Colocation Mining: Exploring Local and Regional Interesting Patterns with the Map-Based Instance Table Approach

Abigail Kelly

Qualifying Exam

Advisor: Dr. Sainju

Department of Computational and Data Science

Middle Tennessee State University

Outline

- Motivation
- Basic Concepts
- Problem Definition
- Related Work
- Distance Threshold Calculation
- Map-Based Approach
- Results

Motivation

- Colocation: set of spatial features that commonly appear near each other within a geographical area

Motivation

- Colocation: set of spatial features that commonly appear near each other within a geographical area
- Example:
  - Ecology: symbiotic relationships between animals or plants

Motivation

- Colocation: set of spatial features that commonly appear near each other within a geographical area
- Example:
  - Ecology: symbiotic relationships between animals or plants
  - Public Health: diseases and environmental generators

Air Pollution and Lung Cancer
https://www.beckman.kr/

Motivation

- Colocation: set of spatial features that commonly appear near each other within a geographical area
- Example:
  - Ecology: symbiotic relationships between animals or plants
  - Public Health: diseases and environmental generators
  - Public Safety: crime events and sources

Air Pollution and Lung Cancer
https://www.beckman.kr/

Mall Closing and Crime Events
https://www.istockphoto.com/

Basic Concepts

- Region
    - Area of interest comprising countries with shared borders
    - Individual countries are considered sub-regions within the larger region

Basic Concepts

- Region
  - Area of interest comprising countries with shared borders
  - Individual countries are considered sub-regions within the larger region
- Spatial Feature
  - Categorical attribute such as a terrorist attack type (e.g., bombing, hijack)

Basic Concepts

- Region
  - Area of interest comprising countries with shared borders
  - Individual countries are considered sub-regions within the larger region
- Spatial Feature
  - Categorical attribute such as a terrorist attack type (e.g., bombing, hijack)
- Feature Instance
  - Occurrence of a spatial feature at the same/different location

Basic Concepts

- Region
  - Area of interest comprising countries with shared borders
  - Individual countries are considered sub-regions within the larger region
- Spatial Feature
  - Categorical attribute such as a terrorist attack type (e.g., bombing, hijack)
- Feature Instance
  - Occurrence of a spatial feature at the same/different location
- Distance Threshold (d)
  - Two instances of different features are in a neighborhood relationship if and only if the distance between them is less than or equal to the distance threshold

Basic Concepts

- Region
  - Area of interest comprising countries with shared borders
  - Individual countries are considered sub-regions within the larger region
- Spatial Feature
  - Categorical attribute such as a terrorist attack type (e.g., bombing, hijack)
- Feature Instance
  - Occurrence of a spatial feature at the same/different location
- Distance Threshold (d)
  - Two instances of different features are in a neighborhood relationship if and only if the distance between them is less than or equal to the distance threshold
- Clique
  - Two or more instances if every pair of instances are spatial neighbors

Basic Concepts

- Region
  - Area of interest comprising countries with shared borders
  - Individual countries are considered sub-regions within the larger region
- Spatial Feature
  - Categorical attribute such as a terrorist attack type (e.g., bombing, hijack)
- Feature Instance
  - Occurrence of a spatial feature at the same/different location
- Distance Threshold (d)
  - Two instances of different features are in a neighborhood relationship if and only if the distance between them is less than or equal to the distance threshold
- Clique
  - Two or more instances if every pair of instances are spatial neighbors
- Colocation Pattern (CP)
  - Corresponding set of features to the CP instance (clique with different features)

Interestingness Measure

- Participation Ratio (PR)
    - Ratio of the number of unique feature instances that participate in colocation instances to the total number of feature instances

Interestingness Measure

- Participation Ratio (PR)
  - Ratio of the number of unique feature instances that participate in colocation instances to the total number of feature instances
- Participation Index (PI)
  - Minimum PR among all member features

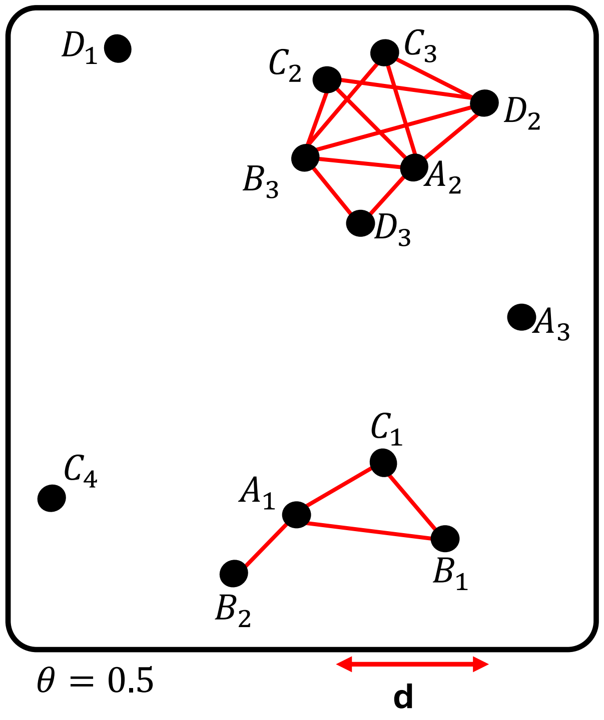
Interestingness Measure

- Participation Ratio (PR)
  - Ratio of the number of unique feature instances that participate in colocation instances to the total number of feature instances
- Participation Index (PI)
  - Minimum PR among all member features
- Prevalence Threshold (θ)
  - User-defined minimum threshold for participation index

Interestingness Measure

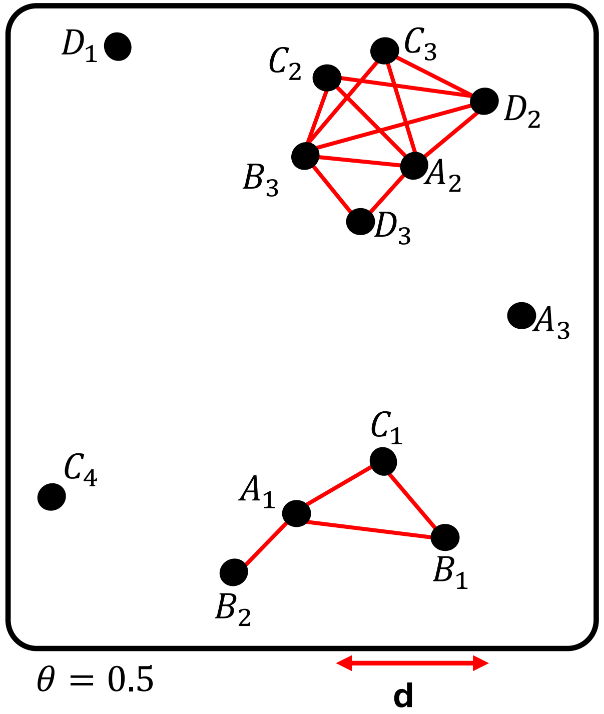
- Participation Ratio (PR)
  - Ratio of the number of unique feature instances that participate in colocation instances to the total number of feature instances
- Participation Index (PI)
  - Minimum PR among all member features
- Prevalence Threshold (θ)
  - User-defined minimum threshold for participation index
- Prevalent
  - A colocation pattern C is prevalent if and only if $PI \geq \theta$

Basic Concept Example: Array-Based



- Spatial Feature Types:
  - A, B, C, D

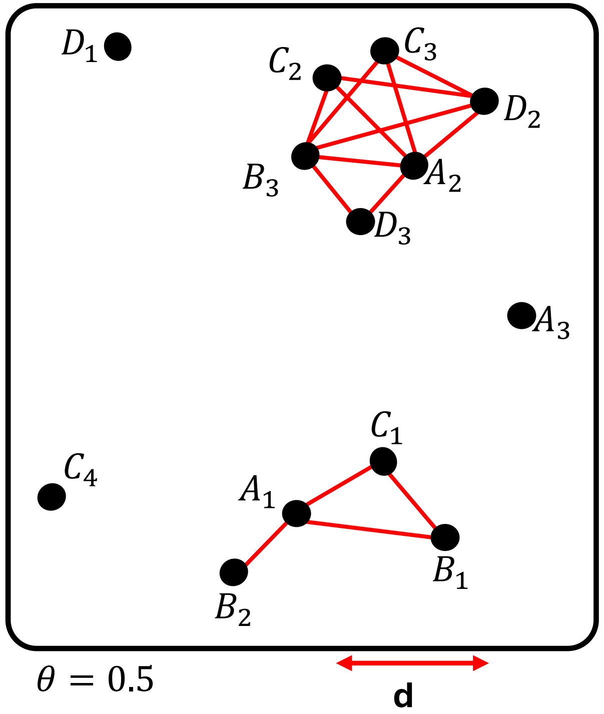
Basic Concept Example: Array-Based



- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

Basic Concept Example: Array-Based

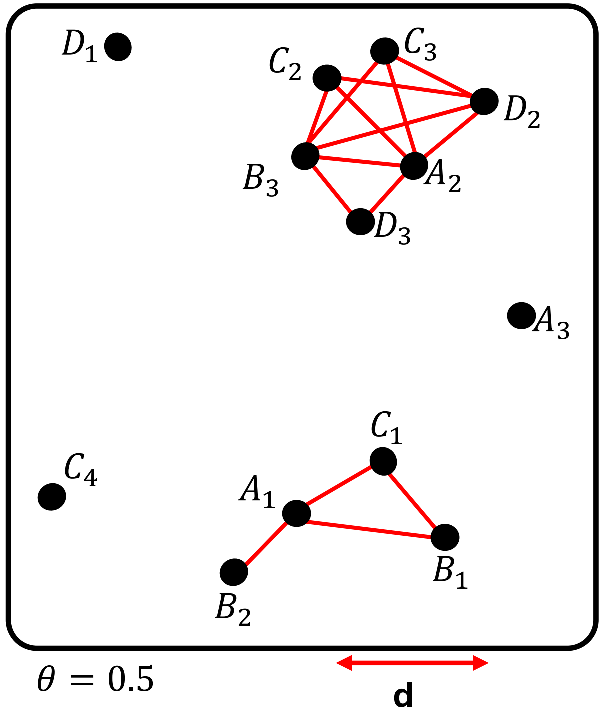


- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

- Candidate Colocation:
  - {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C, D}, {A,B,C}, {A,B,D}, {B,C,D}, {A,B,C,D}

Basic Concept Example: Array-Based

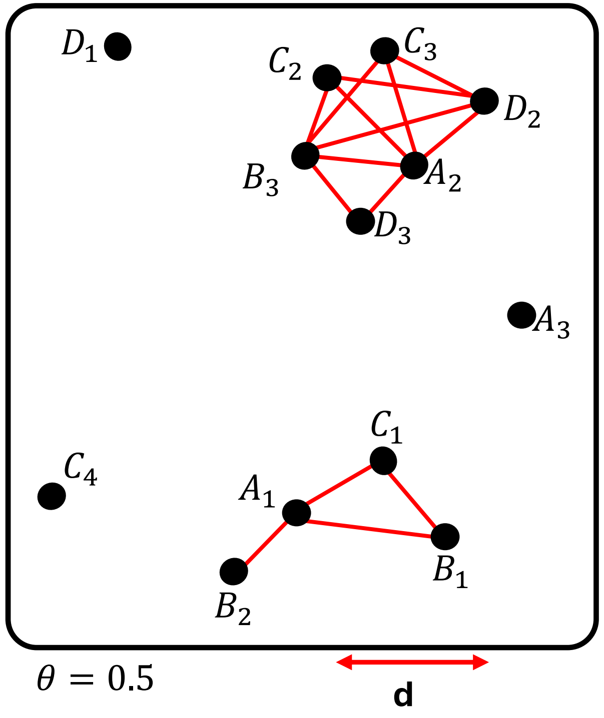


- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

- Candidate Colocation:
  - {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C, D}, {A,B,C}, {A,B,D}, {B,C,D}, {A,B,C,D}
- Neighbor Relationship (solid line)
  - (A_1,B_1), (A_1,B_2),…

Basic Concept Example: Array-Based



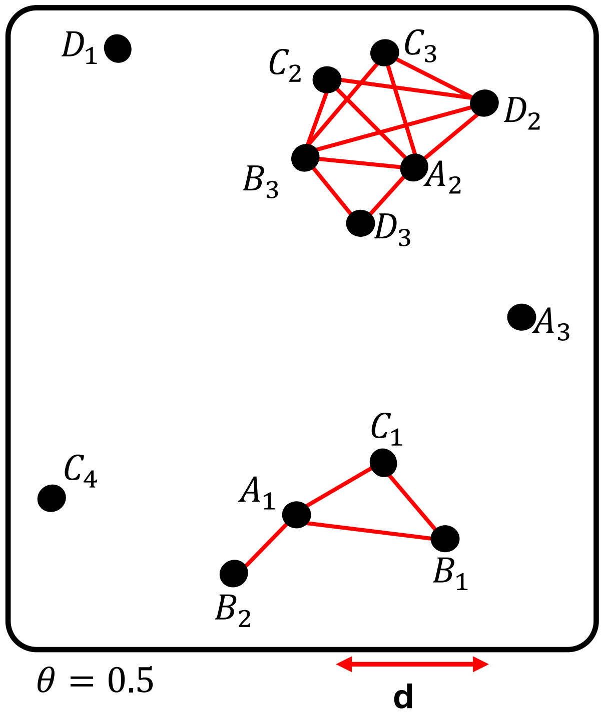
- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

- Candidate Colocation:
  - {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C, D}, {A,B,C}, {A,B,D}, {B,C,D}, {A,B,C,D}
- Neighbor Relationship (solid line)
  - (A_1,B_1), (A_1,B_2),…
- Table Instance:

A	B
A_1	B_1
A_1	B_2
A_2	B_3

Basic Concept Example: Array-Based



- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

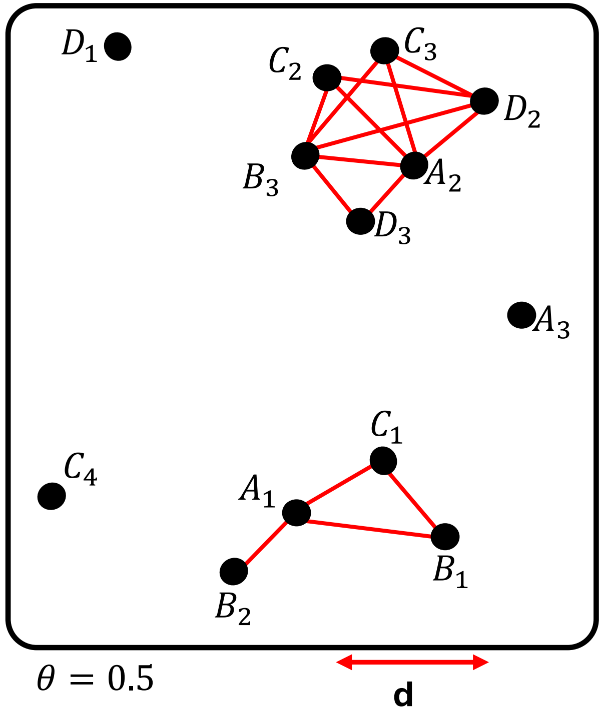
A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

- Candidate Colocation:
  - {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C, D}, {A,B,C}, {A,B,D}, {B,C,D}, {A,B,C,D}
- Neighbor Relationship (solid line)
  - (A_1,B_1), (A_1,B_2),...
- Table Instance:

A	B
A_1	B_1
A_1	B_2
A_2	B_3

⇨ $PR((A,B),A) = 2/3$ => 0.67
$PR((A,B),B) = 3/3$ => 1

Basic Concept Example: Array-Based



- Spatial Feature Types:
  - A, B, C, D
- Feature Instances:

A	B	C	D
A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
		C_4	

- Candidate Colocation:
  - {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C, D}, {A,B,C}, {A,B,D}, {B,C,D}, {A,B,C,D}
- Neighbor Relationship (solid line)
  - (A_1,B_1), (A_1,B_2),...
- Table Instance:

A	B
A_1	B_1
A_1	B_2
A_2	B_3

⇨

$PR((A, B), A) = 2/3$ => 0.67
$PR((A, B), B) = 3/3$ => 1
$PI = 0.67$

Instance Table

Array-Based

Map-Based

Instance Table

Array-Based

- Table holding all instances of colocation pattern C

Map-Based

Instance Table

Array-Based

- Table holding all instances of colocation pattern C
- Row Instance (RI) of a colocation pattern C
  - Each feature type in C appear only once

Map-Based

Instance Table

Array-Based

- Table holding all instances of colocation pattern C
- Row Instance (RI) of a colocation pattern C
  - Each feature type in C appear only once
- Table Instance (TI) of a colocation pattern C
  - Collection of all row instances of C

Map-Based

Instance Table

Array-Based

- Table holding all instances of colocation pattern C
- Row Instance (RI) of a colocation pattern C
  - Each feature type in C appear only once
- Table Instance (TI) of a colocation pattern C
  - Collection of all row instances of C

Map-Based

- Map-like structure with key-value pairs
  - Key: Instances of a subset of features of a pattern
  - Value: Instances of a subset of features that are colocated with the instance of the key

Instance Table

Array-Based

- Table holding all instances of colocation pattern C
- Row Instance (RI) of a colocation pattern C
  - Each feature type in C appear only once
- Table Instance (TI) of a colocation pattern C
  - Collection of all row instances of C

Map-Based

- Map-like structure with key-value pairs
  - Key: Instances of a subset of features of a pattern
  - Value: Instances of a subset of features that are colocated with the instance of the key
- Row Instance (RI) of a colocation pattern C
  - Pair consisting of the key and one of its values
  - Denoted $RI(C)$

Instance Table

Array-Based

- Table holding all instances of colocation pattern C
- Row Instance (RI) of a colocation pattern C
  - Each feature type in C appear only once
- Table Instance (TI) of a colocation pattern C
  - Collection of all row instances of C

Map-Based

- Map-like structure with key-value pairs
  - Key: Instances of a subset of features of a pattern
  - Value: Instances of a subset of features that are colocated with the instance of the key
- Row Instance (RI) of a colocation pattern C
  - Pair consisting of the key and one of its values
  - Denoted $RI(C)$
- Table Instance (TI) of a colocation pattern C
  - Collection of all row instances C
  - Denoted $TI(C)$ or $TI(C, f)$ where f is a feature of C

Problem Definition

- Given
  - A set of spatial features and their instances
  - Prevalence Threshold: θ

Problem Definition

- Given
  - A set of spatial features and their instances
  - Prevalence Threshold: θ
- Find
  - Spatial neighborhood relationship constraint: d
  - All colocation patterns with $PI \geq \theta$ in sub-regions and entire region

Problem Definition

- Given
  - A set of spatial features and their instances
  - Prevalence Threshold: θ
- Find
  - Spatial neighborhood relationship constraint: d
  - All colocation patterns with $PI \geq \theta$ in sub-regions and entire region
- Objective
  - Estimate the spatial neighborhood relationship constraint
  - Reduce memory utilization

Challenges

- Checking spatial neighborhood relationships between instances of different types

Challenges

- Checking spatial neighborhood relationships between instances of different types
- Number of candidate colocation patterns can grow exponentially

Challenges

- Checking spatial neighborhood relationships between instances of different types
- Number of candidate colocation patterns can grow exponentially
- Storing the intermediate collection of colocated instances for each pattern

Challenges

- Checking spatial neighborhood relationships between instances of different types
- Number of candidate colocation patterns can grow exponentially
- Storing the intermediate collection of colocated instances for each pattern
- Determining spatial neighborhood relationship constraint

Related Work

Colocation Mining

Related Work

Colocation Mining

Parallel Approaches

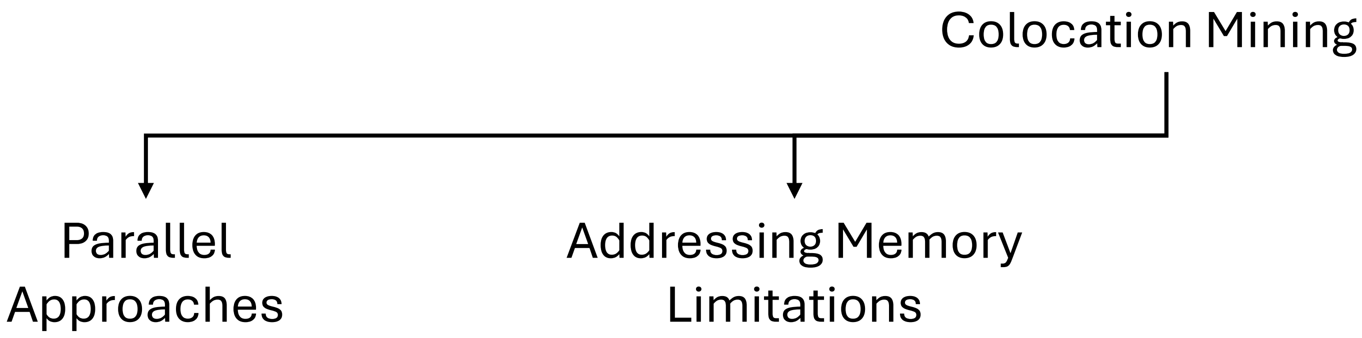
Related Work

Colocation Mining

Parallel Approaches

- Map Reduce Approach [Yoo, 2014]
- Grid-Based Approach [Sainju, 2017]
- GPU Algorithms [Sainju, 2018]

Related Work

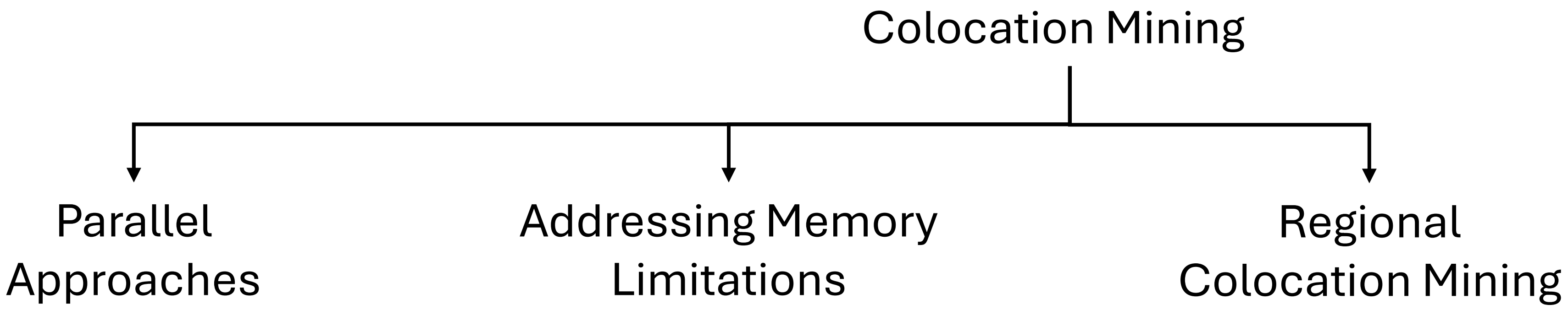


Related Work

Colocation Mining

- Parallel Approaches
- Addressing Memory Limitations
  - CPI-Tree [Wang, 2008]
  - iCPI-Tree [Wang, 2009]
  - CP-Tree [Sundaram, 2015]

Related Work



Related Work

Colocation Mining

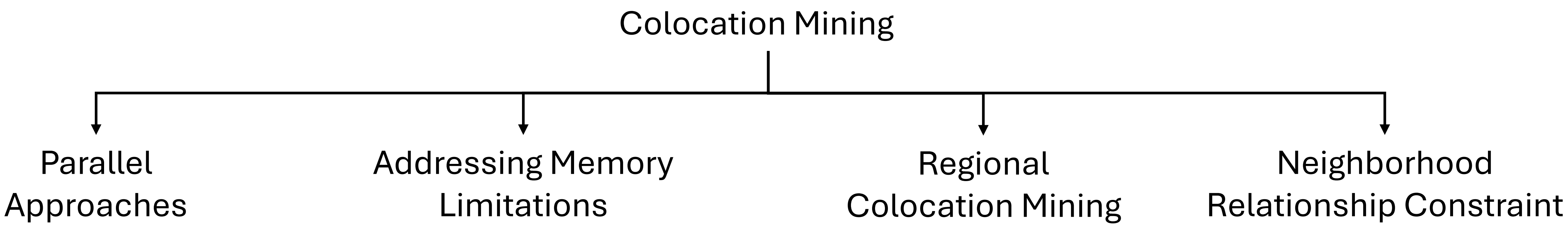
Parallel Approaches

Addressing Memory Limitations

Regional Colocation Mining

- Interesting Regions [Eick, 2008]
- Regional Interest Measures [Mohan, 2011]
- Multi-level Framework [Deng, 2017]

Related Work



Related Work

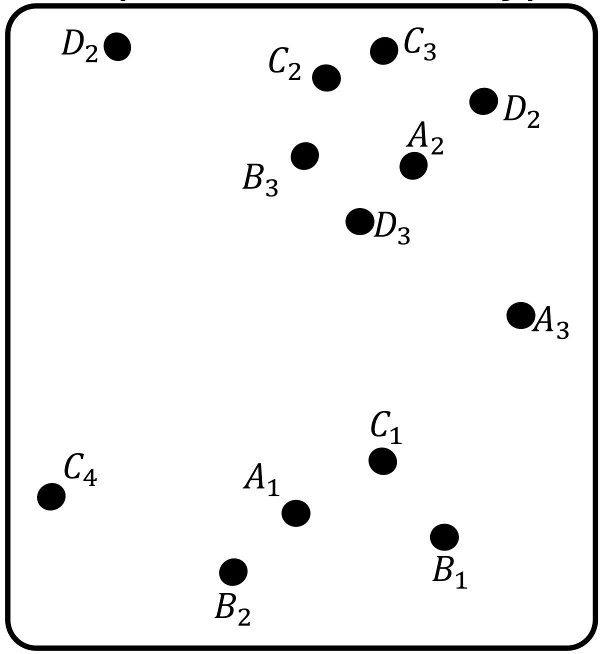
Colocation Mining

- Parallel Approaches
- Addressing Memory Limitations
- Regional Colocation Mining
- Neighborhood Relationship Constraint
  - Iterative Edge Selection [Qian, 2012]
  - Kth Nearest Neighbor [Yoo, 2012]
  - Nearest Neighbor Graph [Qian, 2014]

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:

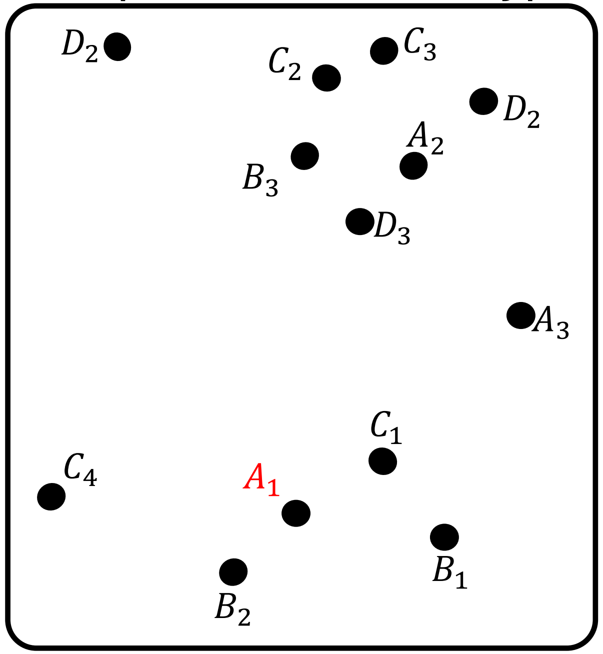


$K_{max} = \sqrt{13} \approx 4$
[Hassanat, 2014]

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:

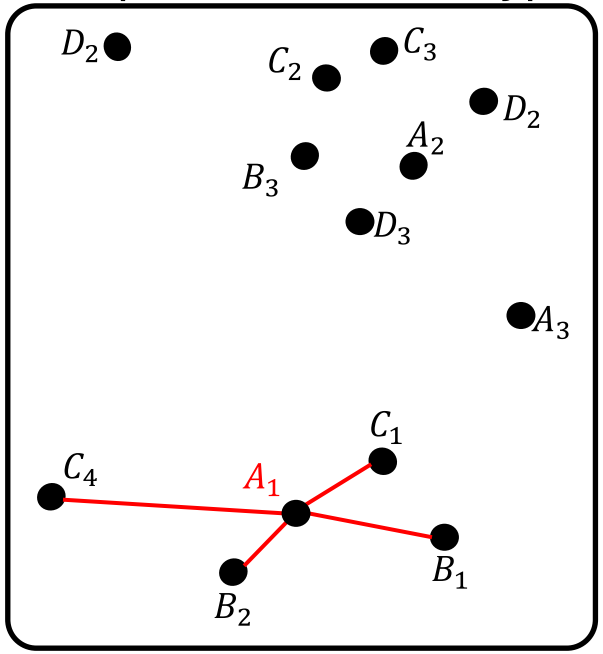


$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



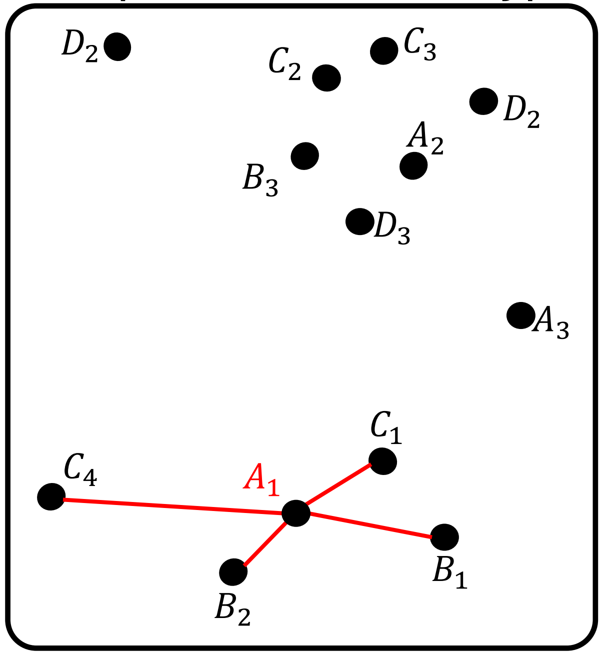
$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|]]$$

Example with feature type A:



$$K_{max} = \sqrt{13} \approx 4$$

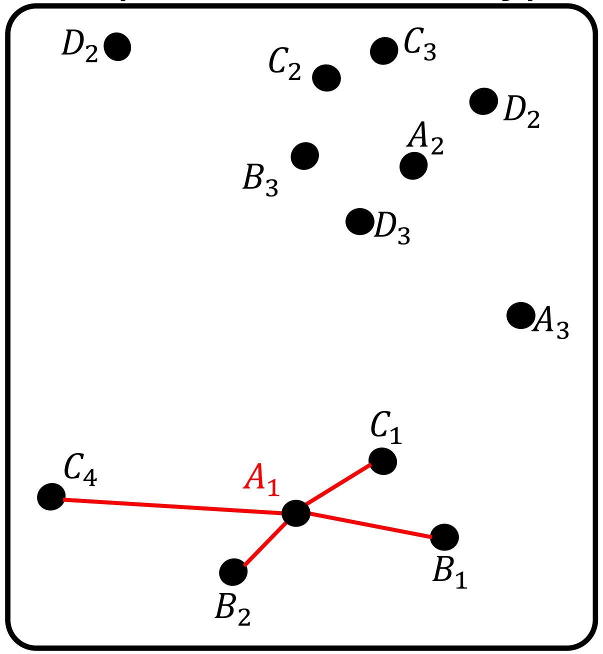
Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|]]$

shortest distance → longest distance

Example with feature type A:

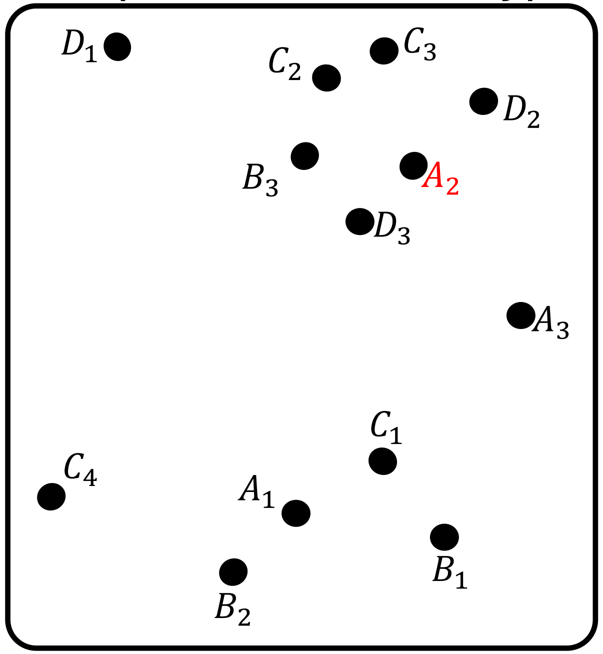


$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:

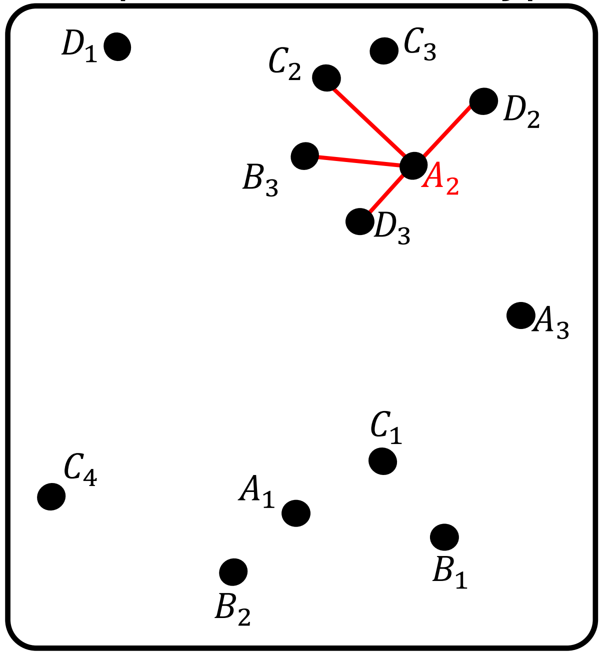


$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



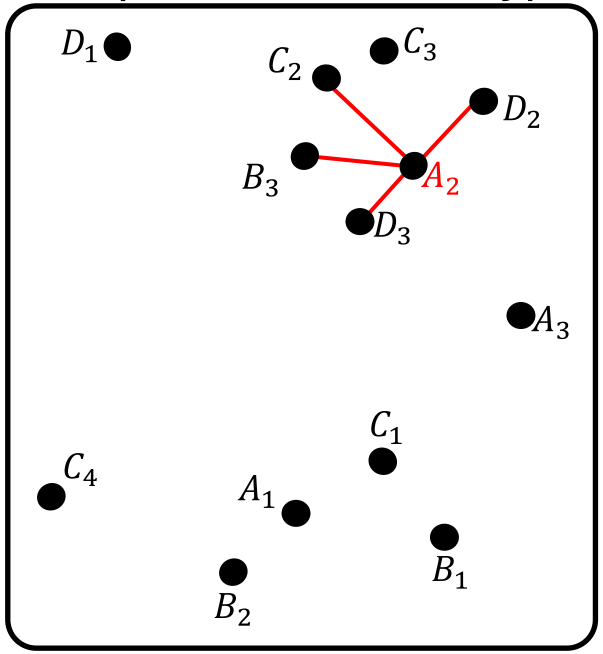
$$K_{max} = \sqrt{13} \approx 4$$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|]]$$

Example with feature type A:



$K_{max} = \sqrt{13} \approx 4$

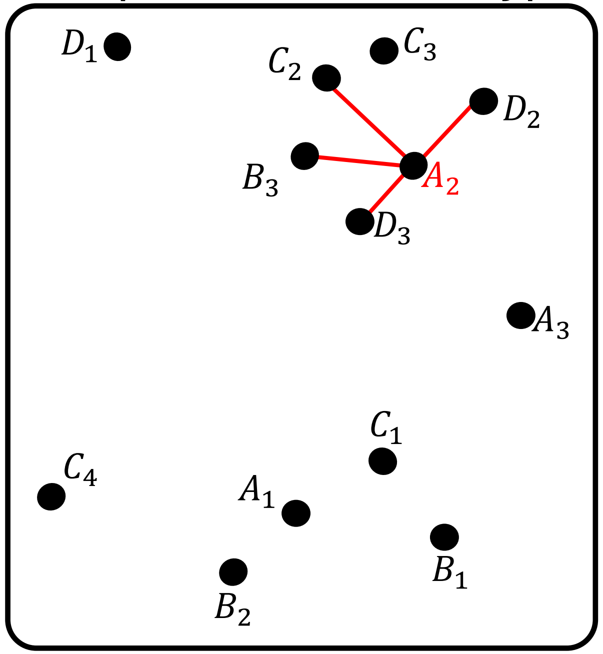
Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], \\ [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|]]$$

shortest distance → longest distance

Example with feature type A:

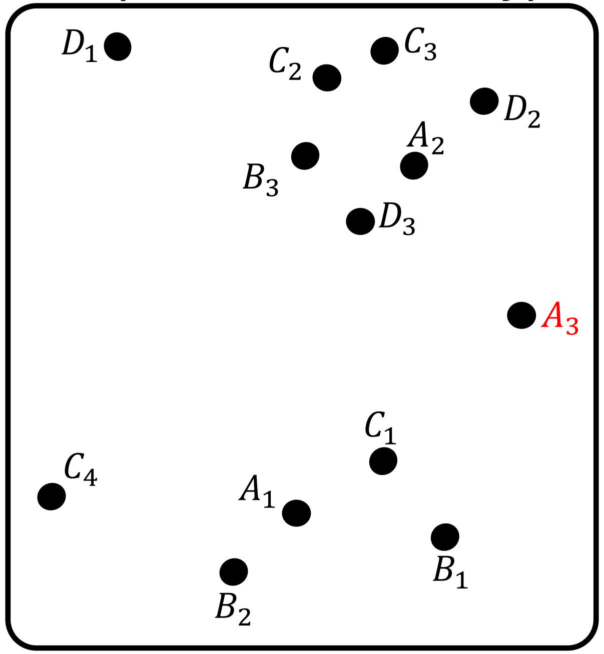


$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:

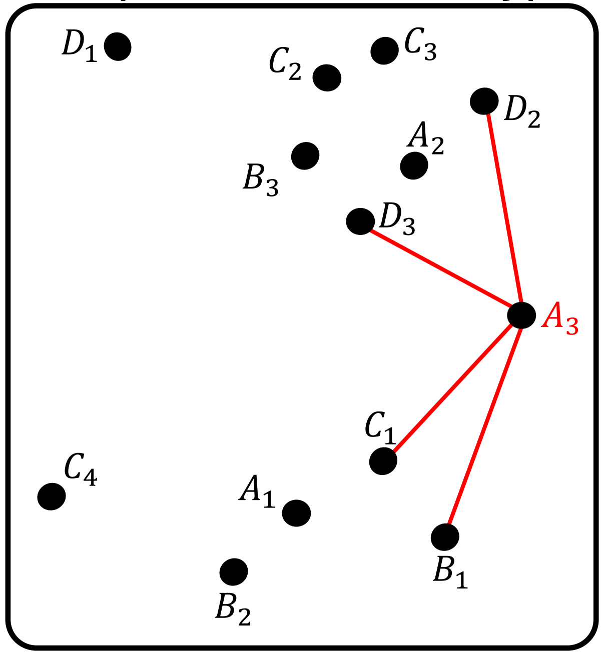


$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



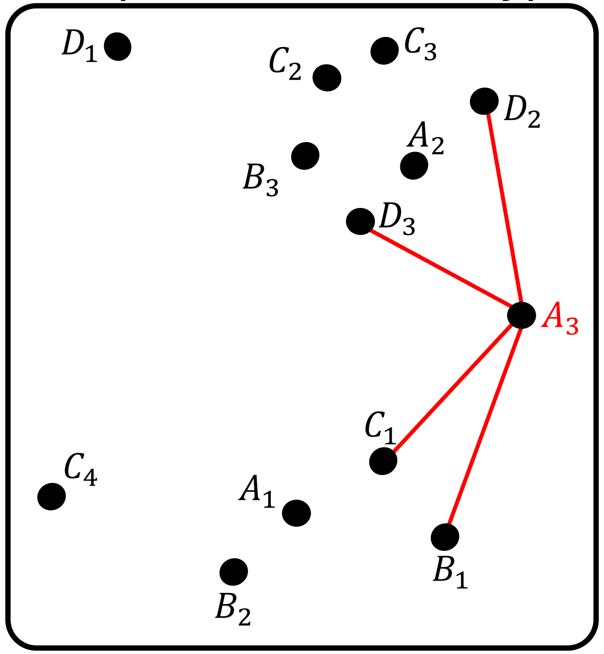
$K_{max} = \sqrt{13} \approx 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], [|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots]$$

Example with feature type A:



$$K_{max} = \sqrt{13} \approx 4$$

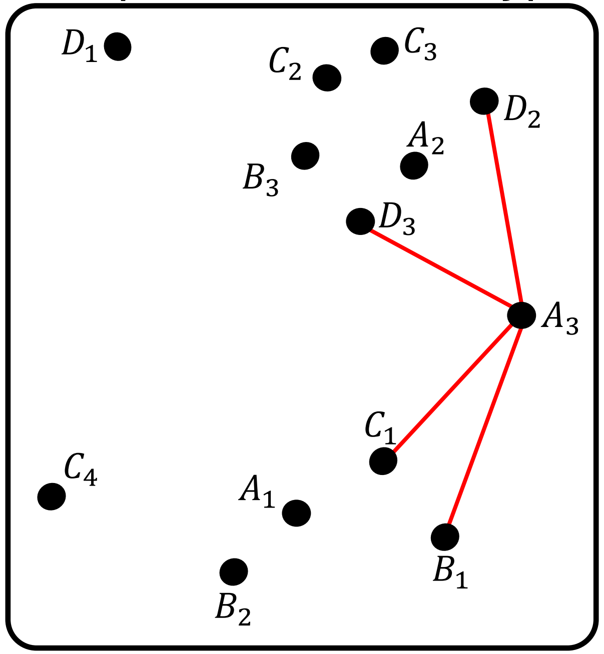
Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], [|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots]$$

shortest distance → longest distance

Example with feature type A:

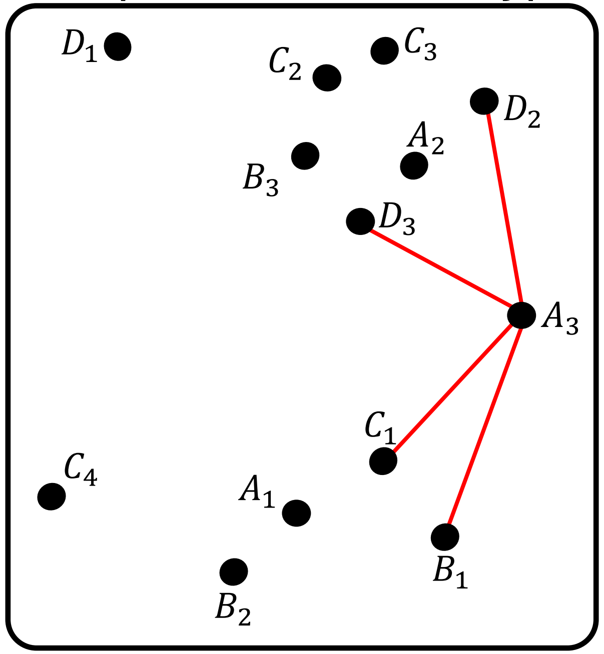


$$K_{max} = \sqrt{13} \approx 4$$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



$K_{max} = \sqrt{13} \approx 4$

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], [|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots]$$

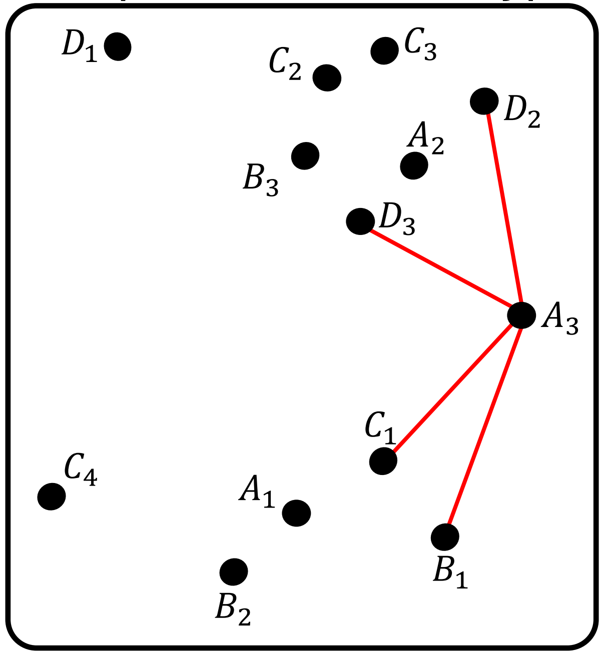
shortest distance → longest distance

$$T = [[0, 0, |A_1B_2| + |A_1C_1| + |A_1B_1|, |A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|], [0, 0, |A_2D_3| + |A_2D_2| + |A_2B_3|, |A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|], [0, 0, |A_3D_3| + |A_3C_1| + |A_3D_2|, |A_3D_3| + |A_3C_1| + |A_3D_2| + |A_3B_1|], \dots]$$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



$$K_{max} = \sqrt{13} \approx 4$$

$$\begin{aligned} D = [&[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], \\ &[|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], \\ &[|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots] \end{aligned}$$

shortest distance → longest distance

$$\begin{aligned} T = [&[0, 0, |A_1B_2| + |A_1C_1| + |A_1B_1|, |A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|], \\ &[0, 0, |A_2D_3| + |A_2D_2| + |A_2B_3|, |A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|], \\ &[0, 0, |A_3D_3| + |A_3C_1| + |A_3D_2|, |A_3D_3| + |A_3C_1| + |A_3D_2| + |A_3B_1|], \dots] \end{aligned}$$

$k = 3$

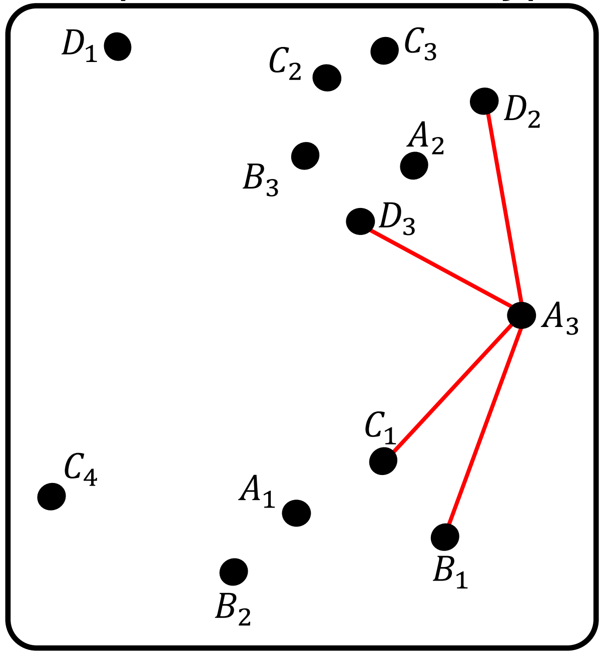
Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], [|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \ldots]$$

shortest distance → longest distance

Example with feature type A:



$$K_{max} = \sqrt{13} \approx 4$$

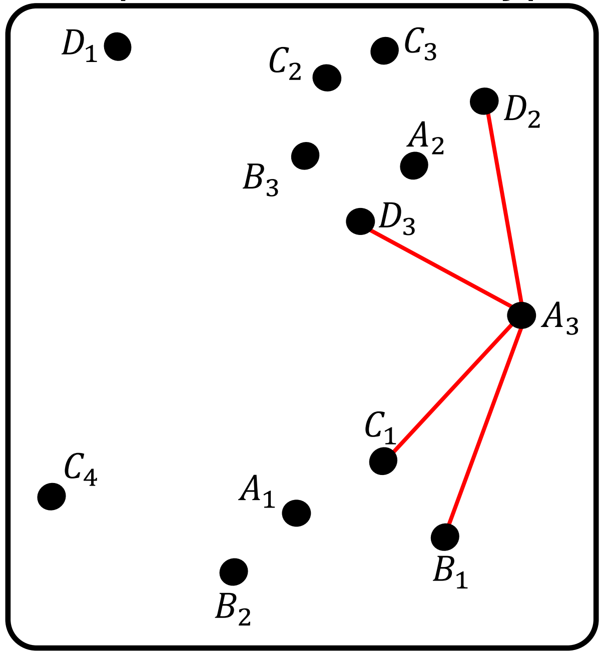
$$T = [[0, 0, \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1|}, \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|}], [0, 0, \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3|}, \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|}], [0, 0, \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2|}, \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2| + |A_3B_1|}], \ldots]$$

$k = 3$ $\quad$ $k = 4$

Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



$K_{max} = \sqrt{13} \approx 4$

$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|],$$
$$[|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|],$$
$$[|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots]$$

shortest distance → longest distance

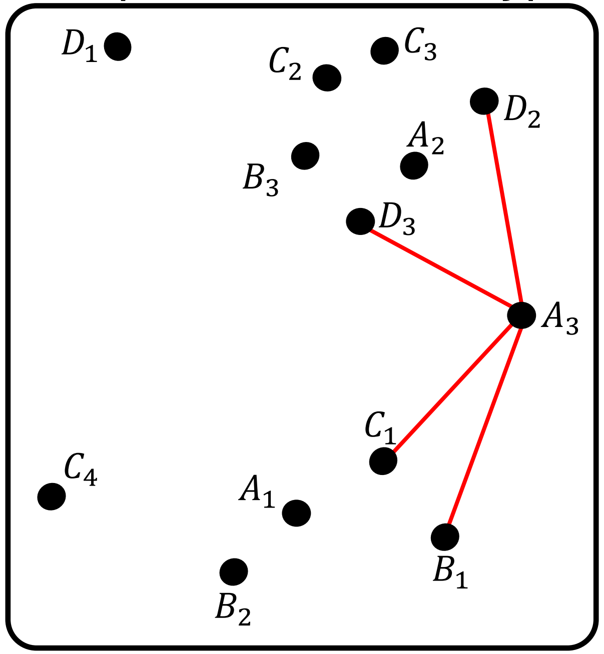
$$T = [[0, 0, \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1|}, \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|}],$$
$$[0, 0, \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3|}, \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|}],$$
$$[0, 0, \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2|}, \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2| + |A_3B_1|}], \dots]$$

$k = 3$ calculate average $k = 4$

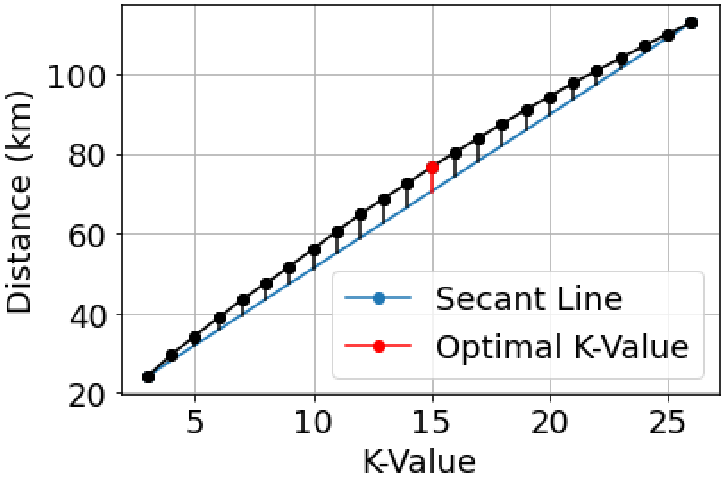
Our Distance Estimation Approach

- Major Steps:
  - Calculate distance
    - R-Tree
    - Dynamic Programming Table
  - Estimate optimal k-value
    - Knee Method

Example with feature type A:



$$K_{max} = \sqrt{13} \approx 4$$



$$D = [[|A_1B_2|, |A_1C_1|, |A_1B_1|, |A_1C_4|], \\ [|A_2D_3|, |A_2D_2|, |A_2B_3|, |A_2C_2|], \\ [|A_3D_3|, |A_3C_1|, |A_3D_2|, |A_3B_1|], \dots]$$

shortest distance → longest distance

$$T = [[0, 0, \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1|,} \boxed{|A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|],} \\ [0, 0, \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3|,} \boxed{|A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|],} \\ [0, 0, \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2|,} \boxed{|A_3D_3| + |A_3C_1| + |A_3D_2| + |A_3B_1|], \dots}]$$

$k = 3$ calculate average $k = 4$

Time Complexity

Nearest Neighbors

For each feature type F

$$O(NLog(N) + Mk(Log(N) + Log(k)))$$

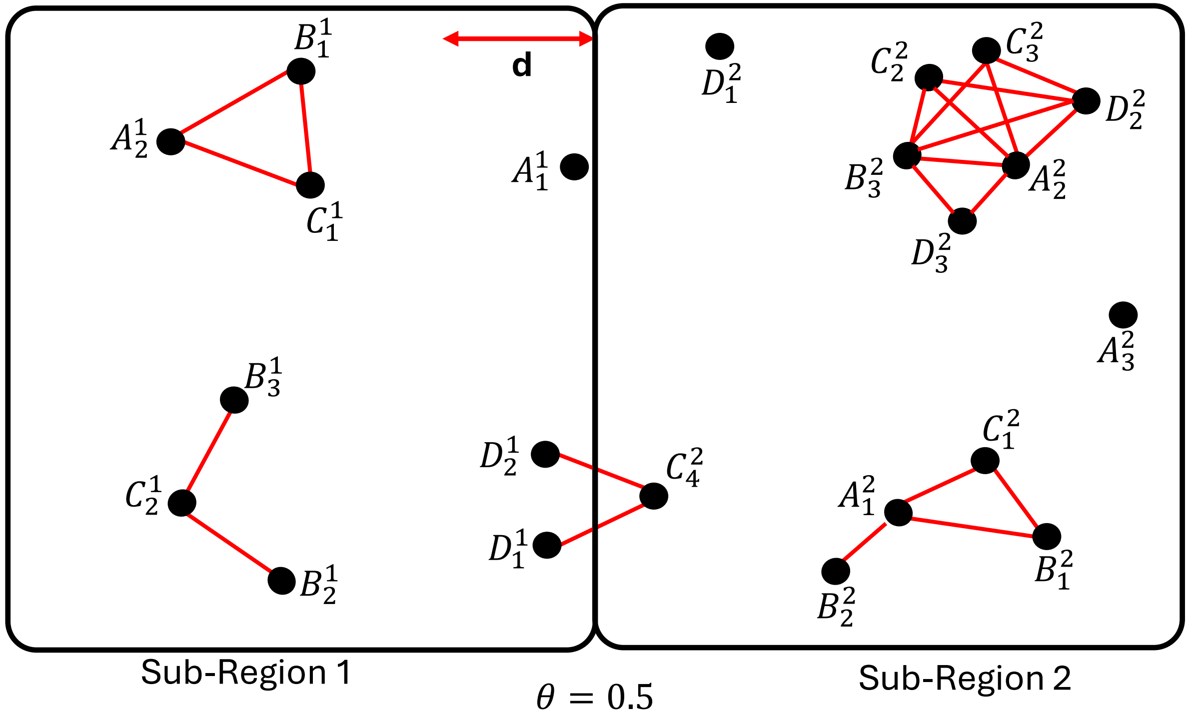
Dynamic Programming Table

$$O(I \times k)$$

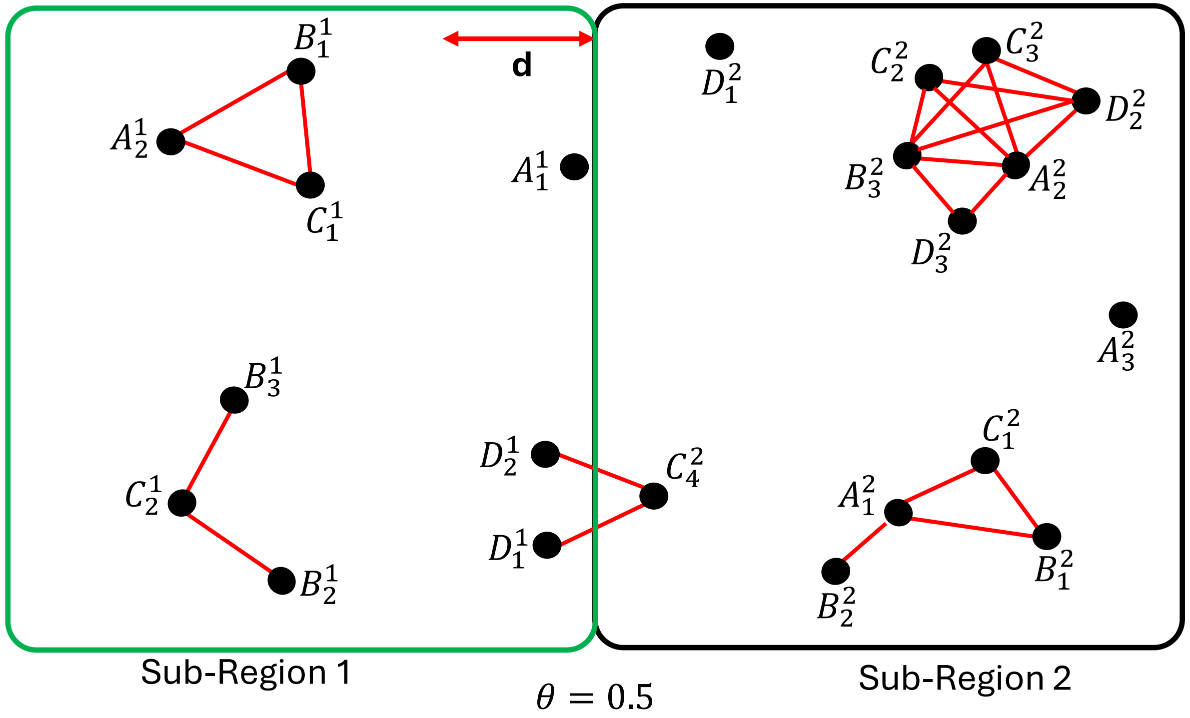
where

- I: total number of instances
- N: number of instances not of feature type F
- M: number of instances of feature type F
- k: $\sqrt{I}$

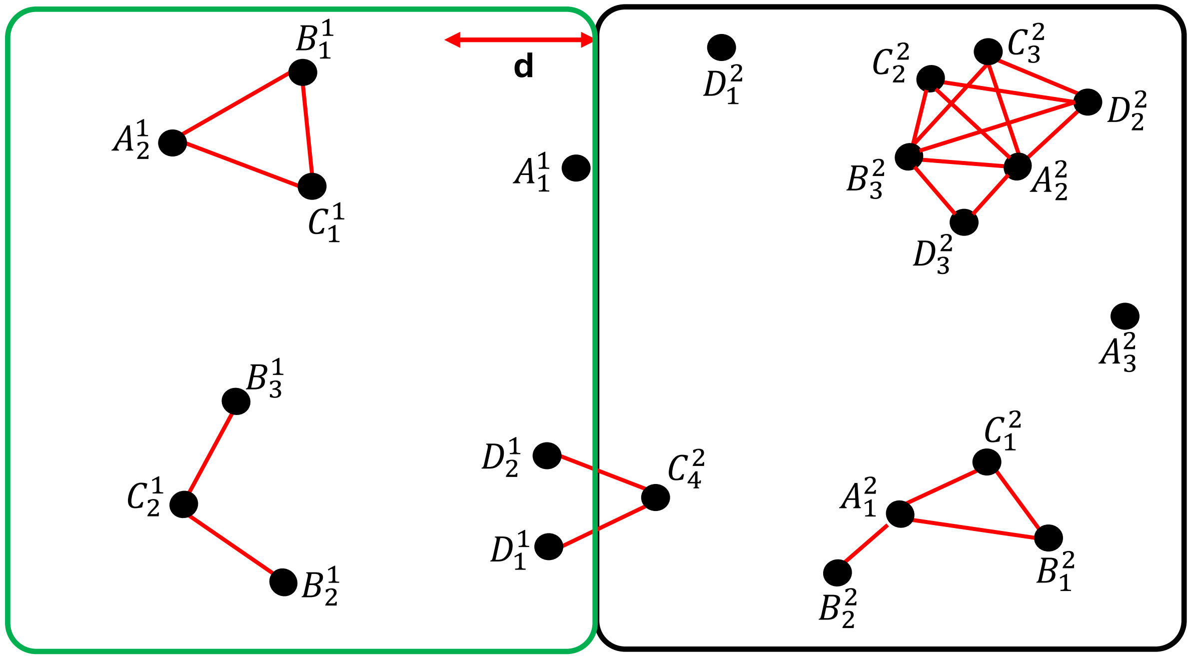
Our Map-Based Approach



Our Map-Based Approach



Our Map-Based Approach



Sub-Region 1 $\theta = 0.5$ Sub-Region 2

Sub-Region 1 Instance Table:

k = 1:

$(A) \rightarrow$	$\{A_1^1, A_2^1\}$	PI = 1
$(B) \rightarrow$	$\{B_1^1, B_2^1, B_3^1\}$	PI = 1
$(C) \rightarrow$	$\{C_1^1, C_2^1\}$	PI = 1
$(D) \rightarrow$	$\{D_1^1, D_2^1\}$	PI = 1

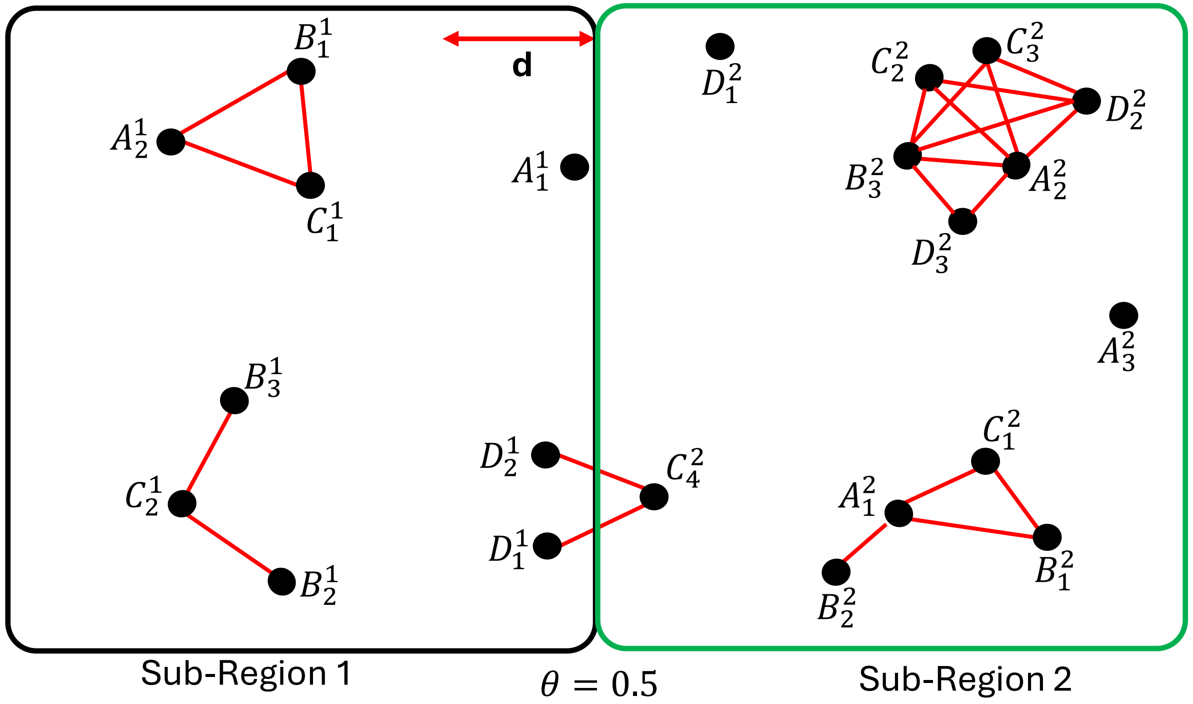
k = 2:

$(A, B) \rightarrow$	$\{(A_2^1) \rightarrow [B_1^1]\}$	PI = 0.33 ✖
$(A, C) \rightarrow$	$\{(A_2^1) \rightarrow [C_1^1]\}$	PI = 0.50
$(B, C) \rightarrow$	$\{(B_1^1) \rightarrow [C_1^1], (B_2^1) \rightarrow [C_2^1], (B_3^1) \rightarrow [C_2^1]\}$	PI = 1

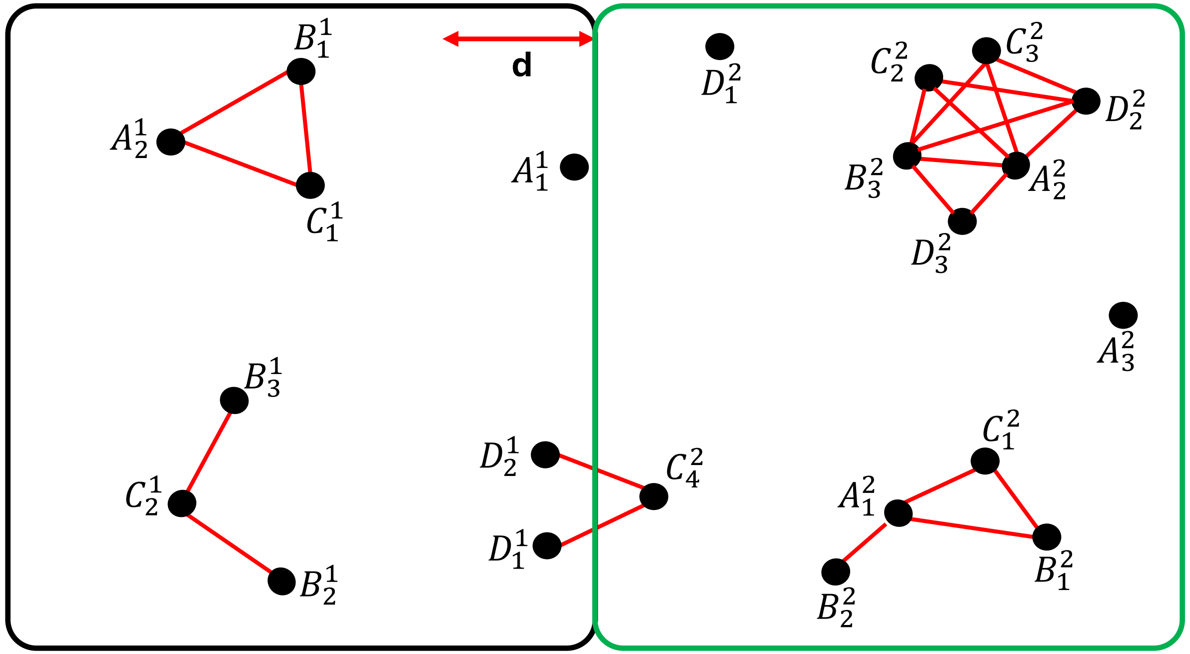
k = 3:

$(A, B, C) \rightarrow$	$\{(A_2^1, B_1^1) \rightarrow [C_1^1]\}$	PI = 0.33 ✖

Our Map-Based Approach



Our Map-Based Approach



Sub-Region 2 Instance Table:

k = 1:

$(A) \rightarrow \{A_1^2, A_2^2, A_3^2\}$ PI = 1

$(B) \rightarrow \{B_1^2, B_2^2, B_3^2\}$ PI = 1

$(C) \rightarrow \{C_1^2, C_2^2, C_3^2, C_4^2\}$ PI = 1

$(D) \rightarrow \{D_1^2, D_2^2, D_3^2\}$ PI = 1

k = 2:

$(A, B) \rightarrow \{(A_1^2) \rightarrow [B_1^2, B_2^2], (A_2^2) \rightarrow [B_3^2]\}$ PI = 0.66

$(A, C) \rightarrow \{(A_1^2) \rightarrow [C_1^2], (A_2^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.66

$(A, D) \rightarrow \{(A_2^2) \rightarrow [D_2^2, D_3^2]\}$ PI = 0.33 ✖

$(B, C) \rightarrow \{(B_1^2) \rightarrow [C_1^2], (B_3^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.66

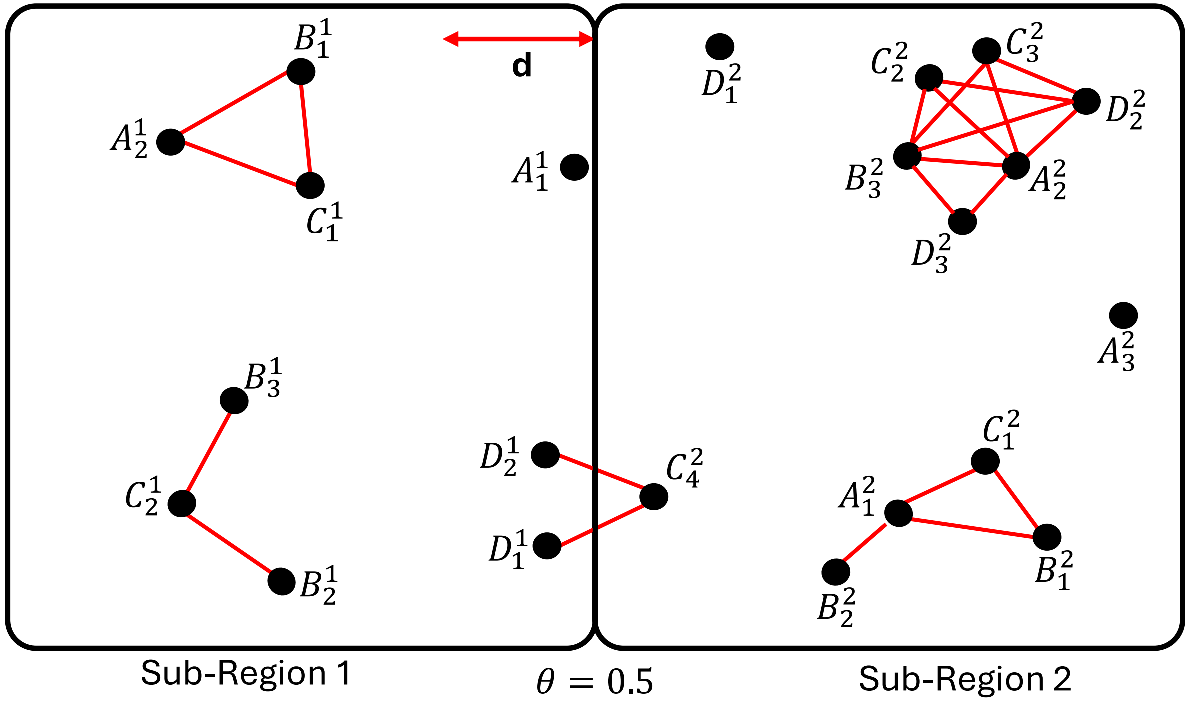
$(B, D) \rightarrow \{(B_3^2) \rightarrow [D_2^2, D_3^2]\}$ PI = 0.33 ✖

$(C, D) \rightarrow \{(C_2^2) \rightarrow [D_2^2], (C_3^2) \rightarrow [D_2^2]\}$ PI = 0.33 ✖

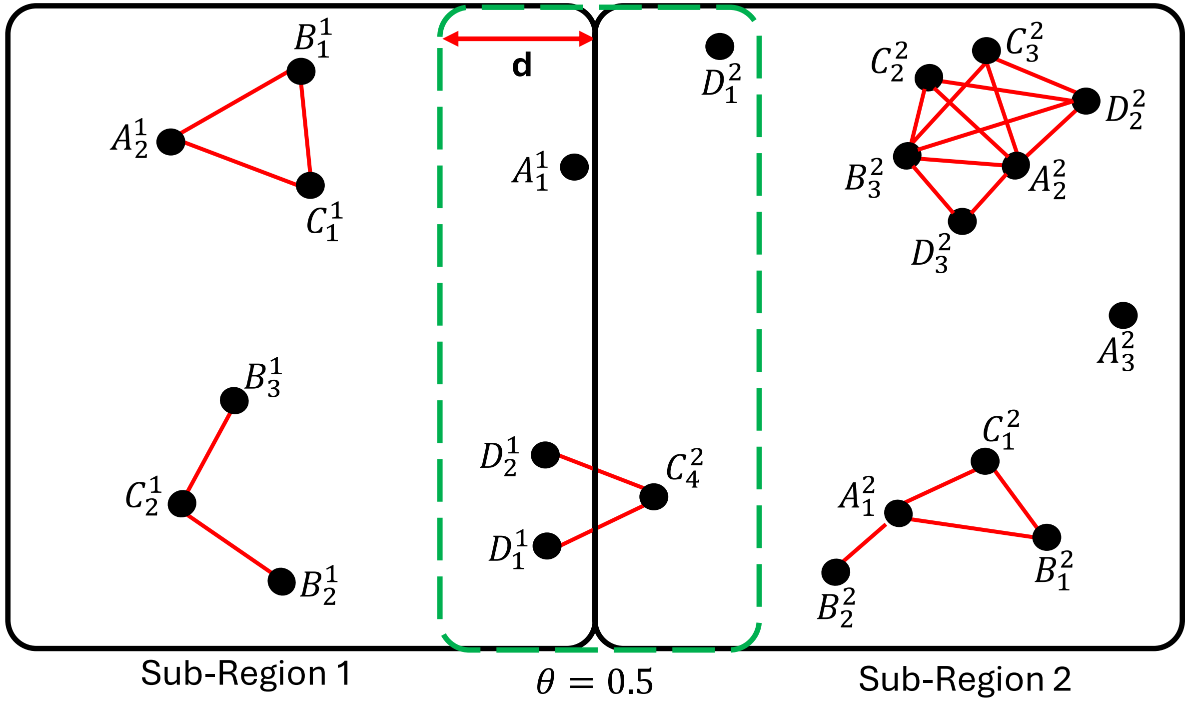
k = 3:

$(A, B, C) \rightarrow \{(A_1^2, B_1^2) \rightarrow [C_1^2], (A_2^2, B_3^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.66

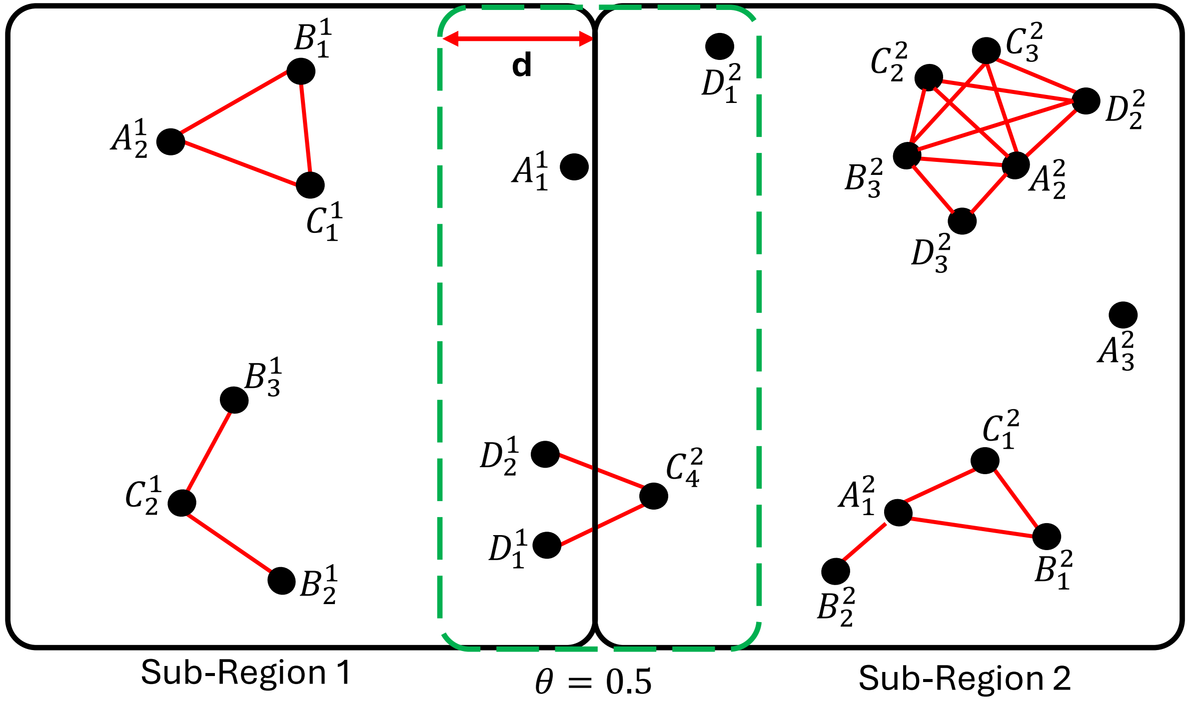
Our Map-Based Approach



Our Map-Based Approach



Our Map-Based Approach



Border Instance Table:

k = 1:

$(A) \rightarrow \{A_1^1\}$

$(C) \rightarrow \{C_4^2\}$

$(D) \rightarrow \{D_1^1, D_2^1, D_1^2\}$

k = 2:

$(C, D) \rightarrow \{(C_4^2) \rightarrow [D_1^1, D_2^1]\}$ PI = 0.66

Our Map-Based Approach

Sub-Region 1 Instance Table:

k = 2:

$(A,B) \rightarrow \{(A_2^1) \rightarrow [B_1^1]\}$

$(A,C) \rightarrow \{(A_2^1) \rightarrow [C_1^1]\}$

$(B,C) \rightarrow \{(B_1^1) \rightarrow [C_1^1],$
$(B_2^1) \rightarrow [C_2^1],$
$(B_3^1) \rightarrow [C_2^1]\}$

Border Instance Table:

k = 2:

$(C,D) \rightarrow \{(C_4^2) \rightarrow [D_1^1, D_2^1]\}$

Sub-Region 2 Instance Table:

k = 2:

$(A,B) \rightarrow \{(A_1^2) \rightarrow [B_1^2, B_2^2],$
$(A_2^2) \rightarrow [B_3^2]\}$

$(A,C) \rightarrow \{(A_1^2) \rightarrow [C_1^2],$
$(A_2^2) \rightarrow [C_2^2, C_3^2]\}$

$(A,D) \rightarrow \{(A_2^2) \rightarrow [D_2^2, D_3^2]\}$

$(B,C) \rightarrow \{(B_1^2) \rightarrow [C_1^2],$
$(B_3^2) \rightarrow [C_2^2, C_3^2]\}$

$(B,D) \rightarrow \{(B_3^2) \rightarrow [D_2^2, D_3^2]\}$

$(C,D) \rightarrow \{(C_2^2) \rightarrow [D_2^2],$
$(C_3^2) \rightarrow [D_2^2]\}$

Our Map-Based Approach

Sub-Region 1 Instance Table:

k = 2:

$(A,B) \rightarrow \{(A_2^1) \rightarrow [B_1^1]\}$

$(A,C) \rightarrow \{(A_2^1) \rightarrow [C_1^1]\}$

$(B,C) \rightarrow \{(B_1^1) \rightarrow [C_1^1],$
$(B_2^1) \rightarrow [C_2^1],$
$(B_3^1) \rightarrow [C_2^1]\}$

Border Instance Table:

k = 2:

$(C,D) \rightarrow \{(C_4^2) \rightarrow [D_1^1, D_2^1]\}$

Sub-Region 2 Instance Table:

k = 2:

$(A,B) \rightarrow \{(A_1^2) \rightarrow [B_1^2, B_2^2],$
$(A_2^2) \rightarrow [B_3^2]\}$

$(A,C) \rightarrow \{(A_1^2) \rightarrow [C_1^2],$
$(A_2^2) \rightarrow [C_2^2, C_3^2]\}$

$(A,D) \rightarrow \{(A_2^2) \rightarrow [D_2^2, D_3^2]\}$

$(B,C) \rightarrow \{(B_1^2) \rightarrow [C_1^2],$
$(B_3^2) \rightarrow [C_2^2, C_3^2]\}$

$(B,D) \rightarrow \{(B_3^2) \rightarrow [D_2^2, D_3^2]\}$

$(C,D) \rightarrow \{(C_2^2) \rightarrow [D_2^2],$
$(C_3^2) \rightarrow [D_2^2]\}$

Regional Instance Table:

k = 2:

$(A,B) \rightarrow \{(A_2^1) \rightarrow [B_1^1], (A_1^2) \rightarrow [B_1^2, B_2^2],$
$(A_2^2) \rightarrow [B_3^2]\}$ PI = 0.60

$(A,C) \rightarrow \{(A_2^1) \rightarrow [C_1^1], (A_1^2) \rightarrow [C_1^2], (A_2^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.60

$(B,C) \rightarrow \{(B_1^1) \rightarrow [C_1^1], (B_2^1) \rightarrow [C_2^1],$
$(B_3^1) \rightarrow [C_2^1], (B_1^2) \rightarrow [C_1^2],$
$(B_3^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.83

$(C,D) \rightarrow \{(C_2^2) \rightarrow [D_2^2], (C_3^2) \rightarrow [D_2^2], (C_4^2) \rightarrow [D_1^1, D_2^1]\}$ PI = 0.50

Note: *(A,D) and (B,D) are not interesting in neither of the sub-regions and the border region. Hence, pruned.*

$\theta = 0.5$

Time Complexity

For each colocation pattern C_k

$$O\left(\left|I_{k-1}\right|\left(kLog(M) + N(Log(M) + k)\right)\right)$$

where

- k: cardinality of colocation pattern C_k
- I_{k-1}: average number of entries in instance table of previous degree
- M: average length of star neighborhood for each instance
- N: average number of neighbors for each key combination

Lemma 1

$Let\ R\ be\ a\ region\ with\ n\ sub\text{-}regions\ s_1, s_2, \ldots, s_n.$
$Let\ C\ be\ a\ colocation\ pattern\ such\ that\ PI(C) \geq \theta\ \forall s_1, s_2, \ldots, s_n$
$where\ \theta\ is\ the\ prevalence\ threshold.\ Then, PI(C) \geq \theta\ for\ R.$

Lemma 1

$Let\ R\ be\ a\ region\ with\ n\ sub\text{-}regions\ s_1, s_2, \dots, s_n.$
$Let\ C\ be\ a\ colocation\ pattern\ such\ that\ PI(C) \geq \theta\ \forall s_1, s_2, \dots, s_n$
$where\ \theta\ is\ the\ prevalence\ threshold.\ Then, PI(C) \geq \theta\ for\ R.$

Proof:

Let f_i be a feature of cardinality k colocation pattern $C = (f_1, f_2, \dots, f_k)$.

Denote $I^s = \{I_{f_i}^{s_1}, \dots, I_{f_i}^{s_n}\}$ as a set of instances of feature f_i participating in C in sub-regions $s_1, s_2, \dots, s_n$.

The PR of each f_i of C in each sub-region is denoted $PR^{s_p}(C, f_i) = \frac{|TI^{s_p}(C, f_i^{s_p})|}{|I_{f_i}^{s_p}|}, \forall p \leq n$.

We know $\frac{|TI^{s_p}(C, f_i^{s_p})|}{|I_{f_i}^{s_p}|} \geq \theta$, so $\frac{\sum_{p=1}^{n} |TI^{s_p}(C, f_i^{s_p})|}{\sum_{p=1}^{n} |I_{f_i}^{s_p}} \geq \theta$.

So, $PR^R\left(C, f_i^R\right) = \frac{\sum_{p=1}^{n} |TI^{s_p}(C, f_i^{s_p})|}{\sum_{p=1}^{n} |I_{f_i}^{s_p}|} \geq \theta$.

Therefore, $PI^R(C) = \min\left(PR^R(C, f_1^R), PR^R(C, f_2^R), \dots, PR^R\left(C, f_k^R\right)\right) \geq \theta$.

Making C a prevalent pattern for the entire region R.

Lemma 2

$Let\ R\ be\ a\ region\ with\ n\ subregions\ s_1, s_2, \ldots, s_n, and\ m\ border\ regions$ $b_1, b_2, \ldots, b_m.\ A\ border\ region\ is\ an\ overlapping\ geographical\ area\ where$ $two\ subregions\ touch.\ Let\ C\ be\ a\ colocation\ pattern\ and\ f\ be\ the\ feature$ $in\ C\ such\ that\ PR(C, f) < \theta\ \forall s_1, s_2, \ldots, s_n\ and\ PR < \theta\ \forall b_1, b_2, \ldots, b_m.$ $Then, PI(C) < \theta\ for\ R.$

Lemma 2

$Let\ R\ be\ a\ region\ with\ n\ subregions\ s_1, s_2, \ldots, s_n, and\ m\ border\ regions\ b_1, b_2, \ldots, b_m.\ A\ border\ region\ is\ an\ overlapping\ geographical\ area\ where\ two\ subregions\ touch.\ Let\ C\ be\ a\ colocation\ pattern\ and\ f\ be\ the\ feature\ in\ C\ such\ that\ PR(C, f) < \theta\ \forall s_1, s_2, \ldots, s_n\ and\ PR(C, f) < \theta\ \forall b_1, b_2, \ldots, b_m.\ Then, PI(C) < \theta\ for\ R.$

Proof:

Denote $I_f^s = \{I_f^{s_1}, \ldots, I_f^{s_n}\}$ as a set of the instances of feature f participating in C in sub-regions $s_1, s_2, \ldots, s_n$.

Denote $I_f^b = \{I_f^{b_1}, \ldots, I_f^{b_m}\}$ as a set of the instances of feature f participating in C in border regions $b_1, b_2, \ldots, b_m$ where $I_f^{b_j}$ denotes the set of instances of f where at least two instances in the row instance of C occur in two distinct sub-regions.

The PR of f in C for each sub-region and border region is denoted

$PR^{s_p}(C, f) = \frac{|TI^{s_p}(C,f)|}{|I_f^{s_p}|}, \forall p \leq n$ and $PR^{b_j}(C, f) = \frac{|TI^{b_j}(C,f)|}{|I_f^{b_j}|}, \forall j \leq m$, respectively.

We know $\frac{|TI^{s_p}(C,f)|}{|I_f^{s_p}|} < \theta$ and $\frac{|TI^{b_j}(C,f)|}{|I_f^{b_j}|} < \theta$ for each sub-region and border region.

Lemma 2

Let R be a region with n subregions $s_1, s_2, \ldots, s_n$, and m border regions $b_1, b_2, \ldots, b_m$. A border region is an overlapping geographical area where two subregions touch. Let C be a colocation pattern and f be the feature in C such that $PR(C, f) < \theta \ \forall s_1, s_2, \ldots, s_n$ and $PR < \theta \ \forall b_1, b_2, \ldots, b_m$. Then, $PI(C) < \theta$ for R.

Proof (cont.):

So, $|TI^{s_p}(C, f)| < \theta |I_f^{s_p}|$ and $\left|TI^{b_j}(C, f)\right| < \theta \left|I_f^{b_j}\right|$

$\Rightarrow \sum_{p=1}^{n} |TI^{s_p}(C, f)| < \theta \sum_{p=1}^{n} |I_f^{s_p}|$ and $\sum_{j=1}^{m} \left|TI^{b_j}(C, f)\right| < \theta \sum_{j=1}^{m} |I_f^{b_j}|$.

So, $\sum_{p=1}^{n} |TI^{s_p}(C, f)| + \sum_{j=1}^{m} \left|TI^{b_j}(C, f)\right| < \theta (\sum_{p=1}^{n} |I_f^{s_p}| + \sum_{j=1}^{m} |I_f^{b_j}|)$

$$\Rightarrow PR^R(C, f) = \frac{\sum_{p=1}^{n} |TI^{s_p}(C, f)| + \sum_{j=1}^{m} \left|TI^{b_j}(C, f)\right|}{\sum_{p=1}^{n} |I_f^{s_p}| + \sum_{j=1}^{m} |I_f^{b_j}|} < \theta.$$

Therefore, $PI^R(C) < \theta$, making C not a prevalent pattern in R.

Theorem

$The\ regional\ colocation\ pattern\ calculation\ framework\ is\ correct\ and\ complete.$

Proof:

The algorithm is complete and does not mistakenly prune out any prevalent patterns due to Lemma 2.

Theorem

$The\ regional\ colocation\ pattern\ calculation\ framework\ is\ correct\ and\ complete.$

Proof:

The algorithm is complete and does not mistakenly prune out any prevalent patterns due to Lemma 2.

The algorithm is correct since it computes the exact participation index of each candidate pattern in each sub-region and the entire region.

Evaluation

- Goals
  - Evaluate the difference in spatial neighborhood relationship constraints across 3 case study regions
  - Compare the memory optimization percentage of our proposed map-based approach with the array-based approach

Data Set

Real World

Synthetic

Data Set

Real World

- Global Terrorism Database (https://www.start.umd.edu/gtd)
- Year: 1970-2020
- Instances: 215k
- Number of Attack Types: 8

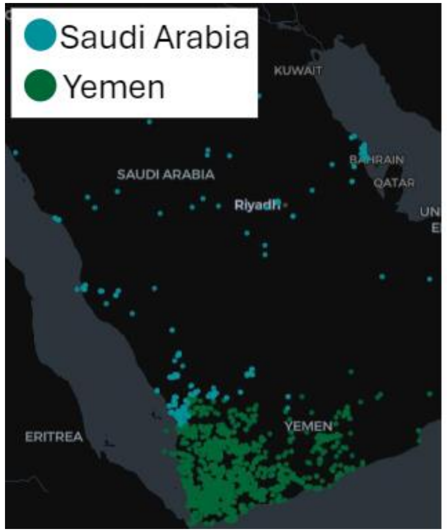
Synthetic

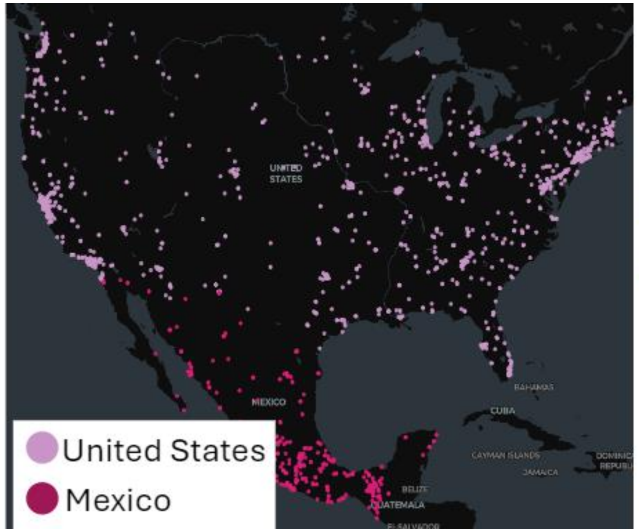
Data Set

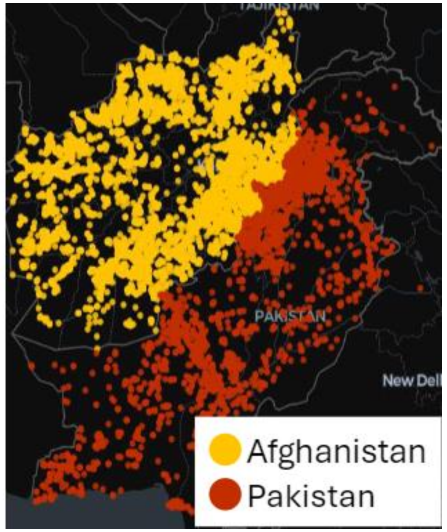
Real World

- Global Terrorism Database (https://www.start.umd.edu/gtd)
- Year: 1970-2020
- Instances: 215k
- Number of Attack Types: 8

Synthetic







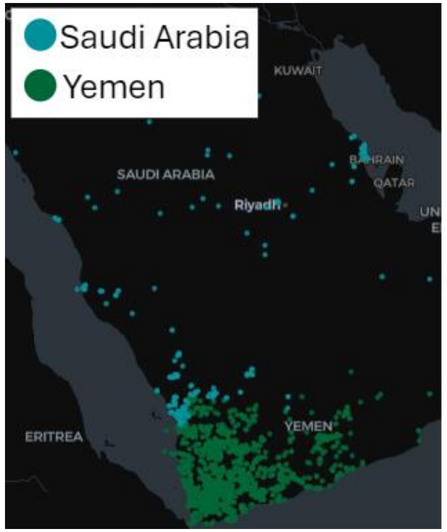
Data Set

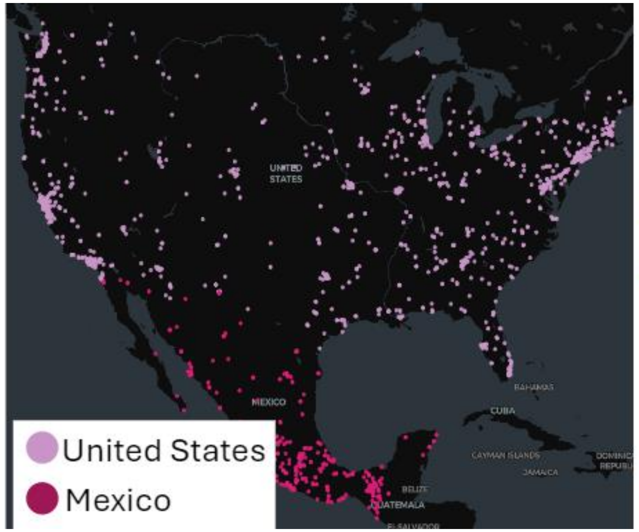
Real World

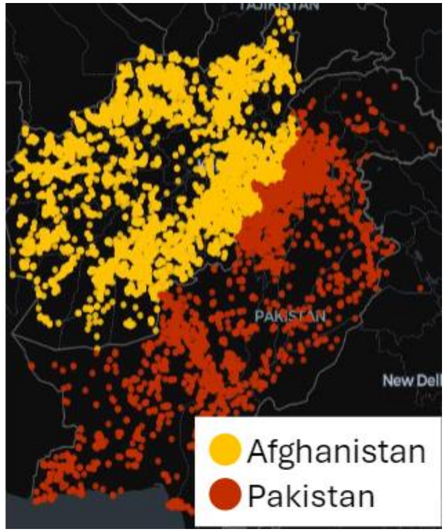
- Global Terrorism Database (https://www.start.umd.edu/gtd)
- Year: 1970-2020
- Instances: 215k
- Number of Attack Types: 8

Synthetic

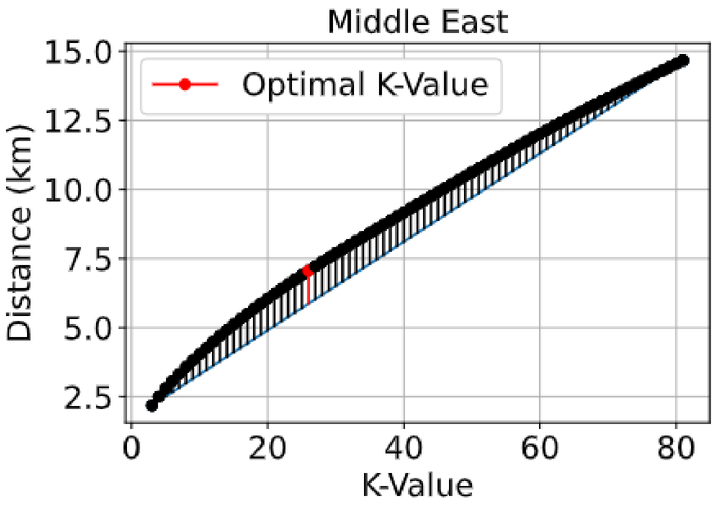
- Pre-generate all final prevalent patterns
- Based on (Colocation Mining: A General Approach) [Huang, 2004]
- Varying Clumpiness



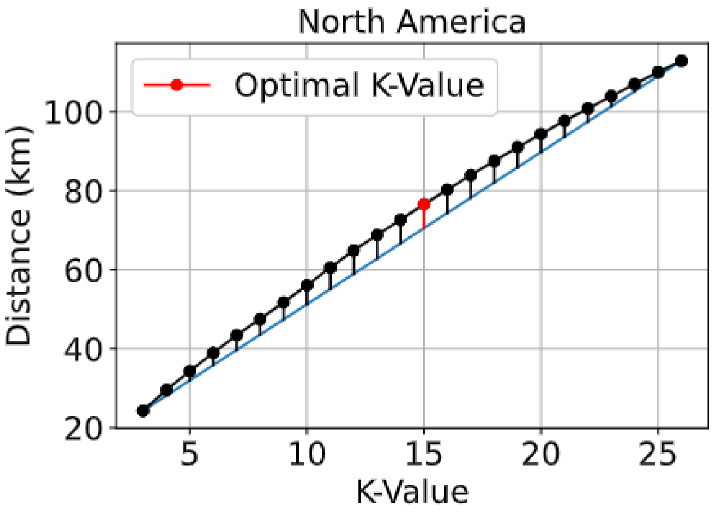




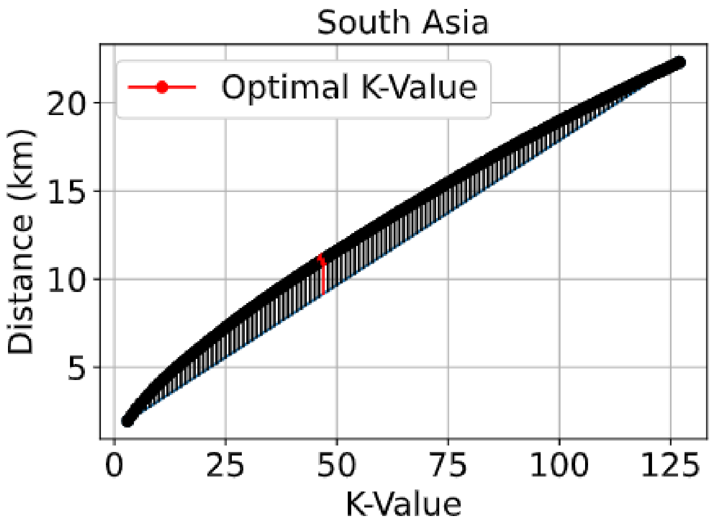
Results: Real-World Data Set



(a)



(b)



(c)

Observation:

- Different spatial neighborhood relationship constraint for each region

Results: Real-World Data Set

Area	Interesting Patterns
Saudi Arabia	(0, 2)
Yemen	(0, 1, 2), (0, 1, 6), (1, 2, 6)
Middle East	(1, 3), (0, 1, 2, 6)
United States	(2, 5), (3, 7), (1, 2, 7)
Mexico	(0, 6), (2, 6), (3, 5)
North America	(1, 6), (0, 1, 2), (0, 2, 3)
Afghanistan	(3, 6), (1, 2, 3), (2, 3, 6)
Pakistan	(1, 4), (1, 5), (0, 2, 3)
South Asia	(0, 1, 2, 3), (0, 1, 2, 6), (0, 2, 3, 6)

Attack Type	Identifier
Armed Assault	0
Assassination	1
Bombing	2
Facility Attack	3
Hijacking	4
Hostage Taking (Barricade)	5
Hostage Taking (Kidnapping)	6
Unarmed Assault	7

Observation:

- Different interesting patterns in each region

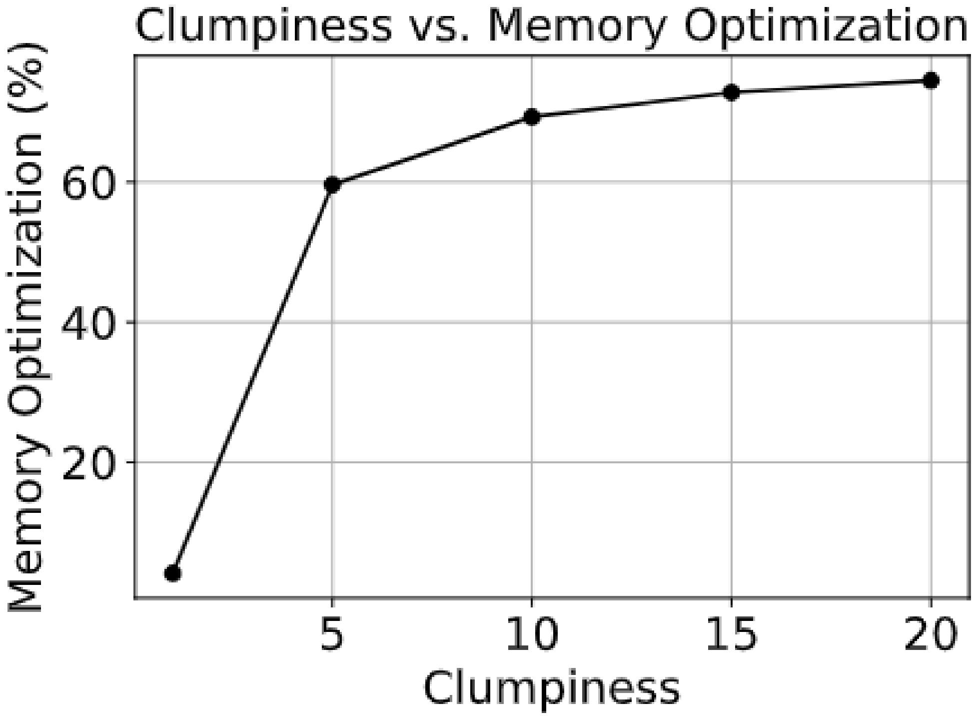
Results: Real-World Data Set

Region	Degree	Map-Based Approach	Array-Based Approach	Memory Optimization
Middle East (1)	2	0.0017 GB	0.0033 GB	48.5%
	3	0.1178 GB	0.3408 GB	65.4%
	4	3.4795 GB	13.0831 GB	73.4%
	Total	3.5990 GB	13.4272 GB	73.2%
North America (2)	2	0.0006 GB	0.0011 GB	48.2%
	3	0.0134 GB	0.0395 GB	65.9%
	Total	0.0140 GB	0.0406 GB	65.5%
South Asia (3)	2	0.0185 GB	0.0362 GB	49.0%
	3	4.0720 GB	11.9411 GB	65.9%
	4	438.4528 GB	1662.6944 GB	73.6%
	Total	442.5433 GB	1674.6717 GB	73.6%

Observation:

- The higher the degree, the higher the memory optimization
- Our approach uses approximately 70% less memory than pre-existing array-based approaches

Results: Synthetic Data Set



$$\frac{Array\ Approach\ - Map\ Approach}{Array\ Approach} \times 100$$

Observation:

- The clumpier the data, the higher the memory optimization percentage
- Our approach uses approximately 70% less memory than pre-existing array-based approaches

Future Work

- Further memory optimization in higher degrees
- Differing sub-regional spatial neighborhood relationship constraints

Thank you

Appendix

Algorithm 2 Dynamic Neighborhood Relationship Estimate

Input: A set of spatial features F
Input: Instances of each spatial feature $I[F]$
Output: Neighborhood relationship constraint d

Initialize $D \leftarrow \emptyset$, $K_{max} \leftarrow \sqrt{|I[F]|} + 1$, $A \leftarrow \emptyset$
Initialize memoization table $T \leftarrow \emptyset$ of size $|I[F]| \times K_{max}$
for f **in** F **do**
  $S_f \leftarrow$ EXTRACT_BY_FEATURE_TYPE(f) $\cancel{2I}$
  $I[F]_{Excludingfeature} \leftarrow I[F] - S_f$
  $r \leftarrow$ RTREE()
  ADD_POINTS_TO_RTREE$(r, I[F]_{Excludingfeature})$ $NLog(N)$
  for p in S_f **do** M
    $x \leftarrow p[0]$, $y \leftarrow p[1]$
    $N = r$.nearest$((x, y), K_{max})$ $kLog(N)$
    D.append(SORT_NEIGHBORS_DISTANCES(N)) $kLog(k)$
  end for
end for

For each feature type F

$$O(NLog(N) + Mk(Log(N) + Log(k)))$$

$T[:, 2] =$ ROWWISE_SUM$(D[:, :3])$ $\cancel{I}$
for i **in** range$(|I[F]|$ **do** I
  for k **in** range$(3, K_{max})$ **do** k
    $T[i, k] = T[i, k-1] + D[i][k]$
  end for
end for
$C =$ COLUMNWISE_SUM(T) $I \times k$
for k **in** range$(2, K_{max})$ **do** $\cancel{k}$
  A.append$(C[k]/(|I[F]| \times (k+1)))$
end for
$d \leftarrow$ KNEE_METHOD(A) $\cancel{k}$
return d

$$O(I \times k)$$

where

- I: total number of instances
- N: number of instances not of feature type F
- M: number of instances of feature type F
- k: $\sqrt{I}$

Algorithm 3 Map-Based Instance Table Pattern Calculation

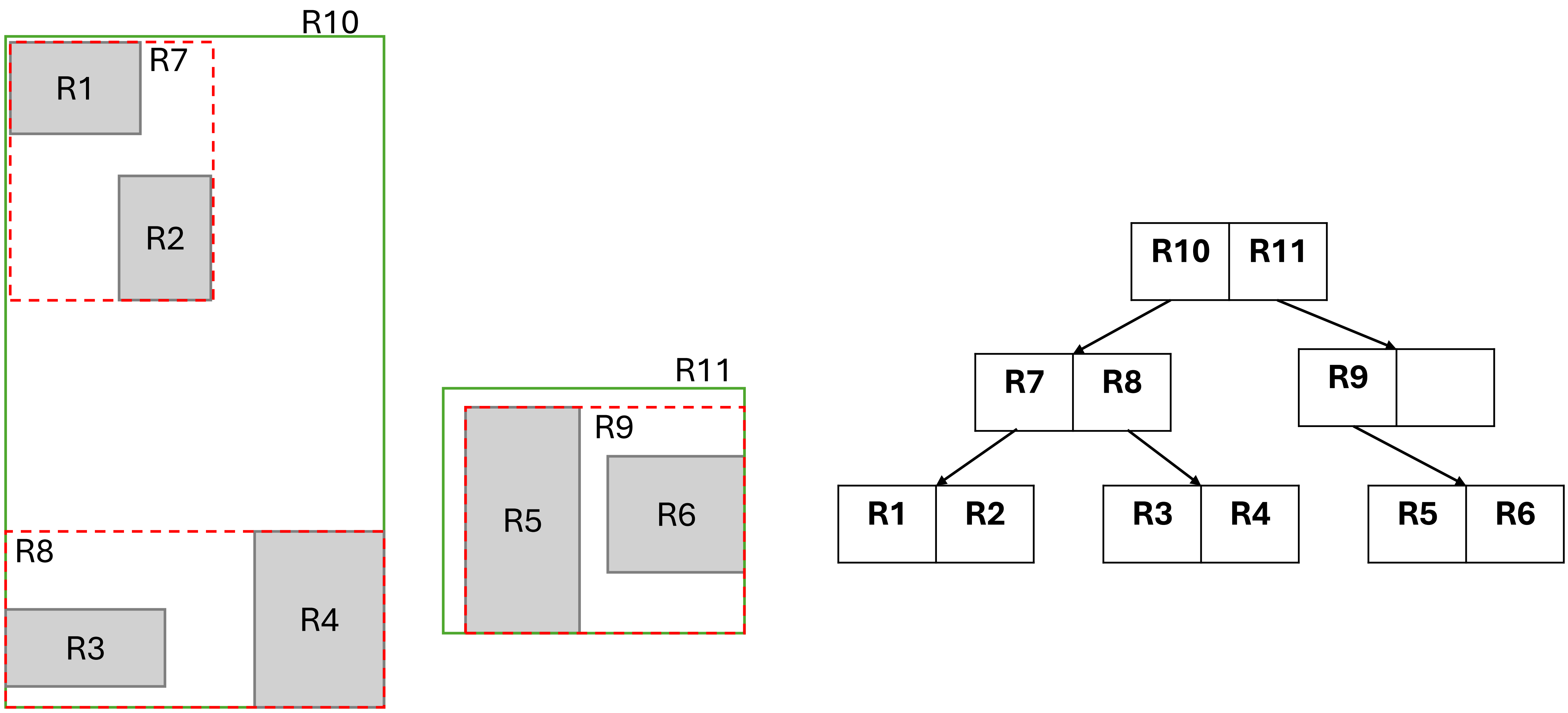
Input: List of candidate patterns C_k of size k
Input: Instance table I_{k-1} for all size $k-1$ patterns
Input: Hash Map that holds the starting and ending indices and instance count of each feature F_{info}
Input: Star neighbors of instances of each spatial feature S
Output: Filled in instance table I_k
Output: List of prevalent patterns P_k

Initialize $P_k \leftarrow \emptyset, I_k \leftarrow \emptyset, H \leftarrow \emptyset$
for c in C_k **do**
  $B_{pattern} \leftarrow c[0 : k-1], L_f \leftarrow c[k-1]$
  Initialize $I_k[c] \leftarrow \emptyset, H[c] \leftarrow \emptyset$
  $H[c] \leftarrow \{f : \emptyset$ **for** f **in** $c\}$
  $I_{base} \leftarrow I_{k-1}[B_{pattern}]$
  for key **in** I_{base} **do** $|I_{k-1}|$
    $N \leftarrow \emptyset$
    for id **in** key **do** $k\cancel{-1}$
      if not N **then**
        $N \leftarrow \text{NEIGHBOR}(S[id], F_{info}[L_f])$ $Log(M)$
      else
        $N \leftarrow N \cap \text{NEIGHBOR}(S[id], F_{info}[L_f])$
      end if
    end for
    for i in $I_{base}[key]$ **do** N
      $n = N \cap \text{NEIGHBOR}(S[i], F_{info}[L_f])$ $Log(M)$
      if n **then**
        $key_{new} = key.\text{append}(i)$
        $I_k[c][key_{new}] \leftarrow n$
        for $j \in key_{new}$ **do** k
          $f \leftarrow \text{GET\_FEATURE\_ID}(j)$
          $H[c][f].\text{add}(j)$
        end for
        $H[c][L_f].\text{update}(n)$
      end if
    end for
  end for
  $PR \leftarrow \emptyset$
  for f **in** c **do**
    $PR.\text{append}(|H[c][f]|/F_{info}[f].count)$
  end for
  $PI = \min(PR)$
  $P_k.\text{append}(c)$ **if** $PI \geq \theta$
end for
return P_k, I_k

For each colocation pattern C_k, $O\big(|I_{k-1}|\big(kLog(M) + N(Log(M) + k)\big)\big)$

where

k: cardinality of colocation pattern C_k, I_{k-1}: average number of entries in instance table of previous degree, M: average length of star neighborhood for each instance, N: average number of neighbors for each key combination


R10
R7
R1
R2
R8
R3
R4
R11
R9
R5
R6
R10
R11
R7
R8
R9
R1
R2
R3
R4
R5
R6