A neighborhood graph based approach to regional co-location pattern discovery: A summary of results

Pradeep Mohan
Department of Computer Science
University of Minnesota
Minneapolis, USA
mohan@cs.umn.edu

Shashi Shekhar
Department of Computer Science
University of Minnesota
Minneapolis, USA
shekhar@cs.umn.edu

James A. Shine
Geospatial Engineering and Research Division
Engineer Research and Development Center
Alexandria, VA, USA

James P. Rogers
Geospatial Engineering and Research Division
Engineer Research and Development Center
Alexandria, VA, USA

Zhe Jiang
Department of Computer Science
University of Minnesota
Minneapolis, USA
zhe@cs.umn.edu

Nicole Wayant
Geospatial Engineering and Research Division
Engineer Research and Development Center
Alexandria, VA, USA

ABSTRACT

Regional co-location patterns (RCPs) represent collections of feature types frequently located together in certain localities. For example, RCP \(< (\text{Bar}, \text{Alcohol} – \text{Crimes})\), Downtown > suggests that a co-location pattern involving alcohol-related crimes and bars is often localized to downtown regions. Given a set of Boolean feature types, their geo-located instances, a spatial neighbor relation, and a prevalence threshold, the RCP discovery problem finds all prevalent RCPs (pairs of co-locations and their prevalence localities). RCP discovery is important in many societal applications, including public safety, public health, climate science and ecology. The RCP discovery problem involves three major challenges: (a) an exponential number of subsets of feature types, (b) an exponential number of candidate localities and (c) a tradeoff between accurately modeling pattern locality and achieving computational efficiency. Related work does not provide computationally efficient methods to discover all interesting RCPs with their natural prevalence localities. To address these limitations, this paper proposes a neighborhood graph based approach that discovers all interesting RCPs and is aware of a pattern’s prevalence localities. We identify partitions based on the pattern instances and neighbor graph. We introduce two new interest measures, a regional participation ratio and a regional participation index to quantify the strength of RCPs. We present two new algorithms, Pattern Space (PS) enumeration and Maximal Locality (ML) enumeration and show that they are correct and complete. Experiments using real crime datasets show that ML pruning outperforms PS enumeration.

Categories and Subject Descriptors

H.2 [Information Systems]: Data Management; H.2.8 [Data Mining]: Spatial Databases and GIS

General Terms

Algorithms, Performance, Experimentation

Keywords

Spatial analysis, Spatial heterogeneity, Regional co-location patterns, Regional participation index, prevalence localities, maximal localities

1. INTRODUCTION

Regional co-location patterns (RCPs) represent subsets of feature types frequently located together in certain localities in a study area. For example, in public safety, the RCP \(< (\text{Bar}, \text{Alcohol} – \text{Crimes}), \text{Downtown} > \) indicates that a co-location pattern involving alcohol-related crimes and bars is often localized in downtown regions.

RCP discovery problem: Given Boolean feature types and their geo-located instances, along with a spatial neighbor relation and a prevalence threshold, RCP discovery finds all prevalent RCPs (pairs of co-locations and their prevalence localities). For example, Figure 1(a), shows an illustrative crime dataset consisting of three feature types\(^1\). Bars, Assault crimes and Drunk Driving. Assault crimes are represented by blue triangles, Bars are represented by red circles and drunk driving by green squares. Given this dataset, a spatial neighbor relation, and some prevalence threshold, the RCP discovery process identifies RCPs as shown in Figure 1(b). In the figure, the RCPs (shown) correspond to the co-location pattern \{ABC\}. The dotted purple polygons represent localities where the pattern \{ABC\} may be

---

\(^1\)For brevity, the term “feature types” and “feature” are used interchangeably in this paper. For example, a bar feature may correspond to a bar feature type such as bar closing or happy hour etc., that occurs at a bar location.
prevalent and the shaded orange triangles represent individual instances of the pattern \{ABC\}. 

**Illustrative Application:** RCPs may provide useful insights about the localities of patterns in many applications, including public safety, public health, ecology and climate science. For example, in public health, the RCP <\textit{Asbestos, Lung Cancer}, \textit{Mining Areas}> may be interpreted as a frequent localization of the co-location between asbestos and lung cancer in areas with intense mining activity. In ecology, the RCP <\textit{Egyptian – Plover – Bird, Crocodile}, \textit{Nile Delta}> may suggest a frequent localization of the symbiosis between Egyptian plover birds and crocodiles in the Nile river delta. An emergency management example may be the RCP, <\textit{Hurricane, Economic Damage}, \textit{East Coast}> which may imply a frequent economic damage due to Hurricanes in East coast of the US.

For a more detailed illustration, we revisit our public safety example. Police and other law enforcement agencies seek to identify strategies to mitigate crime in many unsafe localities. RCP discovery may help police departments distinguish between normal and crime prone areas within a city. For example, Figure 2 shows an example RCP <\textit{Bar, Assault}, \textit{Downtown}> represented by a blue colored polygon. The black squares in Figure 2 are bar locations, small red circles are assault crime locations. The number 0.209 shown in Figure 2 adjoining the blue polygon refers to the probability that the bars within the polygon may be related to frequent occurrence of assaults in that area. This may lead to an investigation of regional differences between bars downtown and bars in other regions. Some questions would be: Why do downtown bars often lead to assaults crimes but bars in other regions seldom do so? Is it regional differences in geographic concentration? Are there regional differences in patron demography or crowd density? Are there policy differences in screening, bouncing, policing? Answering these questions make it easier to plan effective intervention strategies, including identifying suitable patrol routes, appropriate placement of police vehicles etc.

**Challenges:** Discovering RCPs from spatial datasets poses several challenges. First, the number of possible subsets of feature types is exponential. For example, the number of candidate subsets is \(O(2^{|M|})\), where \(|M|\) is the number of feature types. Second, the number of possible localities for each of these subsets may be exponential in dataset size. One solution to reduce the large space of localities is to use a neighbor relation. This yields a cardinality of \(\Omega(2^{|M|})\) (including all overlapping localities) candidate localities, where \(|M|\) is the number of feature types. For example, Figure 1(c) shows three prevalence localities (as dotted purple polygons) for the pattern \{ABC\} in Figure 1(b).

![Figure 1](image1.png)

**Figure 1:** Illustrative example of Regional co-location patterns (Best viewed in color)

![Figure 2](image2.png)

**Figure 2:** An RCP from Lincoln, NE
and achieving computational efficiency. This creates a need to design new interest measures that can model local prevalence of RCPs in addition to providing desirable computational properties to quickly prune non prevalent RCPs.

**Related Work:** RCP discovery literature falls into two categories: zoning-based approaches [7] and fitness function based clustering methods [10]. Zoning based approaches require a user specified space partitioning scheme (e.g., Quad tree). Given user specified partitions, a global co-location pattern discovery algorithm [11] is applied in each partition. The thick black lines in Figure 1(c) represent the user specified partitioning scheme. The partitioning method employed by this approach is independent of the natural shape of a pattern’s prevalence localities that may be present in the data. Hence, it may break up a prevalence locality. For example, in Figure 1(c), the localities $P_{L1}(\{ABC\})$ and $P_{L2}(\{ABC\})$ of the pattern $\{ABC\}$ overlap with multiple quads in a Quad tree. In this scenario, the zoning based approach partitions these prevalence localities and may miss them completely.

The second category of related work is a fitness function based clustering approach that finds a collection of regions and identifies the most interesting pattern in each region based on a maximum interestingness criterion [10]. Clustering is a data partitioning approach for detecting regions within a spatial dataset and may not be aware of a pattern’s prevalence localities. Such an approach may not discover all prevalent RCPs and their prevalence localities due to the maximum interestingness criterion employed in the region discovery process. In addition, fitness function based clustering does not explore pruning strategies to reduce the search space of candidate RCPs. In summary, the related work does not provide computationally efficient methods to discover all interesting RCPs with their natural prevalence localities. To address the limitations of related work, this paper proposes a spatial neighborhood graph based approach as shown in Figure 1.

**Contributions:** Specific contributions of our work are as follows: (a) We model regional co-location pattern (RCP)s and formulate the RCP discovery problem; (b) We introduce two new interest measures, the Regional Participation Ratio and the Regional Participation Index to accurately model pattern localities; (c) We present two new algorithms, **Pattern Space (PS) Enumeration** and **Maximal Locality (ML) enumeration** and characterize their correctness and completeness; (d) Experimental evaluation with real datasets shows that ML pruning outperforms PS enumeration.

**Scope:** This paper focuses on modeling and discovering RCPs. A Spatial neighbor relation is an important input parameter; however, the paper does not address issues related to the choice of spatial neighbor relation. This issue is application domain dependent and will be addressed in future work. Prevalence localities of RCPs are modeled as a convex hull. Alternative representation of prevalence localities are beyond the scope of this paper. A geometric or graph theoretical enumeration and pruning of prevalence localities (e.g. convex hulls) is also outside the scope of this paper.

**Outline:** The rest of the paper is organized as follows: Section II defines some basic concepts and formalizes the RCP discovery problem. Section III describes the computational approaches and proves their correctness and completeness. Results of a computational evaluation are given in Section IV. We discuss issues related to the RCP discovery problem in Section V. Section VI concludes the paper.

2. PROBLEM FORMULATION

This section recalls some basic concepts from co-location pattern discovery [11, 14], introduces the new interest measures and formulates the RCP discovery problem.

2.1 Basic Concepts

A **spatial feature instance** represents a location embedded in a continuous spatial framework that may contain other non-spatial information including a label that represents instances of real world entities such as bars, crime reports etc. A feature instance is boolean if it does not contain any additional quantitative information other than its location and a label. A **spatial feature type** is a collection of spatial feature instances that represent the same kind of real world entity. For example, in public safety, spatial feature types may be crime types, bars etc. Spatial feature types are usually stored as individual tables in a spatial database.

Spatial feature instances can be related to each other using a **spatial neighbor relation**. A spatial neighbor relation ($R$) is a symmetric and transitive relation defined over the set of all spatial feature instances in a spatial dataset. The spatial neighbor relation can be specified using a spatial neighborhood size threshold (i.e., a distance threshold) that may be a few miles. For example, a bar and an assault crime may be considered as spatial neighbors if they are within a distance of 1 mile. A spatial neighbor relation over a set $S_I$ of spatial feature instances may be formally represented as a **spatial neighborhood graph**, $G = (S_I, E_I)$, where, $E_I$ represents a set of undirected edges representing pairs of feature instances in $S_I \times S_I$. For example, the applica-
A **Prevalence Locality** ($P_L$) of a co-location pattern is a subset of the spatial framework which contains a concentration (a large fraction) of instances participating in the co-location pattern referred to in an RCP. For example, Figure 1(b) shows the instances of the pattern $\{ABC\}$ represented as shaded orange triangles. A collection of one or more of these triangles is enclosed by a prevalence locality shown as purple polygons with dotted boundaries.

A prevalence locality can also be represented as a convex hull containing the instances of a pattern. A **regional co-location pattern (RCP)** is generally a subset of feature types occurring in certain localities of a non-uniform spatial framework. An RCP can be represented as a combination of a co-location pattern and one of its prevalence localities.

For example, Figure 1(b) shows examples of three RCPs corresponding to the co-location pattern $\{ABC\}$, namely $RCP1 = \langle \{ABC\}, P_{L1}(\{ABC\}) \rangle$ present in the south-east corner of the map, $RCP2 = \langle \{ABC\}, P_{L2}(\{ABC\}) \rangle$ in the center and $RCP3 = \langle \{ABC\}, P_{L3}(\{ABC\}) \rangle$ in the north-east corner of the map.

### 2.2 Quantifying the Prevalence of RCPs

In spatial data mining, the goal of interest measure design is to balance statistical interpretation and computational efficiency. In many spatial analysis applications, the non-uniform nature of the spatial framework leads to an important spatial property called **spatial heterogeneity** where the properties of a spatial pattern are not necessarily the same across geographic space [5, 8, 9]. For instance, in public safety, the opportunities of committing an offense are not uniform across space. In addition, an important application constraint that influences interest measure design is the ability to predict the instances of an RCP given an instance of a participating feature type and its location. Based on these constraints we define our interest measures as follows:

**Definition 1**: A Regional participation ratio, $RPR(\text{RCP}, f)$, is the conditional probability $Pr(\text{RCP} | f)$ of finding an instance of an RCP given an instance of spatial feature type $f$. Formally, $RPR(\text{RCP}, f)$ can be written as:

$$RPR(\text{RCP}, f) = \frac{\text{# instances of } f \text{ participating in } \text{RCP}}{\text{# instances of } f \text{ in dataset}},$$

where $f$ is a participating spatial feature type in an RCP.

**Definition 2**: A Regional participation index, $RPI(\text{RCP})$, is defined as the lower bound on the conditional probability of observing an instance of an RCP having observed an instance of all its participating feature types. In other words, $RPI$ is defined as the minimum of $RPR(\text{RCP}, f)$ over all the feature types participating in an RCP. Hence, $RPI$ can be formally written as:

$$RPI(\text{RCP}) = \min_{f} RPR(\text{RCP}, f)$$

For example, for the RCP, $\langle \{ABC\}, P_{L2} \rangle$ shown in Figure 1(b), two of the four instances of spatial feature type $A$ participate in the co-location pattern $\{ABC\}$ within the prevalence locality $P_{L2}$, so the RPR for $A$ is $\frac{2}{4} = 0.5$. Similarly, the RPR for $B$ and $C$ are $\frac{2}{4}$ and $\frac{1}{4}$ respectively, so the RPI of the RCP $\langle \{ABC\}, P_{L2}(\{ABC\}) \rangle$ is

$$RPI(\langle \{ABC\}, P_{L2} \rangle) = \min \left\{ \frac{2}{4}, \frac{2}{4}, \frac{1}{4} \right\} = \frac{1}{4} = 0.25.$$
The enumeration phase is shown in this figure as dotted blue arrows. Figure 5 shows the enumeration of candidate RCPs for the example dataset in Figure 1(a). There are two key computational steps in the enumeration phase, namely, a spatial join to compute pattern instances (shown as the spatial join operation in Figure 5) and connected component identification to compute prevalence localities (shown as the convex polygon in Figure 5). The spatial join operation is computed using the partition based spatial merge (PBSM) join algorithm [16, 12]. Connected component identification is performed using a simple breadth first search on the set of pattern instances. Once a connected component is identified, prevalence localities are generated as a convex hull that encloses the connected component. By this means, PS enumeration ensures that the entire space of patterns is explored to generate all possible RCPs.

The second phase of PS enumeration is pruning. During this phase, the algorithm computes the RPI of RCPs starting from size \( k = 2 \). If the RPI(RCP) \( \geq P_\theta \), the RCP is accepted and the prevalence locality of its co-location pattern is marked as strong. Otherwise, the RCP is pruned. The pruning phase iterates until all RCPs corresponding to the co-location pattern of the maximum size are tested. Figure 5 shows the pruning phase as thick double arrows. During this phase all RCPs corresponding to all co-location patterns are examined and tested for prevalence. Figure 5 shows RCPs that do not pass the prevalence threshold test and which get pruned out using red crosses. A detailed pseudocode of PS enumeration can be found in the appendix.

**Correctness and Completeness:** PS enumeration ensures that a correct set of pattern instances is computed via the spatial join algorithm. The connected component and prevalence locality computation steps of PS enumeration ensure that the prevalence localities of co-location patterns are computed correctly. Based on the above steps, a correct set of RCPs is computed. Since PS enumeration is a brute force algorithm, it does not prune any RCP unless the entire space is enumerated. Hence, PS enumeration ensures that no correct RCP is left out during the enumeration process. Thus, during the pruning phase, the algorithm only needs to test each RCP for the user specified threshold and prune it out if the threshold is not met. Hence, PS enumeration reports a correct and complete set of RCPs.

**Limitations:** PS enumeration is computationally very expensive because it looks at the entire space of candidate RCPs. In many practical situations, many spurious patterns will be generated, only to be pruned out later towards the end. To enhance computational performance and achieve quick pruning of non-prevalent RCPs, we propose the Maximal Locality (ML) pruning algorithm.

### 3.2 Maximal Locality (ML) pruning

In this part, we introduce a new notion called a Maximal Locality. We show that the RPI exhibits desirable computational properties (e.g., antimonotonicity) within a maximal locality. The ML pruning algorithm takes advantage of these properties to enhance computational performance by pruning non-prevalent RCPs quickly.

**Definition 3:** A Maximal Locality is a subset of the spatial framework which contains a concentration (a large fraction) of instances corresponding to different feature types.
A Maximal locality may be viewed as a connected subset of the spatial neighborhood graph, \( G = (S, E) \). For example, Figure 6 shows a collection of maximal localities by brown polygons with dotted boundaries. The collection of feature instances within these polygons are connected within a subset of the spatial neighborhood graph.

From Figure 6 and Figure 1(b), it can be seen that a maximal locality may contain several RCPs and often encloses all the prevalence localities corresponding to different RCPs. Another important observation from Figure 6 is that maximal localities by definition are mutually disjoint. For example, Figure 6 contains three maximal localities, \( ML_1 \), \( ML_2 \) and \( ML_3 \) in the southeast corner, center and northeast corner respectively. The three maximal localities are mutually disjoint as they do not share any feature instances.

Based on the above definition, we describe two key phases of the ML pruning algorithm: (a) maximal locality computation and (b) local enumeration and pruning. Performance optimization is performed during the local enumeration and pruning phase.

Algorithm 1 ML Pruning

**Input:** (a) Spatial Dataset  
(b) Boolean Spatial feature Types  
(c) Spatial Neighbor Relation \((R)\)  
(d) Prevalence Threshold, \( P_0 \)

**Output:** A set of RCPs such that \( RPI(RCP) \geq P_0 \)

1: Initialization
2: Generate Maximal Localities using a join under neighbor relation \( R \) on the spatial dataset.
3: for all \( ML_i \) \& Maximal localities do
   4: for all \( k \) such that \( 2 \leq k \leq \text{MaxPossibleSize} \) OR no prevalent patterns do
      5: Generate size-\( (k) \) candidate co-locations.
      6: for all size-\( (k) \) candidate co-locations do
         7: Generate pattern instances.
         8: Compute \( RPI(\text{size} - (k) \text{ co-location}) \) within \( ML_i \).
         9: if \( RPI \leq P_0 \) then
            10: Prune pattern from \( ML_i \).
            11: else
            12: Generate Prevalence localities of size-\( (k) \) co-location instances within \( ML_i \).
            13: Generate RCPs of size-\( (k) \) co-location and its prevalence localities within \( ML_i \).
            14: Test all RCPs for prevalence and return prevalent RCPs.
         15: end if
      16: end for
5: end for
6: end for

Detailed explanation of steps in the algorithm:

**Steps 1-2** are the first phase of the algorithm, that is, **Maximal locality computation**. Maximal localities can be computed in several ways. A simple way to identify them is to compute the spatial neighborhood graph represented as an edge list and then identify a connected component using simple graph search operations such as a breadth first search or depth first search. The identified connected components are stored as edge lists and correspond to instances of size-\( 2 \) co-location patterns. The spatial neighborhood graph itself can be computed using a spatial join under the neighbor relation \( R \) applied to the entire spatial dataset. The PBSM [16, 12] spatial join algorithm is used to compute the neighborhood graph. Figure 7 shows the detailed working of the algorithm. The maximal locality computation step is shown by the box with a dotted red boundary. As shown in the figure, the algorithm makes use of a spatial join to compute the neighborhood graph shown as a join operation represented with dotted brown lines. Maximal localities identified for the example dataset in Figure 1(a) are represented as polygons with a dotted brown boundary and are labeled as \( ML_1 \), \( ML_2 \) and \( ML_3 \) respectively. The maximal localities are represented as a convex polygon that encloses the connected components identified from the neighborhood graph.

**Steps 3 to 19** enumerate over all the maximal localities identified in the previous steps and compute the prevalent RCPs. This phase is called the Maximal locality pruning phase. During maximal locality pruning, the algorithm computes pattern instances of size-\( k \) co-location patterns within each maximal locality using a spatial join operation. The generated size-\( k \) candidate co-locations are evaluated for prevalence within the maximal locality and pruned if they do not meet the prevalence threshold specified by the user.

**Steps 3 to 19** achieve performance optimization by taking advantage of two important properties exhibited by the RPI within a maximal locality. (a) the RPI of a co-location within a maximal locality is an upper bound to the RPI of any RCP corresponding to \( C \) within the maximal locality and (b) the RPI is anti-monotonic [4] within a maximal locality. ML pruning takes advantage of the first property to prune out all RCPs corresponding to a co-location pattern within a maximal locality using a single RPI evaluation. By exploiting anti-monotonicity of the RPI within a maximal locality, ML pruning ensures that filtering a size-\( k \) co-location automatically removes all its supersets. The effect of exploiting these properties is shown in Figure 7 for the maximal locality \( ML_1 \). In this locality, co-location patterns \{\( AB \)\} and \{\( BC \)\} are not prevalent. Pruning these patterns based on the former property ensures that the RCPs of these patterns need not be enumerated. The use of the second property ensures that a superset of patterns, \{\( ABC \)\} i.e. the pattern \{\( ABC \)\}, is automatically pruned out before even being generated.

**Step 5** generates candidate size-\( (k) \) co-location patterns (not RCPs) from size-\( (k-1) \) prevalent co-locations. **Steps 6-15** perform pattern instance computation and pruning within each Maximal locality. If a co-location pattern meets the user specified threshold, then ML pruning enumerates all RCPs contained within the maximal locality, computes their RPI and prunes the non prevalent ones. For example, in Figure 7, even though the RPI of the pattern \{\( BC \)\} is 0.5 and satisfies the threshold within the maximal locality. Whereas, the constituent RCPs containing the pattern \{\( BC \)\} do not pass the threshold. Hence, they are all examined and pruned.

3.3 Analytical Evaluation

ML pruning exploits desirable properties of the RPI within a maximal locality to enhance computational per-
formance. Based on these properties, we show that the ML pruning approach returns a correct and complete set of RCPs. We first prove these properties and then prove the correctness and completeness of ML Pruning.

**Lemma 1.** The RPI of a co-location $C$ in a maximal locality $M_L$, is an upper bound to the RPI of any RCP corresponding to $C$ contained within $M_L$. That is, $\text{RPI}(C) \geq \text{RPI}(RCP_j)$, $\forall RCP_j$ of the form $<C, P_{k+1}>$ contained within $M_L$.

**Proof.** Let $C$ be a co-location pattern and let $f$ be any feature type participating in $C$. Based on the definitions in Section 2.2, we have $\text{RPI}(C, f)$ within $M_L$

\[= \text{# instances of feature type } f \text{ participating in } C \text{ within } M_L\]

Let, $RCP_j$ be any RCP corresponding to the co-location $C$ and contained entirely within $M_L$. Then based on Definition 1, $\text{RPI}(RCP_j, f) = \text{# instances of feature type } f \text{ participating in } RCP_j$

Since $M_L$ contains many RCPs corresponding to co-location $C$, we have

$\text{# instances of feature type } f \text{ participating in } C \text{ within } M_L = \text{# instances of feature type } f \text{ participating in } RCP_j + \hspace{1cm} (2)$

$\geq \text{# instances of } f \text{ from other RCPs} + \hspace{1cm} (3)$

$\geq \text{# instances of feature type } f \text{ participating in } C \text{ within } M_L$

Dividing both sides by the $\# \text{ instances of } f \text{ in the dataset}$ which is a positive quantity, the above inequality gets preserved and we get,

$\text{RPI}(C, f) \geq \text{RPI}(RCP_j, f)$ \hspace{1cm} (1)

Inequality (1) is true for every feature $f$ participating within co-location $C$ including the one with the least RPI.

$\Rightarrow \text{RPI}(C) \geq \text{RPI}(RCP_j)$ \hspace{1cm} (2)

Inequality (2) is true for any RCP contained within $M_L$.

Hence, the RPI of a co-location within a maximal locality is an upper bound to any RCP corresponding to the co-location contained within the maximal locality.

**The ML pruning approach exploits Lemma 1 to ensure that, pruning a co-location pattern within a maximal locality also prunes out all the RCPs within the locality.**

**Lemma 2.** The RPI is an anti-monotone within a Maximal Locality. That is, given two co-location patterns, $C(k)$ and $C(k+1)$, which are both present in the Maximal Locality, with $C(k) < C(k+1)$, we have $\text{RPI}(C(k)) \geq \text{RPI}(C(k+1))$, where $k$ is the number of feature types participating in the co-location.

**Proof.** Let $f$ be a feature type such that $f \in C(k+1)$ and $f \notin C(k)$. From the definitions in Section 2.2, we have $\text{RPI}(C(k+1))$ within $M_L$ by $\text{RPI}(C(k))$, $\text{RPI}(C(k+1), f)$ In the construction of $C(k+1)$ from $C(k)$ no new instances of features belonging to $C(k)$ are added. Only instances of features that are not present in the instances of $C(k)$ are added to create $C(k+1)$ and its instances. This is because, maximal localities are by definition mutually disjoint connected subsets. Hence, any addition of features and instances to create larger size co-locations cannot include features and their instances from other maximal localities. Hence, $\text{RPI}(C(k+1))$ within $M_L$ $\leq \text{min}\text{RPI}(C(k)), \text{RPI}(C(k+1), f)$. \hspace{1cm} (4)

Lemma 2 is true only for co-location patterns within Maximal localities. It is not true for RCPs.

**Lemma 3.** ML Pruning is correct. That is, ML Pruning returns only prevalent RCPs.

**Proof.** Two key constraints for correctness are: (a) pattern instances are computed correctly and (b) during pruning, when a co-location pattern is prevalent within a maximal locality, its RCPs within the locality are enumerated and only prevalent RCPs are reported. The first constraint is satisfied by Step 7 of the ML pruning algorithm, which performs a spatial join to compute the correct set of instances of a co-location pattern within a maximal locality. The second constraint is met by Steps 12-14 of ML pruning. In these
steps, the algorithm enumerates all the RCPs corresponding to an accepted co-location. During this enumeration only RCPs whose RPI value crosses the user specified prevalence threshold are reported. Hence, ML pruning returns a correct set of RCPs. □

**Lemma 4.** ML Pruning is complete. That is, ML pruning returns all prevalent RCPs and does not miss any pattern.

**Proof.** Completeness of ML pruning depends on two constraints: (a) pruning a $size - (k)$ co-location pattern within a maximal locality does not eliminate any valid RCPs and (b) pruning a co-location ensures that its supersets will never be prevalent. The first constraint is guaranteed by Lemma 1, ensuring that a non-prevalent co-location pattern within a maximal locality cannot have prevalent RCPs and they can be pruned automatically without further examination. Lemma 2 guarantees that a $size - (k)$ co-location that is not prevalent cannot have a $size(k+1)$ co-location that is prevalent within any Maximal locality. Satisfying these constraints, ML pruning returns a complete set of prevalent RCPs. □

4. **EXPERIMENTAL EVALUATION**

This section presents an experimental evaluation of the two RCP discovery algorithms using real crime datasets from Lincoln City, NE, USA. The dataset contains over forty crime types (e.g. vandalism, assaults and larceny) and other ancillary features including locations of bars, parking lots, entertainment places etc. [1]. Figure 8 shows a subset of the Lincoln crime dataset containing three features, namely, bars, alcohol-related crimes and assaults. The black squares in Figure 8 represent bar locations, the red plus signs represent assault crimes and the green diamonds represent alcohol-related crimes from the year 2007. For the purpose of experimental evaluation, subsets of the original Lincoln data were extracted in order to characterize dominance zones in performance of different RCP discovery algorithms.

The goal of experimental evaluation was to compare the performance of the two RCP discovery algorithms, namely, dataset size, number of features, the prevalence threshold ($P_\theta$) and the spatial neighborhood size. The metric for ranking different algorithms was run time as measured by both CPU time and I/O time. The algorithms were implemented in C++ and the performance evaluation was performed on Macintosh running OS-X 10.6 with 16GB main memory and 2.26 GHz dual quad core Intel Xeon processors. In addition to a run time based ranking of the two approaches we also report the number of RCP evaluations performed by each approach.

**Effect of prevalence threshold:** Understanding the effect of prevalence threshold on RCP discovery can influence user decisions regarding appropriate values for different performance and data analysis requirements. The prevalence threshold was varied from 0.001 to 0.5 on a 7498 instance dataset with 5 feature types while fixing the spatial neighborhood size at 700 feet. The performance comparison of both algorithms is shown in Figure 10(a). The performance of the PS enumeration approach does not depend on the user specified prevalence threshold as it examines all possible RCPs. On the other hand, ML pruning performance is sensitive to the user specified threshold even though it outperforms the PS enumeration approach for the entire range of prevalence thresholds. In agreement with this trend, the number of RCP evaluations performed by the ML pruning approach is shown in Figure 10(b). The performance of ML pruning is complete. That is, ML pruning returns a correct set of prevalent RCPs.

Figure 9: Experimental setup

Figure 10: Effect of prevalence threshold on RCP discovery algorithms of the PS enumeration approach does not depend on the user specified prevalence threshold as it examines all possible RCPs. On the other hand, ML pruning performance is sensitive to the user specified threshold even though it outperforms the PS enumeration approach for the entire range of prevalence thresholds. In agreement with this trend, the number of RCP evaluations performed by the ML pruning approach is shown in Figure 10(b).
approach are much lower as compared to the PS enumeration approach and leads to significant savings in computational cost. Figure 10(b) shows that the ML pruning approach evaluates only a very small portion of RCPs compared to the numbers evaluated by the PS enumeration approach.

**Effect of spatial neighborhood size:** The crime dataset with 7498 instances and 5 feature types was used and the spatial neighborhood size was varied from 100 feet to 700 feet. The prevalence threshold was fixed at 0.07. Figure 11(a) shows that the computational performance of both RCP discovery algorithms increases significantly with the increase in spatial neighborhood size. However, ML pruning approach still outperforms the PS enumeration by a factor of 1.5 to 5. Figure 11(b) shows the number of RCP evaluations performed by the two approaches. The ML pruning approach performs only one fourth the number of RCP evaluations performed by the PS enumeration approach and hence outperforms the PS enumeration approach by a large factor.

**Effect of number of feature types:** The effect of number of feature types on RCP discovery is exponential. Figure 12(a) shows that the ML pruning approach evaluates only a very small portion of RCPs compared to the numbers evaluated by the PS enumeration approach and leads to significant savings in computational cost. Figure 10(b) shows that the ML pruning approach evaluates only a very small portion of RCPs compared to the numbers evaluated by the PS enumeration approach.

Table 1: Effect of Data size on RCP discovery approaches

<table>
<thead>
<tr>
<th>Data Size</th>
<th>PS enumeration</th>
<th>ML Pruning</th>
<th>PS enumeration</th>
<th>ML Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3496</td>
<td>20</td>
<td>28</td>
<td>1550</td>
<td>413</td>
</tr>
<tr>
<td>5498</td>
<td>161</td>
<td>117</td>
<td>2174</td>
<td>484</td>
</tr>
<tr>
<td>7638</td>
<td>1574</td>
<td>716</td>
<td>2665</td>
<td>645</td>
</tr>
</tbody>
</table>

Figure 11: Effect of spatial neighborhood size on performance of RCP discovery approaches. Figure 12(a) shows that the ML pruning approach performs the PS enumeration approach by a factor of 1.2. ML pruning enhances computational performance by filtering out co-location patterns within maximal localities. Such filtering reduces the number of RCP evaluations performed by ML pruning and enhances performance. Figure 12(b) shows the number of RCP evaluations performed by both approaches. ML pruning only examines around 50% of the RCPs evaluated by PS enumeration.

**Effect of dataset size:** The number of instances in a dataset influences the number of RCP evaluations performed by either approach. In order to understand this influence, we analyzed the RCP discovery algorithms by varying dataset size. Table 1 shows the performance of both RCP discovery approaches for different dataset sizes. In this experiment, the spatial neighborhood size was fixed at 700 feet and the prevalence threshold was set to 0.07. As the dataset size increases, the performance gap between PS enumeration and ML pruning widens. A similar trend also can be observed for the number of RCP evaluations. From the last two columns of Table 1 it is clear that, the ML pruning approach examines only one-fourth of the RCPs examined by PS enumeration.

5. DISCUSSION

This section discusses the applicability of traditional association rule mining based measures such as support and confidence for RCP discovery. Existing literature in traditional association rule mining and spatial data mining have explored approaches based on the support measure by fitting the notion of a transaction on a continuous spatial framework [2, 3]. When support measure is used to quantify interestingness in a spatial framework, the transactions created may undercount or over count the prevalence of an RCP there by missing some interesting patterns or generating too many spurious patterns. Hence, such a measure may not be natural for handling spatial datasets. A detailed explanation regarding the limitations of support to quantify spatial patterns has been clarified in co-location pattern discovery [11].

6. CONCLUSION AND FUTURE WORK

This paper modeled Regional co-location pattern(RCP) and designed novel interest measures, the regional participation ratio(RPR) and the regional participation index(RPI) to quantify the regional prevalence of RCPs. The computational structure of RCP discovery was established and a
novel Maximal Locality (ML) Pruning algorithm was introduced that exploits certain desirable properties of the RPI within maximal localities to prune non interesting patterns quickly and enhance computational performance. Experimental evaluation using real crime datasets showed that the proposed ML pruning approach outperforms PS enumeration and reduces the number of RCP evaluations.

Several issues merit attention in future work. In this paper, a prevalence region of an RCP is represented as a convex hull. There can be other ways to represent prevalence regions (e.g., non-convex). In most practical situations, understanding and interpreting a closed geometric object such as a convex hull may be easier for domain scientists. In future work, we plan to explore other ways of representing the prevalence region of a pattern including, delaunay triangulation and non convex representations. In addition, we would explore a statistical interpretation of RPI based on existing local statistical measures such as local K-function [15] and alternative interpretations of RPI that may quantify interestingness based on area of prevalence localities. We are also interested in exploring alternative representation models for RCPs (e.g., Raster) based on Kriging and Geographically Weighted regression [6].

The RCP discovery problem makes use of spatial neighbor relation that depends on a user specified neighbor size. A user specifying this input may possibly introduce some bias on the discovered patterns. However, a reasonable guess for a good neighborhood size may be obtained using statistical tools such as, CrimeStat [13], ArcGIS 10 etc. Detailed exploration to identify the right spatial neighbor relation is beyond the scope of this paper and would be explored in future work. We plan to explore a direct enumeration and pruning of prevalence localities using geometric or graph theoretical concepts and provide algebraic cost models comparing costs of various algorithms. In order to outline dominance zones we plan to evaluate the proposed algorithms comparing costs of various algorithms. In order to outline dominance concepts and provide algebraic cost models comparing costs of various algorithms.

Acknowledgement: We thank the members of the spatial database and data mining group for their feedback on the initial versions of this paper. We also thank Kim Koffolt for helping to improve the readability of this paper. We are grateful to Mr. Tom Casady, Police Chief, Lincoln City, NE for sharing the Lincoln crime dataset. We are also grateful to the reviewers of ACM GIS 2011 for their thoughtful comments. This work was supported by the US Army grant number W9132V-09-C-0009.

7. REFERENCES


APPENDIX

A. PSEUDOCODE OF PS ENUMERATION

Algorithm 2 : PS Enumeration

Input: (a) Spatial Dataset (b) Boolean Spatial feature Types (c) Spatial neighbor relation(R) (d) Prevalence Threshold, $P_0$

Output: A set of RCPs, where, $RPI(RCP) \geq P_0$

1: Initialization
2: for all $k$ such that $2 \leq k \leq MaxPossibleSize$ do
3: Generate size-($k$) co-locations from size-($k-1$) co-locations.
4: for all size-($k$) candidates do
5: Generate pattern instances.
6: Generate RCPs.
7: end for
8: end for
9: for $k = 2$ to $MaxPossibleSize$ do
10: Generate RCPs and compute their RPI.
11: if $RPI < P_0$ then
12: Prune RCP
13: end if
14: end for
15: end for
17: end