mining statistically significant complex patterns as discussed in [5]. This is a trivial extension of our current framework and the same test statistics and the null hypothesis that are proposed in this thesis can be used. Achieving computational efficiency will be the main challenge for this problem. The number of statistical tests increases since the number of candidate patterns in the case of complex pattern mining is larger than that in co-location or segregation pattern mining. We see an increase in the amount of computation needed to identify the instances of different complex patterns. However, computational efficiency can be achieved by finding a pruning property to identify candidate patterns for which the computation of the test statistic in a simulation is unnecessary.

- **Mining co-location patterns from Boolean and quantitative spatial features:** Our current mining approach does not take into account quantitative spatial features which may generate different types of interactions. For instance, in a forest, an interaction of a group of tree species may be affected by the quantitative spatial features such as the intensity of sunlight, amount of minerals, and rainfall. Co-locations of these trees could be different from place to place depending on the presence of the quantitative spatial features. Our next goal is to develop a general mining approach to find meaningful co-locations of Boolean and quantitative spatial features. Such co-location patterns give more meaningful insights for a domain scientist and a better understanding on mechanisms of the underlying processes.

In finding co-locations in the presence of quantitative spatial features, we can discretize the values of a quantitative spatial feature into intervals and find co-locations of these intervals of a quantitative spatial feature with other spatial features. In creating the number of intervals, several issues such as “required execution time” or “generation of many co-locations” may arise. Similar issues are discussed and solved in quantitative association rule mining [61]. A genetic algorithm proposed in [55] dynamically discovers good intervals by optimizing the support and confidence. We would like to investigate if the discretization techniques proposed to create intervals for a quantitative attribute of a relational table can be adapted to discretize the values of a quantitative spatial feature and design a quantitative co-location pattern mining approach.

Bibliography

[1] Aibek Adilmagambetov, Osmar R. Zaïane, and Alvaro Osornio-Vargas. Discovering Co-location Patterns in Datasets with Extended Spatial Objects. In *Proc. of the* 15*th Int'l Conference on Data Warehousing and Knowledge Discovery*, pages 84–96, 2013.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proc. of the* 20*th Int'l Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] Oscar Aguirrea, Gangying Huib, Klaus von Gadowb, and Javier Jiménez. An Analysis of Spatial Forest Structure Using Neighbourhood-Based Variables. *Forest Ecol. Manag.*, 183(1–3):137–145, 2003.

[4] Maria-Luiza Antonie and Osmar R. Zaïane. Mining Positive and Negative Association Rules: An Approach for Confined Rules. In *Proc. of the* 8*th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 27–38, 2004.

[5] Bavani Arunasalam, Sanjay Chawla, and Pei Sun. Striking Two Birds With One Stone: Simultaneous Mining of Positive and Negative Spatial Patterns. In *Proc. of the* 5*th SIAM Int'l Conference on Data Mining*, pages 173–182, 2005.

[6] Adrian J. Baddeley. Spatial Point Processes and their Applications. In *Lecture Notes in Mathematics: Stochastic Geometry*. Springer Verlag, Berlin Heidelberg, 2007.

[7] Adrian J. Baddeley. Analysing Spatial Point Patterns in R. Workshop Notes, Version 3, 2008.

[8] Adrian J. Baddeley. Multivariate and marked point processes. In Alan E. Gelfand, Peter J. Diggle, Peter Guttorp, and Montserrat Fuentes, editors, *Handbook of Spatial Statistics*, chapter 21, pages 371–402. Chapman and Hall / CRC, 2010.

[9] Sajib Barua and Jörg Sander. SSCP: Mining Statistically Significant Co-location Patterns. In *Proc. of the* 12*th Int'l Symposium on Advances in Spatial and Temporal Databases*, pages 2–20, 2011.

[10] Sajib Barua and Jörg Sander. Mining Statistically Significant Co-location and Segregation Patterns. *IEEE Trans. Knowl. Data Eng.*, 99(PrePrints):1, 2013.

[11] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. of the Royal Statist. Society. Series B*, 57(1):289–300, 1995.

[12] Julian Besag and Peter J. Diggle. Simple Monte Carlo Tests for Spatial Patterns. *Appl. Statist.*, 26(3):327–333, 1977.

[13] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond Market Baskets: Generalizing Association Rules to Correlations. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, pages 265–276, 1997.

[14] Coen Bron and Joep Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph. *Communications of the ACM*, 16(9):575–577, 1973.

[15] Mete Celik, Shashi Shekhar, James P. Rogers, and James A. Shine. Mixed-Drove Spatiotemporal Co-occurence Pattern Mining. *IEEE Trans. Knowl. Data Eng.*, 20(10):1322–1335, 2008.

[16] Noel A. C. Cressie. *Statistics for Spatial Data*. Wiley, 1993.

[17] Mark R. T. Dale. *Spatial Pattern Analysis in Plant Ecology*. Cambridge University Press, 2000.

[18] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, New York, Inc., Secaucus, NJ, 1997.

[19] Peter J. Diggle. Displaced Amacrine Cells in the Retina of a Rabbit : Analysis of a Bivariate Spatial Point Pattern. *J. of Neuroscience Methods*, 18(1–2):115–25, 1986.

[20] Peter J. Diggle. *Statistical Analysis of Spatial Point Patterns (2nd edn.)*. Arnold, London, UK, 2003.

[21] Peter J. Diggle and Richard J. Gratton. Monte Carlo Methods of Inference for Implicit Statistical Models. *J. of the Royal Statist. Society, Series B*, 46(2):193–227, 1984.

[22] Marie-Josée Fortin and Geoffrey M. Jacquez. Randomization Tests and Spatially Auto-Correlated Data. *Bulletin of the Ecological Society of America*, 81(3):201–205, 2000.

[23] Liqiang Geng and Howard J. Hamilton. Interestingness Measures for Data Mining: A Survey. *ACM Comput. Surv.*, 38(3), 2006.

[24] Douglas J. Gerrard. *Competition Quotient: A New Measure of the Competition Affecting Individual Forest Trees*. Research Bulletin 20, Agricultural Experiment Station, Michigan State University, 1969.

[25] Charlie J. Geyer. Likelihood Inference for Spatial Point Processes. In O. E. Barndorff-Nielsen, W. S. Kendall, and M.N.M. Van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, number 80 in Monographs on Statistics and Applied Probability #80, chapter 3, pages 79–140. Chapman and Hall / CRC, Monographs on Statistics and Applied Probability, 1999.

[26] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. *ACM Trans. Knowl. Discov. Data*, 1(3), 2007.

[27] Phillip Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Verlag, 1994.

[28] Luciana C. Gusmao and Marymegan Daly. Evolution of Sea Anemones (Cnidaria: Actiniaria: Hormathiidae) Symbiotic With Hermit Crabs. *Molecular Phylogenetics and Evolution*, 56(3):868–877, 2010.

[29] Jaiwei Han, Krzysztof Koperski, and Nebojsa Stefanovic. GeoMiner: a System Prototype for Spatial Data Mining. *ACM SIGMOD Record*, 26(2):553–556, 1997.

[30] Sami Hanhijärvi. Multiple Hypothesis Testing in Pattern Discovery. In *Proc. of the* 14*th Int'l Conference on Discovery science*, pages 122–134, 2011.

[31] R. D. Harkness and Valerie Isham. A Bivariate Spatial Point Pattern of Ants' Nests. *J. of the Royal Statist. Society, Series C (Appl. Statist.)*, 32(3):293–303, 1983.

[32] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *J. of the American Statist. Association*, 58(301):13–30, 1963.

[33] Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scand. J. of Statist.*, 6:65–70, 1979.

[34] Yan Huang, Jian Pei, and Hui Xiong. Mining Co-Location Patterns with Rare Events from Spatial Data Sets. *GeoInformatica*, 10(3):239–260, 2006.

[35] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering Co-location Patterns from Spatial Data Sets: A General Approach. *IEEE Trans. Knowl. Data Eng.*, 16(12):1472–1485, 2004.

[36] Michael J. Hutchings. Standing Crop and Pattern in Pure Stands of Mercurialis Perennis and Rubus Fruticosus in Mixed Deciduous Woodland. *Nordic Society Oikos*, 31(3):351–357, 1979.

[37] Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan. *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley, 2008.

[38] Krzysztof Koperski and Jiawei Han. Discovery of Spatial Association Rules in Geographic Information Databases. In *Proc. of the 4th Int'l Symposium on Advances in Spatial Databases*, pages 47–66, 1995.

[39] Zhi Liu and Yan Huang. Mining Co-locations under Uncertainty. In *In Proc. of the 13th Int'l Symposium on Advances in Spatial and Temporal Databases*, pages 429–446, 2013.

[40] Craig G. Macfarland and William G. Reeder. Cleaning Symbiosis Involving Galapagos Tortoises and Two Species of Darwin's Finches. *Zeitschrift für Tierpsychologie*, 34(5):464–483, 1974.

[41] Sandeep Mane, Carson Murray, Shashi Shekhar, Jaideep Srivastava, and Anne Pusey. Spatial Clustering of Chimpanzee Locations for Neighborhood Identification. In *Proc. of the 5th IEEE Int'l Conference on Data Mining*, pages 737–740, 2005.

[42] Francis H. C. Marriott. Barnard's Monte Carlo Tests: How Many Simulations? *Appl. Statist.*, 28(1):75–77, 1979.

[43] Michael D. McKay, Richard J. Beckman, and William J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 42(1):55–61, 2000.

[44] Pradeep Mohan, Shashi Shekhar, James A. Shine, and James P. Rogers. Cascading Spatio-temporal Pattern Discovery: A Summary of Results. In *Proc. of the 10th SIAM Int'l Conference on Data Mining*, pages 327–338, 2010.

[45] Yasuhiko Morimoto. Mining Frequent Neighboring Class Sets in Spatial Databases. In *Proc. of the 7th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 353–358, 2001.

[46] Robert Munro, Sanjay Chawla, and Pei Sun. Complex Spatial Relationships. In *Proc. of the 3rd IEEE Int'l Conference on Data Mining*, pages 227–234, 2003.

[47] Jerzy Neyman and Elizabeth L. Scott. Statistical Approach to Problems of Cosmology. *J. of the Royal Statist. Society, Series B*, 20(1):1–43, 1958.

[48] Hawaii Institute of Marine Biology. Mall and Parking Lot. http://hawaii.edu/himb/directions.htm. [Online; accessed 1-January-2014].

[49] George L. W. Perry, Ben P. Miller, and Neal J. Enright. A Comparison of Methods for the Statistical Analysis of Spatial Point Patterns in Plant Ecology. *Plant Ecol.*, 187:59–82, 2006.

[50] Feng Qian, Qinming He, and Jiangfeng He. Mining Spatial Co-location Patterns with Dynamic Neighborhood Constraint. In *In the Proc. of the 13th European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 238–253, 2009.

[51] Brian D. Ripley. The Second-Order Analysis of Stationary Point Processes. *J. of Appl. Probability*, 13(2):255–266, 1976.

[52] Klaus Rohde. *Nonequilibrium Ecology*. Cambridge University Press, 2006.

[53] Stephen H. Roxburgh and Mamoru Matsuki. The Statistical Validation of Null Models Used in Spatial Association Analyses. *Nordic Society Oikos*, 85(1):68–78, 1999.

[54] Hans Sagan. *Space-Filling Curve*. Springer Verlag, 1995.

[55] Ansaf Salleb-Aouissi, Christel Vrain, and Cyril Nortet. QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules. In *Proc. of the* 20*th Int'l Joint Conference on Artifical Intelligence*, pages 1035–1040, 2007.

[56] Ashok Savasere, Edward Omiecinski, and Shamkant B. Navathe. Mining for Strong Negative Associations in a Large Database of Customer Transactions. In *Proc. of the* 14*th Int'l Conference on Data Engineering*, pages 494–502, 1998.

[57] Henry Scherren. *Popular Natural History*. Cassell and Co., 1913.

[58] Katja Schladitz and Adrian J. Baddeley. A Third Order Point Process Characteristic. *Scandinavian Journal of Statistics*, 27(4):657–671, 2000.

[59] Small Science. Crocodile and the Plover Bird. http://coglab.hbcse.tifr.res.in/teacher-resources/multimedia-resources/symbiosis/crocodile-and-the-plover-bird. [Online; accessed 1-January-2014].

[60] Shashi Shekhar and Yan Huang. Discovering Spatial Co-location Patterns: A Summary of Results. In *Proc. of the 7th Int'l Symposium on Spatial and Temporal Databases*, pages 236–256, 2001.

[61] Ramakrishnan Srikant and Rakesh Agrawal. Mining Quantitative Association Rules in Large Relational Tables. *SIGMOD Rec.*, 25(2):1–12, 1996.

[62] Wei-Guang Teng, Ming-Jyh Hsieh, and Ming-Syan Chen. A Statistical Framework for Mining Substitution Rules. *Knowl. Inf. Syst.*, 7(2):158–178, 2005.

[63] André Tscheschel. *Reconstruction of Random Porous Media Using a Simulated Annealing Method*. Diploma-math, TU Bergakademie Freiberg, Germany, 2001.

[64] Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A New Algorithm for Generating All the Maximal Independent Sets. *SIAM J. Computing*, 6(3):505–517, 1977.

[65] Song Wang, Yan Huang, and Xiaoyang S. Wang. Regional Co-locations of Arbitrary Shapes. In *In Proc. of the* 13*th Int'l Symposium on Advances in Spatial and Temporal Databases*, pages 19–37, 2013.

[66] Elzbieta Wienawa-Narkiewicz. *Light and Electron Microscopic Studies of Retinal Organisation*. Doctoral, Australian National University, Canberra, 1983.

[67] Wikipedia. Tiger. http://en.wikipedia.org/wiki/Tiger, 2013. [Online; accessed 1-January-2014].

[68] Xindong Wu, Chengqi Zhang, and Shichao Zhang. Mining Both Positive and Negative Association Rules. In *Proc. of the* 19*th Int'l Conference on Machine Learning*, pages 658–665, 2002.

[69] Xiangye Xiao, Xing Xie, Qiong Luo, and Wei-Ying Ma. Density Based Co-location Pattern Discovery. In *Proc. of the* 16*th ACM Int'l Symposium on Advances in Geographic Information Systems*, pages 250–259, 2008.

[70] Jin S. Yoo and Mark Bow. Mining Spatial Colocation Patterns: A Different Framework. *Data Min. Knowl. Discov.*, 24(1):159–194, 2012.

[71] Jin S. Yoo and Shashi Shekhar. A Partial Join Approach for Mining Co-location Patterns. In *Proc. of the* 12*th ACM Int'l Workshop on Geographic Information Systems*, pages 241–249, 2004.

[72] Jin S. Yoo and Shashi Shekhar. A Joinless Approach for Mining Spatial Colocation Patterns. *IEEE Trans. Knowl. Data Eng.*, 18(10):1323–1337, 2006.

[73] Jin S. Yoo, Shashi Shekhar, Sangho Kim, and Mete Celik. Discovery of Co-evolving Spatial Event Sets. In *Proc. of the* 6*th SIAM Int'l Conference on Data Mining*, pages 306–315, 2006.

Appendix A

Experimental Results-Synthetic Data Sets

Table A.1, A.2, A.3, A.4 A.5, A.6, A.7, A.8, A.9, and A.10 show complete results for the experiments with synthetic data sets.

Table A.1: Inhibition experiment: a 2-size subset where features have an inhibition relationship

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Pattern reported as
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
$\{\circ, \triangle\}$	0.55	$1, 0.01$	0.075	$0.98, 0.04$	0.225	$1, 0.04$	0.275	$1, 0.02$	0.325	$0.98, 0.02$	Segregation

Table A.2: Inhibition experiment: a 3-size subset where features have an inhibition relationship

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Interaction reported as significant
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos},p_{\text{neg}}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
$\{\circ, \triangle\}$	0.62	$0.91, 0.11$	0.16	$0.85, 0.25$	0.26	$0.91, 0.12$	0.32	$0.89, 0.15$	0.32	$0.82, 0.21$	No
$\{\circ, +\}$	0.7	$0.77, 0.37$	0.16	$0.82, 0.3$	0.38	$0.6, 0.58$	0.4	$0.85, 0.25$	0.48	$0.61, 0.53$	No
$\{\triangle, +\}$	0.66	$0.88, 0.2$	0.14	$0.89, 0.18$	0.34	$0.85, 0.26$	0.46	$0.57, 0.59$	0.46	$0.71, 0.43$	No
$\{\circ, \triangle, +\}$	0.42	$0.99, 0.03$	0.02	$0.98, 0.04$	0.04	$0.98, 0.04$	0.1	$0.99, 0.04$	0.12	$0.99, 0.03$	Segregation

Table A.3: Auto-correlation experiment: a 2-size subset found in a data set with an auto-correlated feature

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Interaction reported as significant?
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
$\{\circ, \triangle\}$	0.46	$0.75, 0.31$	0.167	$0.56, 0.57$	0.275	$0.55, 0.51$	0.334	$0.46, 0.61$	0.342	$0.54, 0.51$	No

Table A.4: Mixed spatial interaction experiment: all subsets of size 2 found in a data set with 5 features

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Interaction reported as significant
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
	0.525	$1, 0.01$	0.025	$1, 0.01$	0.175	$1, 0.01$	0.25	$0.98, 0.03$	0.25	$1, 0.01$	Segregation
$\{\circ, +\}$	1	$0.01, 1$	0.568	$0.01, 1$	0.932	$0.01, 1$	0.95	$0.01, 1$	0.975	$0.01, 1$	Co-location
$\{\circ, \times\}$	1	$0.01, 1$	0.5	$0.01, 1$	0.925	$0.01, 1$	1	$0.01, 1$	1	$0.01, 1$	Co-location
$\{\circ, \Diamond\}$	0.55	$0.44, 0.57$	0.15	$0.46, 0.55$	0.275	$0.44, 0.59$	0.325	$0.35, 0.66$	0.325	$0.54, 0.47$	No
$\{\triangle, +\}$	0.593	$0.97, 0.03$	0.0678	$0.99, 0.03$	0.169	$0.99, 0.03$	0.195	$0.99, 0.02$	0.22	$0.99, 0.02$	Segregation
$\{\triangle, \times\}$	0.475	$0.99, 0.01$	0.1	$0.97, 0.03$	0.25	$0.98, 0.03$	0.275	$0.98, 0.02$	0.275	$0.98, 0.02$	Segregation
$\{\triangle, \Diamond\}$	0.575	$0.44, 0.57$	0.15	$0.41, 0.6$	0.3	$0.25, 0.76$	0.33	$0.32, 0.69$	0.367	$0.27, 0.74$	No
$\{+, \times\}$	1	$0.01, 1$	0.4	$0.01, 1$	0.729	$0.01, 1$	0.805	$0.01, 1$	0.864	$0.01, 1$	Co-location
$\{+, \Diamond\}$	0.559	$0.68, 0.34$	0.144	$0.56, 0.45$	0.254	$0.66, 0.35$	0.322	$0.63, 0.39$	0.372	$0.4, 0.61$	No
$\{\times, \Diamond\}$	0.6	$0.52, 0.49$	0.2	$0.21, 0.8$	0.325	$0.4, 0.62$	0.325	$0.54, 0.48$	0.45	$0.19, 0.82$	No

Table A.5: Mixed spatial interaction experiment: all subsets of size 3 found in a data set with 5 features

Interaction	All-instances-based		Sampling approach with different cell resolution								Interaction
pattern	SSCSP algorithm		$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		reported as
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	significant
$\{\circ, \triangle, +\}$	0.525	$0.35, 0.66$	0.017	$0.8, 0.23$	0.125	$0.37, 0.64$	0.175	$0.34, 0.67$	0.175	$0.76, 0.25$	No
$\{\circ, \triangle, \times\}$	0.4	$0.78, 0.23$	0.025	$0.74, 0.28$	0.125	$0.32, 0.7$	0.2	$0.13, 0.88$	0.2	$0.74, 0.27$	No
$\{\circ, \triangle, \Diamond\}$	0.275	$0.93, 0.08$	0.025	$0.56, 0.45$	0.05	$0.77, 0.24$	0.05	$0.9, 0.11$	0.05	$0.94, 0.07$	No
$\{\circ, +, \times\}$	1	$0.01, 1$	0.279	$0.01, 1$	0.703	$0.01, 1$	0.805	$0.01, 1$	0.864	$0.01, 1$	Co-location
$\{\circ, +, \Diamond\}$	0.55	$0.02, 0.99$	0.102	$0.01, 1$	0.25	$0.01, 1$	0.30	$0.01, 1$	0.325	$0.01, 1$	Co-location
$\{\circ, \times, \Diamond\}$	0.55	$0.01, 1$	0.1	$0.02, 0.99$	0.25	$0.01, 1$	0.325	$0.01, 1$	0.325	$0.01, 1$	Co-location
$\{\triangle, +, \times\}$	0.425	$0.68, 0.33$	0.025	$0.5, 0.54$	0.11	$0.49, 0.52$	0.136	$0.58, 0.43$	0.175	$0.45, 0.56$	No
$\{\triangle, +, \Diamond\}$	0.314	$0.85, 0.16$	0.0169	$0.6, 0.42$	0.059	$0.71, 0.31$	0.059	$0.9, 0.11$	0.076	$0.89, 0.12$	No
$\{\triangle, \times, \Diamond\}$	0.275	$0.88, 0.14$	0.025	$0.62, 0.39$	0.025	$0.94, 0.08$	0.025	$0.96, 0.08$	0.075	$0.88, 0.13$	No
$\{+, \times, \Diamond\}$	0.56	$0.01, 1$	0.13	$0.01, 1$	0.2	$0.02, 0.99$	0.271	$0.01, 1$	0.33	$0.01, 1$	Co-location

Table A.6: Mixed spatial interaction experiment: all subsets of size 4 & 5 found in a data set with 5 features

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Interaction reported as significant
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
$\{\circ, \triangle, +, \times\}$	0.4	$0.1, 0.91$	0.017	$0.24, 0.77$	0.1	$0.11, 0.9$	0.125	$0.12, 0.89$	0.125	$0.17, 0.85$	No
$\{\circ, \triangle, +, \Diamond\}$	0.275	$0.48, 0.55$	0.0169	$0.21, 0.8$	0.05	$0.15, 0.87$	0.05	$0.38, 0.63$	0.05	$0.49, 0.52$	No
$\{\circ, \triangle, \times, \Diamond\}$	0.25	$0.41, 0.6$	0.025	$0.2, 0.81$	0.025	$0.49, 0.52$	0.025	$0.67, 0.34$	0.05	$0.45, 0.58$	No
$\{\circ, +, \times, \Diamond\}$	0.55	$0.01, 1$	0.093	$0.01, 1$	0.194	$0.01, 1$	0.271	$0.01, 1$	0.325	$0.01, 1$	Co-location
$\{\triangle, +, \times, \Diamond\}$	0.25	$0.56, 0.45$	0.0169	$0.18, 0.85$	0.025	$0.57, 0.44$	0.025	$0.72, 0.29$	0.025	$0.8, 0.21$	No
$\{\circ, \triangle, +, \times, \Diamond\}$	0.25	$0.2, 0.81$	0.0169	$0.07, 0.94$	0.025	$0.17, 0.84$	0.025	$0.25, 0.76$	0.025	$0.34, 0.67$	No

Table A.7: Time comparison (in sec)

Join-less co-location algorithm		Sampling approach with different cell resolution				All-instances-based SSCSP
$PI_{\text{thre}} = 0.2$	$PI_{\text{thre}} = 0.5$	$l = R_d$	$l = \frac{R_d}{2}$	$l = \frac{R_d}{3}$	$l = \frac{R_d}{4}$	
7.43	4.78	44.73	96.45	180.46	316.54	600.59

Table A.8: Pairwise interaction experiment: a 2-size subset found significant at multiple distances

Pattern	Pattern found significant at 6 additional PID-values with their PIC-values and PI-values					
	1	2	3	4	5	6
$\{\circ, \triangle\}$	$0.0377, 4, 0.4$	$0.0533, 5, 0.5$	$0.0536, 6, 0.6$	$0.0644, 7, 0.0.7$	$0.0662, 8, 0.8$	$0.0709, 9, 0.9$

Table A.9: Pairwise interaction experiment: a non-significant pattern $\{\circ, \triangle\}$

PID	0.056	0.107	0.112	0.148	0.160	0.188	0.195	0.201	0.203	0.218	0.218	0.233
PIC	1	2	3	4	5	6	7	8	9	10	11	12
p-value	0.602	0.816	0.686	0.87	0.842	0.908	0.882	0.854	0.782	0.812	0.71	0.764
PI-value	0.1	0.2	0.3	0.3	0.4	0.4	0.4	0.5	0.6	0.7	0.7	0.7

Table A.10: Top 5 F-measure values computed for each of 5 different PI-thresholds using a traditional co-location mining approach

PI-threshold→	0.25		0.35		0.45		0.55		0.65	
	F-measure	R_d	F-measure	R_d	F-measure	R_d	F-measure	R_d	F-measure	R_d
Rank 1	0.667	0.051	0.667	0.072	0.667	0.072	0.667	0.078	0.667	0.080
Rank 2	0.6	0.493	0.6	0.529	0.6	0.543	0.6	0.669	0.6	0.769
Rank 3	0.545	0.499	0.545	0.554	0.545	0.583	0.5	0.058	0.5	0.063
Rank 4	0.5	0.028	0.5	0.056	0.5	0.056	0.375	0.722	0.462	0.8452
Rank 5	0.429	0.5069	0.462	0.569	0.462	0.583	0.333	0.722	0.4	0.848

Appendix B

Experimental Results-Real Data Sets

Table B.1, B.2, B.3, B.4 B.5, B.6, and B.7 show complete results for the experiments with real data sets.

Table B.1: Ants data: spatial interaction of $\circ =$ *Cataglyphis* and $\triangle =$ *Messor* ants

Interaction	All-instances-based		Sampling approach with different cell resolution								Pattern
pattern	SSCSP algorithm		$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		reported as
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	significant
$\{\circ, \triangle\}$	0.44	$0.19, 0.82$	0.0882	$0.38, 0.67$	0.1617	$0.60, 0.41$	0.2353	$0.25, 0.78$	0.2353	$0.37, 0.65$	No

Table B.2: Bramble canes data: co-location of newly emergent (mark 1), 1 year old (mark 2), and 2 years old (mark 3) canes

Interaction	All-instances-based		Sampling approach with different cell resolution								Pattern
pattern	SSCSP algorithm		$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		reported as
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	significant
$\{1, 2\}$	0.9896	$0.03, 0.98$	0.8727	$0.01, 1$	0.9610	$0.01, 1$	0.9688	$0.01, 1$	0.9714	$0.01, 1$	Co-location
$\{1, 3\}$	0.9331	$0.01, 1$	0.5097	$0.0, 1$	0.7130	$0.01, 1$	0.7548	$0.01, 1$	0.7744	$0.01, 1$	Co-location
$\{2, 3\}$	0.9169	$0.01, 1$	0.4727	$0.01, 1$	0.6779	$0.01, 1$	0.7247	$0.01, 1$	0.7195	$0.01, 1$	Co-location
$\{1, 2, 3\}$	0.9143	$0.01, 1$	0.4416	$0.01, 1$	0.6675	$0.01, 1$	0.7143	$0.01, 1$	0.7091	$0.01, 1$	Co-location

Table B.3: Found co-location patterns from Lansing Woods data

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Significant co-location pattern
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	p_{pos}-value	PI_{obs}	p_{pos}-value	PI_{obs}	p_{pos}-value	PI_{obs}	p_{pos}-value	PI_{obs}	p_{pos}-value	
$\{\textit{Black oak}, \textit{Hickory}\}$	0.862	0.02	0.456	0.03	0.603	0.03	0.653	0.03	0.681	0.02	Yes
$\{\textit{Maple}, \textit{Misc.}\}$	0.835	0.01	0.389	0.03	0.56	0.02	0.634	0.01	0.642	0.01	Yes
$\{\textit{Black oak}, \textit{Hickory}, \textit{Red oak}\}$	0.855	0.02	0.331	0.04	0.535	0.04	0.598	0.03	0.633	0.02	Yes
$\{\textit{Maple}, \textit{Misc.}, \textit{Red oak}\}$	0.702	0.03	0.28	0.04	0.486	0.04	0.526	0.03	0.552	0.03	Yes

Table B.4: Found segregation patterns from Lansing Woods data

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Significant segregation pattern
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	p_{neg}-value	PI_{obs}	p_{neg}-value	PI_{obs}	p_{neg}-value	PI_{obs}	p_{neg}-value	PI_{obs}	p_{neg}-value	
$\{\textit{Black oak}, \textit{Maple}\}$	0.766	0.03	0.268	0.04	0.414	0.03	0.486	0.03	0.502	0.02	Yes
$\{\textit{Black oak}, \textit{Misc.}\}$	0.43	0.01	0.067	0.01	0.141	0.01	0.178	0.01	0.193	0.01	Yes
$\{\textit{Hickory}, \textit{Maple}\}$	0.866	0.02	0.516	0.01	0.7	0.03	0.75	0.02	0.77	0.01	Yes
$\{\textit{Hickory}, \textit{Misc.}\}$	0.516	0.03	0.198	0.04	0.311	0.02	0.35	0.04	0.354	0.03	Yes
$\{\textit{Hickory}, \textit{Maple}, \textit{White oak}\}$	0.516	0.03	0.151	0.04	0.294	0.03	0.343	0.02	0.346	0.03	Yes

Table B.5: Some non-significant patterns found from Lansing Woods data

Interaction pattern	All-instances-based SSCSP algorithm		Sampling approach with different cell resolution								Pattern reported?
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$		
	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	PI_{obs}	$p_{\text{pos}}, p_{\text{neg}}$	
$\{$*Hickory*, *Red oak*$\}$	1	$0.39, 0.62$	0.75	$0.55, 0.48$	0.91	$0.84, 0.19$	0.96	$0.6, 0.43$	0.96	$0.63, 0.38$	No
$\{$*Hickory*, *White oak*$\}$	1	$0.63, 0.39$	0.74	$0.89, 0.14$	0.95	$0.77, 0.23$	0.97	$0.77, 0.26$	0.982	$0.63, 0.42$	No
$\{$*Hickory*, *Red oak*, *White oak*$\}$	1	$0.31, 0.69$	0.54	$0.90, 0.11$	0.87	$0.83, 0.17$	0.92	$0.73, 0.27$	0.94	$0.69, 0.36$	No

Table B.6: Found statistically significant co-locations from Toronto data. A feature present in a co-location is shown by $\surd$

Low density resid.	Univ.	Fire Station	Police station	College	Other emerg. service	Nursing home	Public prim. school	Separate prim. school	PI_{obs}
√	√								0.363
√		√							0.079
√			√						0.157
	√			√					0.263
	√	√							0.095
	√		√						0.120
	√				√				0.022
		√	√						0.095
√			√			√			0.030
√	√	√							0.047
√	√		√						0.087
√	√				√				0.022
	√	√		√					0.0158
	√	√	√						0.0317
√	√	√	√						0.0158
	√	√					√	√	0.012

Table B.7: PI-values and p_{pos}-values of the reported statistically significant co-locations from Toronto data

Statistically significant co-location	All-instances-based SSCSP algorithm		Sampling approach							
			$l = R_d$		$l = \frac{R_d}{2}$		$l = \frac{R_d}{3}$		$l = \frac{R_d}{4}$	
	PI_{obs}	p_{pos}	PI_{obs}	p_{pos}	PI_{obs}	p_{pos}	PI_{obs}	p_{pos}	PI_{obs}	p_{pos}
Low density resid., Univ.	0.363	0.01	0.147	0.03	0.286	0.02	0.314	0.02	0.314	0.01
Low density resid., Fire st.	0.079	0.01	0.024	0.02	0.03	0.02	0.046	0.03	0.059	0.01
Low density resid., Police st.	0.157	0.02	0.044	0.03	0.075	0.02	0.092	0.02	0.139	0.01
Univ., College	0.263	0.01	0.152	0.02	0.184	0.02	0.25	0.01	0.25	0.01
Univ., Fire st.	0.095	0.02	0.037	0.04	0.054	0.03	0.062	0.02	0.086	0.02
Univ., Police st.	0.120	0.01	0.044	0.02	0.064	0.01	0.092	0.02	0.092	0.01
Univ., Other emerg. srv.	0.022	0.01	0.01	0.04	0.014	0.02	0.018	0.01	0.018	0.01
Fire st., Police st.	0.095	0.03	0.035	0.04	0.049	0.03	0.068	0.02	0.079	0.02
Low density resid., Police st., Nursing home	0.03	0.03	0.008	0.03	0.015	0.03	0.022	0.02	0.028	0.02
Low density resid., Univ, Fire st.	0.047	0.01	0.0154	0.02	0.03	0.01	0.0327	0.01	0.0381	0.01
Low density resid., Univ, Police st.	0.087	0.01	0.02	0.02	0.05	0.01	0.078	0.01	0.078	0.01
Low density resid., Univ, Other Emg Srv.	0.022	0.02	0.006	0.02	0.01	0.03	0.013	0.02	0.018	0.01
Univ, Fire St., College	0.0158	0.01	0.004	0.02	0.01	0.01	0.01	0.01	0.012	0.02
Univ, Fire St., Police St.	0.0317	0.03	0.014	0.04	0.017	0.03	0.024	0.02	0.029	0.02
Low den. res., Univ, Fire St., Police St.	0.0158	0.04	0.008	0.07	0.01	0.02	0.012	0.02	0.0135	0.01
Univ, Fire St., Public prim. sch., Sep. prim. sch.	0.012	0.02	0.005	0.03	0.008	0.03	0.01	0.02	0.01	0.02