

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

Abigail Kelly

Ph.D. Qualifying Exam

Advisor: Dr. Sainju

Department of Computational and Data Science

Middle Tennessee State University

Motivation

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



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
- 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)
 - Spatial neighborhood relationship constraint
 - Any 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

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

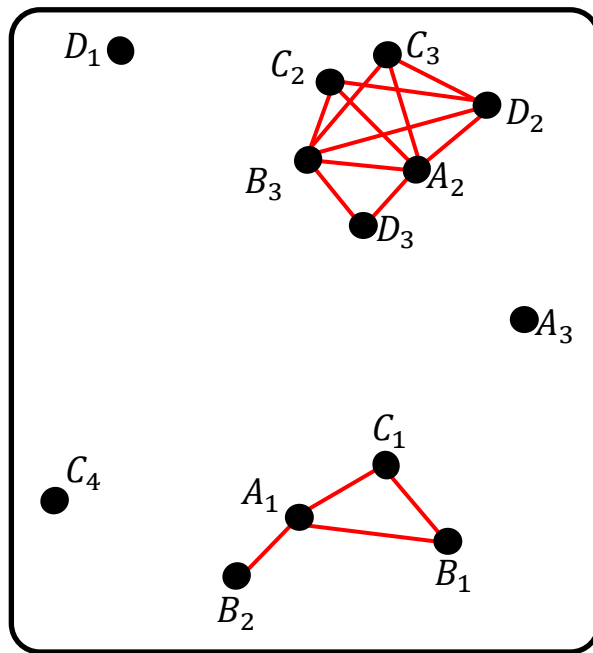
Map-Based

- Map-like structure with key-value pairs
- Keys
 - 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

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: Map-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

$$\Rightarrow \begin{aligned} PR((A, B), A) &= 2/3 \Rightarrow 0.67 \\ PR((A, B), B) &= 3/3 \Rightarrow 1 \\ PI &= 0.67 \end{aligned}$$

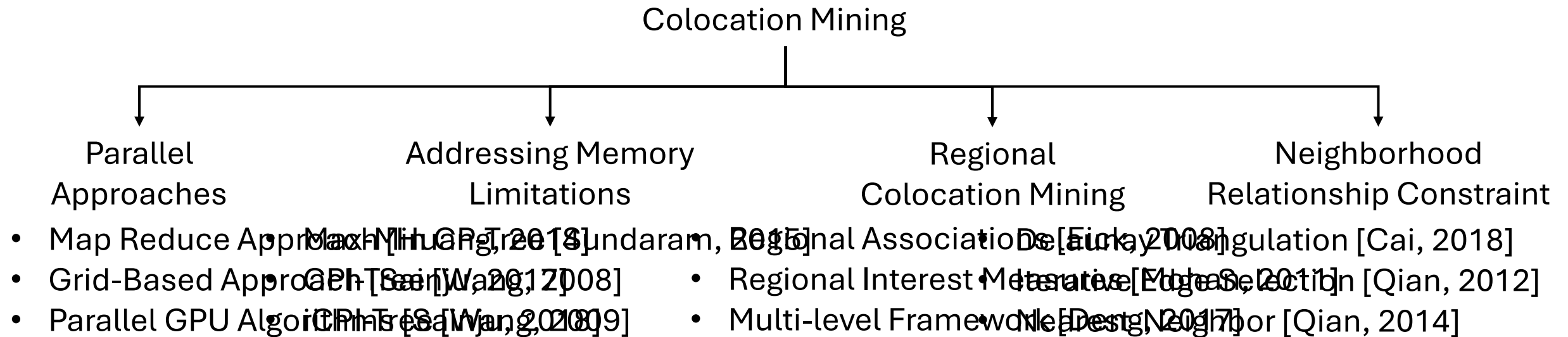
Problem Definition

- 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
- Number of candidate colocation patterns can grow exponentially
- Storing the intermediate collection of collocated instances for each pattern
- Determining neighborhood relationship constraint

Related Work

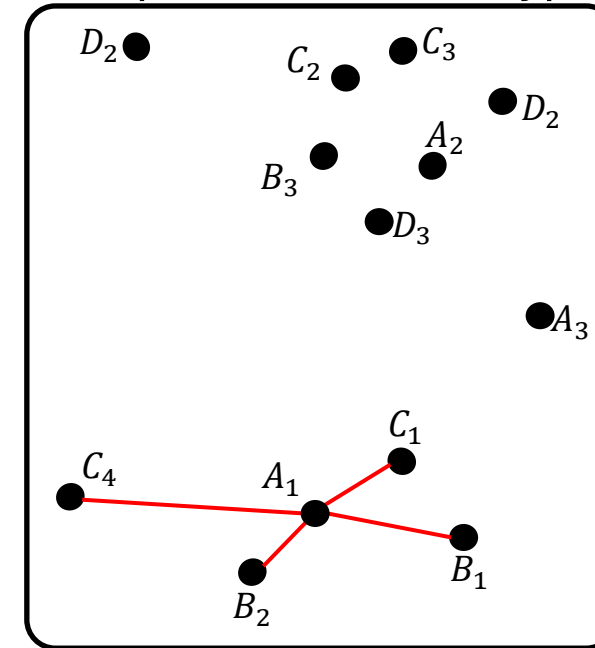


Our Distance Estimation Approach

- 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 \rightarrow longest distance

Example with feature type A :



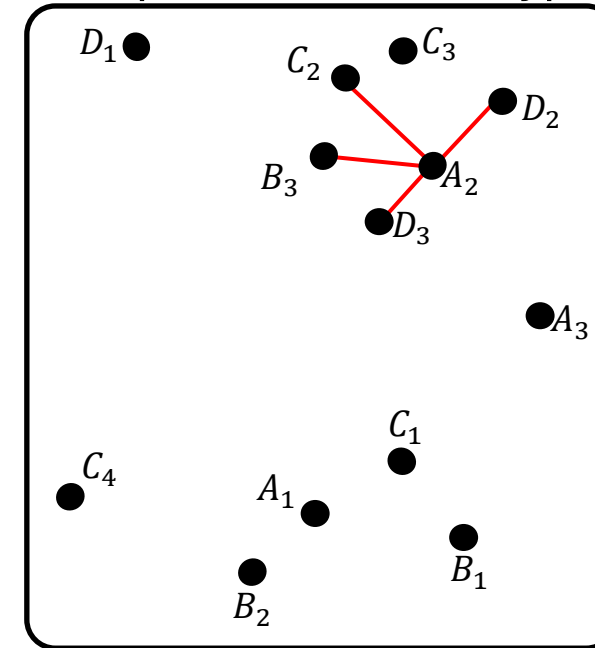
$$K_{max} = \sqrt{13} \approx 4$$

Our Distance Estimation Approach

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

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

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

shortest distance \rightarrow longest distance

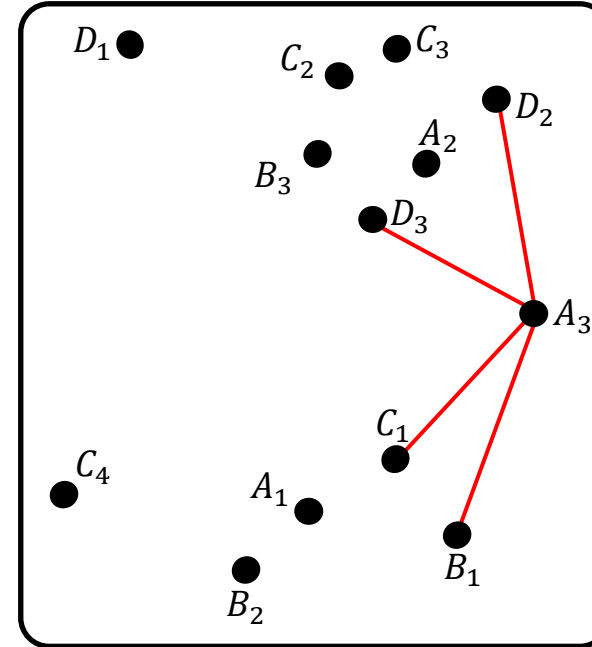
$$T = [[|A_1B_2|, |A_1B_2| + |A_1C_1|, |A_1B_2| + |A_1C_1| + |A_1B_1|, |A_1B_2| + |A_1C_1| + |A_1B_1| + |A_1C_4|], \\ [|A_2D_3|, |A_2D_3| + |A_2D_2|, |A_2D_3| + |A_2D_2| + |A_2B_3|, |A_2D_3| + |A_2D_2| + |A_2B_3| + |A_2C_2|], \\ [|A_3D_3|, |A_3D_3| + |A_3C_1|, |A_3D_3| + |A_3C_1| + |A_3D_2|, |A_3D_3|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

$k = 3$

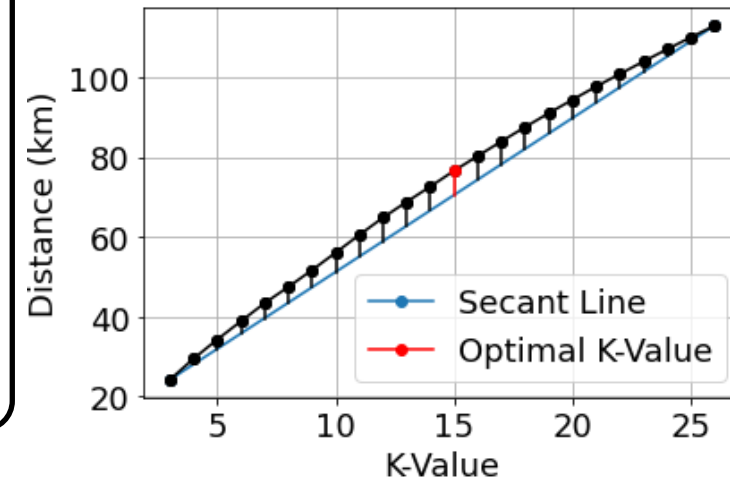
calculate average

$k = 4$

Example with feature type A:



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



Time Complexity

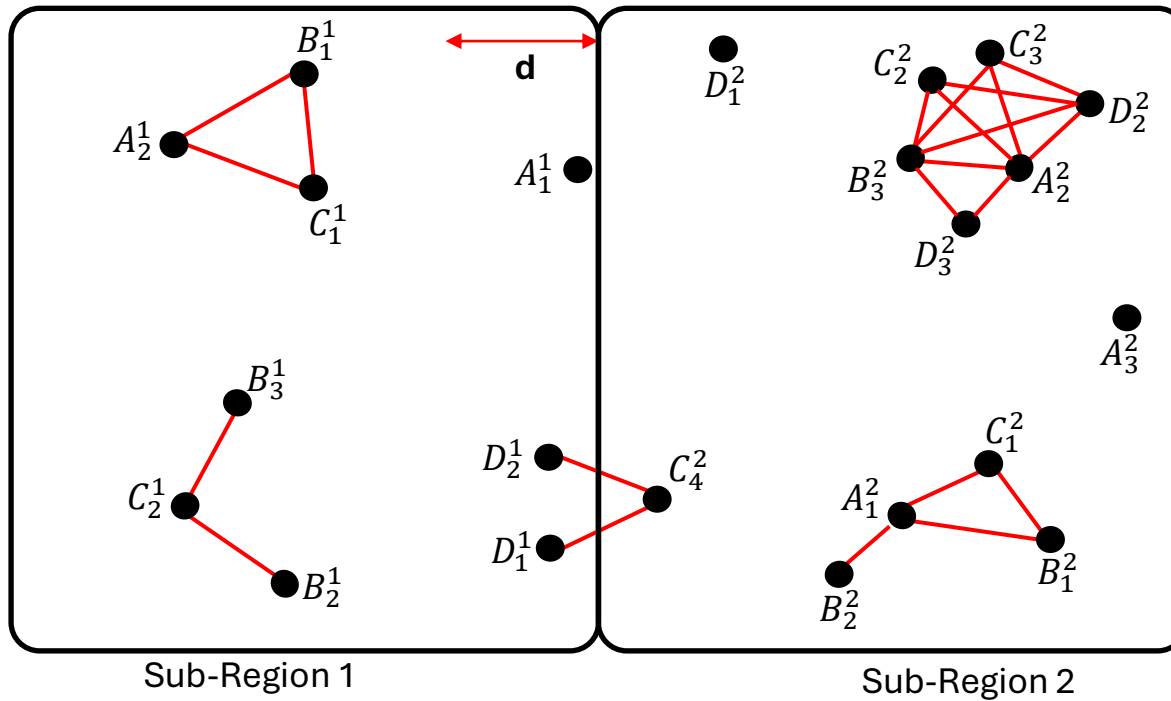
For each feature type F

$$O(N \times \sqrt{N} \log(N))$$

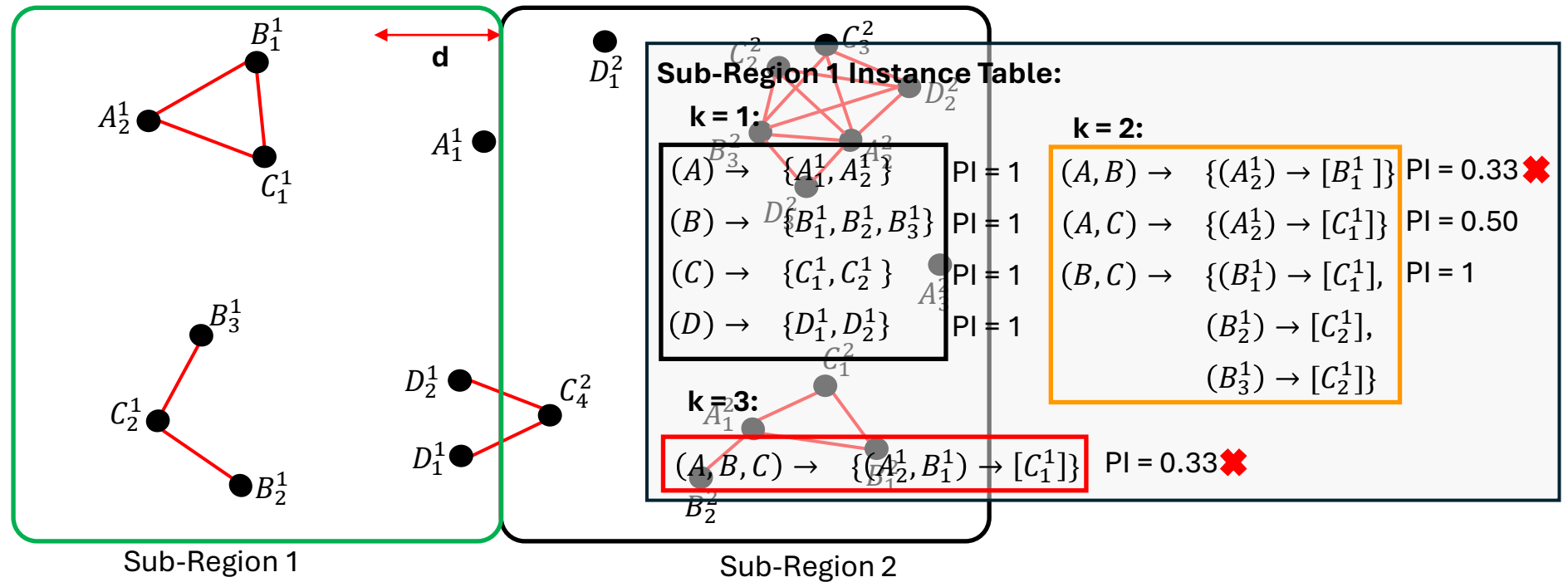
where

- N : number of instances of F

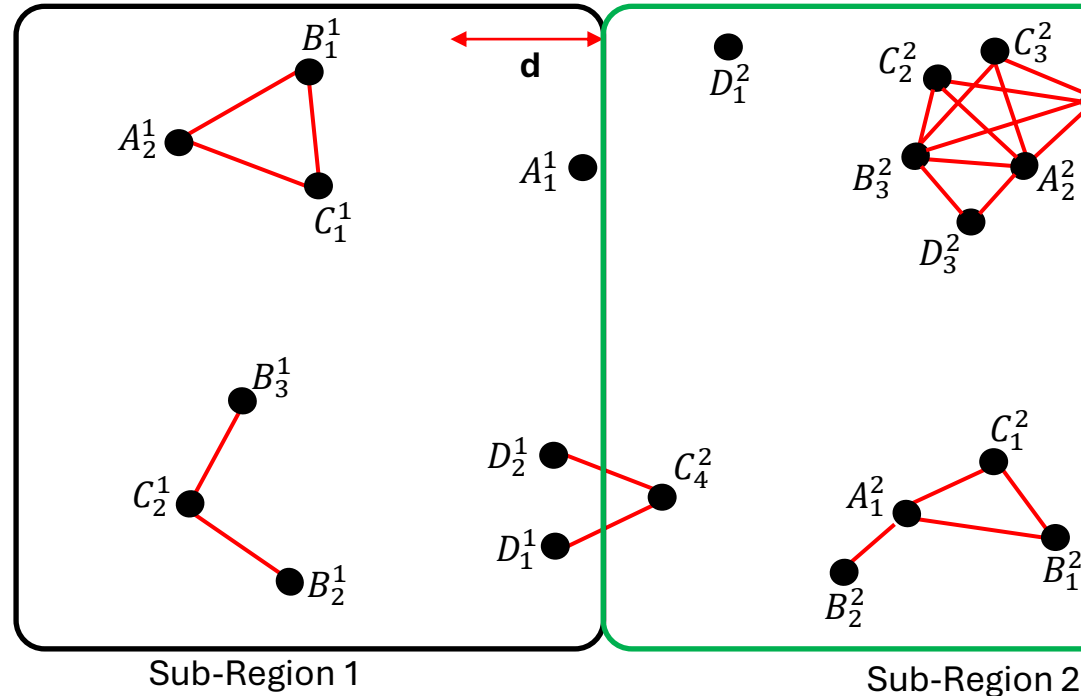
Our Map-Based Approach



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 ✖

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

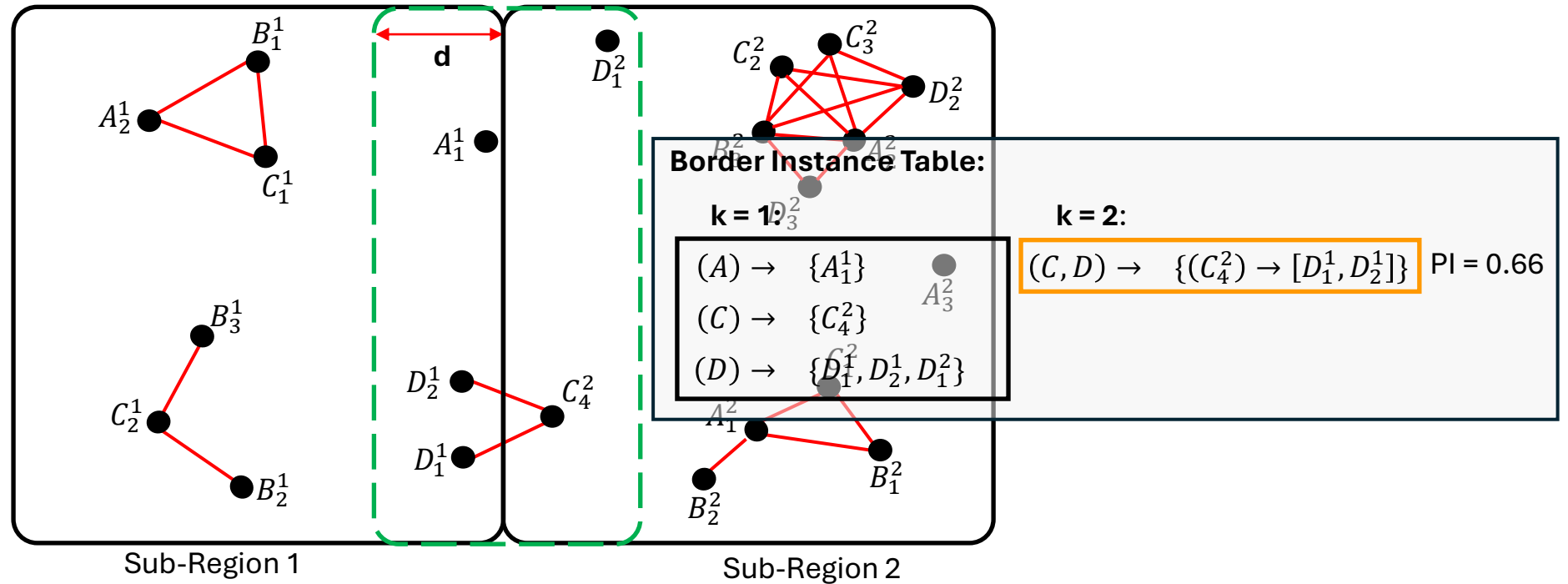
PI = 0.33 ✖

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

Sub-Region 1 Instance Table:

k = 2:

$(A, B) \rightarrow \{(A_1^1) \rightarrow [B_1^1]\}$
 $(A, C) \rightarrow \{(A_1^1) \rightarrow [C_1^1]\}$
 $(B, C) \rightarrow \{(B_1^1) \rightarrow [C_1^1],$
 $(B_2^1) \rightarrow [C_2^1],$
 $(B_3^1) \rightarrow [C_2^1]\}$

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:

$(A, B) \rightarrow \{(A_1^2) \rightarrow [B_1^2, B_2^2],$
 $(A_2^2) \rightarrow [B_3^2]\}$
 $(A, C) \rightarrow \{(A_1^2) \rightarrow [C_1^2],$
 $(A_2^2) \rightarrow [C_2^2, C_3^2]\}$
 $(A, D) \rightarrow \{(A_2^2) \rightarrow [D_2^2, D_3^2]\}$
 $(B, C) \rightarrow \{(B_1^2) \rightarrow [C_1^2],$
 $(B_2^2) \rightarrow [C_2^2, C_3^2]\}$
 $(B, D) \rightarrow \{(B_3^2) \rightarrow [D_2^2, D_3^2]\}$
 $(C, D) \rightarrow \{(C_2^2) \rightarrow [D_2^2],$
 $(C_3^2) \rightarrow [D_2^2]\}$

Regional Instance Table:

k = 2:

$(A, B) \rightarrow \{(A_1^1) \rightarrow [B_1^1], (A_1^2) \rightarrow [B_1^2, B_2^2],$ PI = 0.60
 $(A_2^2) \rightarrow [B_3^2]\}$
 $(A, C) \rightarrow \{(A_1^1) \rightarrow [C_1^1], (A_1^2) \rightarrow [C_1^2], (A_2^2) \rightarrow [C_2^2, C_3^2]\}$ PI = 0.60
 $(B, C) \rightarrow \{(B_1^1) \rightarrow [C_1^1], (B_2^1) \rightarrow [C_2^1],$ PI = 0.83
 $(B_3^1) \rightarrow [C_2^1], (B_1^2) \rightarrow [C_1^2],$
 $(B_2^2) \rightarrow [C_2^2, C_3^2]\}$
 $(C, D) \rightarrow \{(C_2^2) \rightarrow [D_2^2], (C_3^2) \rightarrow [D_2^2], (C_4^2) \rightarrow [D_1^1, D_2^1]\}$ PI = 0.50

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

Time Complexity

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

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

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

Making C a prevalent pattern for the entire region R .

Lemma 2

Let R be a region with n subregions s_1, s_2, \dots, s_n , and m border regions b_1, b_2, \dots, 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 $PR(C, f) < \theta \ \forall s_1, s_2, \dots, s_n$ and $PR < \theta \ \forall b_1, b_2, \dots, b_m$. Then, $PI(C) < \theta$ for R .

Proof:

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

Denote $I_f^b = \{I_f^{b_1}, \dots, I_f^{b_m}\}$ as a set of the instances of feature f participating in C in border regions b_1, b_2, \dots, 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

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

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

Lemma 2

Let R be a region with n subregions s_1, s_2, \dots, s_n , and m border regions b_1, b_2, \dots, 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 $PR(C, f) < \theta \forall s_1, s_2, \dots, s_n$ and $PR < \theta \forall b_1, b_2, \dots, b_m$. Then, $PI(C) < \theta$ for R .

Proof (cont.):

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

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

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

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

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

Evaluation

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

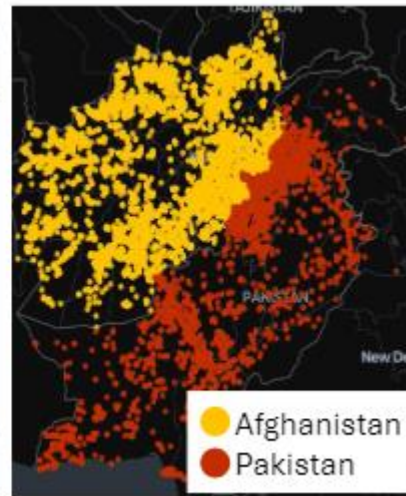
Data Set

Real World

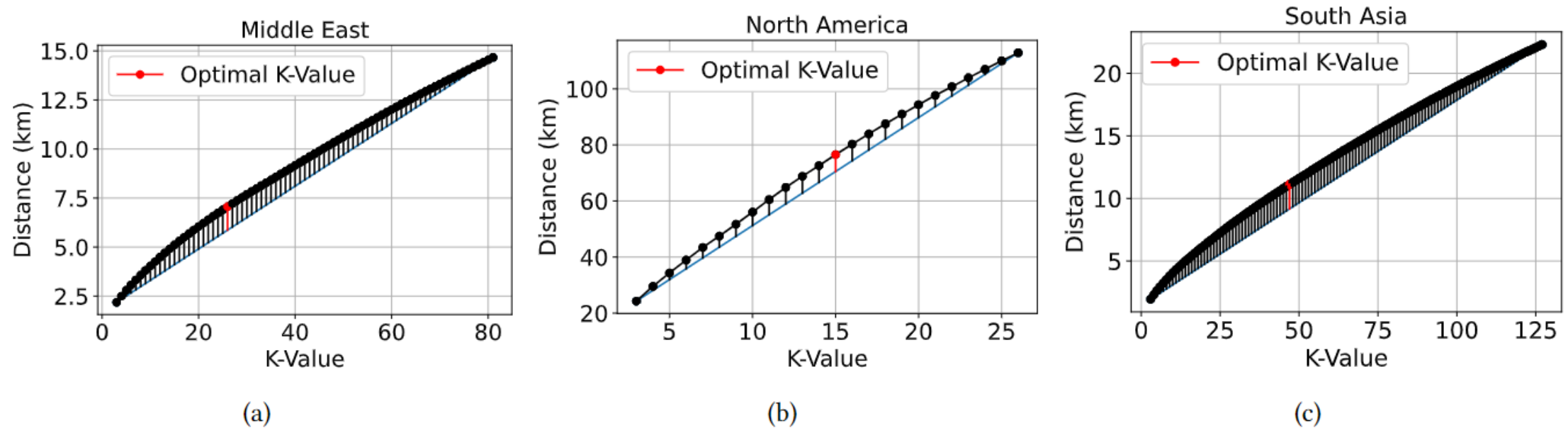
- Global Terrorism Database
- Year: 1970-2020
- Instances: 215k
- Number of Attack Types: 8

Synthetic

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



Results: Real-World Data Set



Observation:

- Different spatial neighborhood relationship constraint for each region

Results: Real-World Data

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	(4, 7), (1, 2, 7), (0, 2, 3, 7)
Mexico	(3, 5), (0, 1, 6), (0, 2, 6)
North America	(1, 6), (0, 1, 2), (0, 2, 3), (2, 3, 7)
Afghanistan	(3, 6), (1, 2, 3), (2, 3, 6)
Pakistan	(1, 4), (1, 5), (0, 1, 2, 6)
South Asia	(0, 1, 2, 6), (0, 1, 2, 3), (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

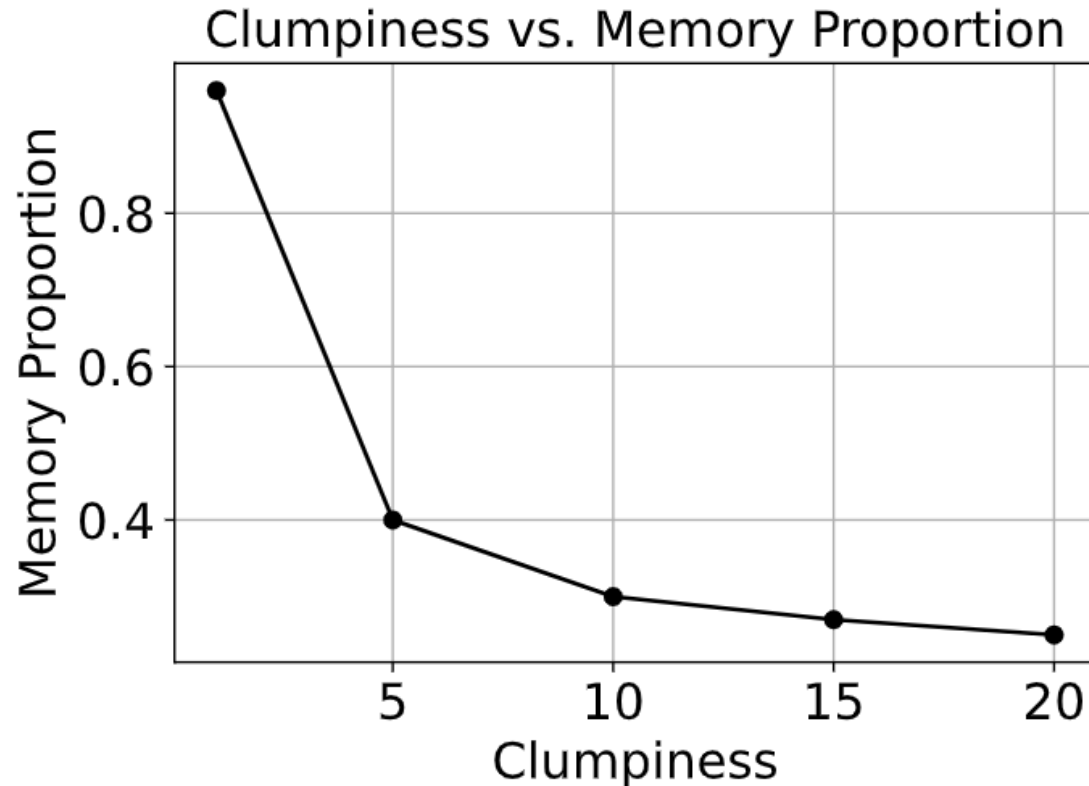
Results: Real World Data Set

Region	Degree	Map-Based Approach	Array-Based Approach	Memory Proportion
Middle East (1)	2	0.0017 GB	0.0033 GB	1/2
	3	0.1178 GB	0.3408 GB	1/3
	4	3.4795 GB	13.0831 GB	1/4
	Total	3.5990 GB	13.4272 GB	1/4
North America (2)	2	0.0006 GB	0.0012 GB	1/2
	3	0.0183 GB	0.0527 GB	1/3
	Total	0.0190 GB	0.0539 GB	1/3
South Asia (3)	2	0.0185 GB	0.0362 GB	1/2
	3	4.0720 GB	11.9411 GB	1/3
	4	438.4528 GB	1662.6944 GB	1/4
	Total	442.5433 GB	1674.6717 GB	1/4

Observation:

- The higher the degree, the smaller the memory proportion
- Our approach uses approximately one-third of the amount of memory of pre-existing array-based approaches

Results: Synthetic Data Set



$$\text{Memory Proportion} = \frac{\text{Our Approach}}{\text{Baseline Approach}}$$

Observation:

- The clumpier the data, the smaller the memory proportion
- Our approach uses one-third of the amount of memory of pre-existing array-based approaches

Thank you

Appendix

Algorithm 1 Regional Colocation Miner

Input: Sub-regions S

Input: A set of spatial features for each sub-region F

Input: Instances of each spatial feature $I[F]$

Input: Minimum prevalence threshold θ

Output: All prevalent sub-regional (s) and regional (R) colocation patterns P^s and P^R respectively

```
1:  $d \leftarrow \text{DYNAMIC\_DISTANCE\_ESTIMATE}(F)$ 
2: for  $s$  in  $S$  do
3:   Initialize  $k \leftarrow 1$ 
4:   Initialize  $C_k^s \leftarrow F, P_k^s \leftarrow F$ 
5:   Initialize  $C_{k+1}^s \leftarrow \text{APRIORI\_GEN}(P_k^s, k + 1)$ 
6:   Initialize instance tables  $I_k^s$  with  $I[F]$  in region  $s$ 
7:   Initialize  $I_{k+1}^s \leftarrow \emptyset, P_{k+1}^s \leftarrow \emptyset$ 
8:    $N^s \leftarrow \text{CALCULATE\_STAR\_NEIGHBORS}(s)$ 
9:   while  $|C_{k+1}^s| > 0$  do
10:     $(P_{k+1}^s, I_{k+1}^s) \leftarrow \text{CALC\_PATTERNS}(k + 1, I_k^s)$ 
11:     $P^s \leftarrow P^s \cup P_{k+1}^s$ 
12:     $k = k + 1$ 
13:     $C_{k+1}^s \leftarrow \text{APRIORI\_GEN}(P_k^s, k + 1)$ 
14:   end while
15: end for
```

```
16:  $k \leftarrow 2$ 
17:  $I_k^b \leftarrow \text{CALCULATE\_BORDER\_PATTERNS}(k)$ 
18:  $I_k^R \leftarrow \text{COMBINE\_INSTANCE\_TABLES}(I_k^s, I_k^b)$ 
19:  $P_k^R \leftarrow \text{PI}(I_k^R)$ 
20:  $C_{k+1}^R \leftarrow \text{APRIORI\_GEN}(P_k^R, k + 1)$ 
21: while  $|C_{k+1}^R| > 0$  do
22:    $(P_{k+1}^R, I_{k+1}^R) \leftarrow \text{CALC\_PATTERNS}(k + 1, I_k^R)$ 
23:    $P^R \leftarrow P^R \cup P_{k+1}^R$ 
24:    $k = k + 1$ 
25:    $C_{k+1}^R \leftarrow \text{APRIORI\_GEN}(P_k^R, k + 1)$ 
26: end while
27: return  $P^s, P^R$ 
```

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

```
1: Initialize  $D \leftarrow \emptyset$ ,  $K_{max} \leftarrow \sqrt{|I[F]|} + 1$ ,  $A \leftarrow \emptyset$ 
2: Initialize memoization table  $T \leftarrow \emptyset$  of size  $|I[F]| \times K_{max}$ 
3: for  $f$  in  $F$  do
4:    $S_f \leftarrow \text{EXTRACT\_BY\_FEATURE\_TYPE}(f)$ 
5:    $I[F]_{\text{Excludingfeature}} \leftarrow I[F] - S_f$ 
6:    $r \leftarrow \text{RTREE}()$ 
7:    $\text{ADD\_POINTS\_TO\_RTREE}(r, I[F]_{\text{Excludingfeature}})$ 
8:   for  $p$  in  $S_f$  do
9:      $x \leftarrow p[0]$ ,  $y \leftarrow p[1]$ 
10:     $N = r.\text{nearest}((x, y), K_{max})$ 
11:     $D.\text{append}(\text{SORT\_NEIGHBORS\_DISTANCES}(N))$ 
12:   end for
13: end for
```

```
14:  $T[:, 2] = \text{ROWWISE\_SUM}(D[:, : 3])$ 
15: for  $i$  in  $\text{range}(|I[F]|)$  do
16:   for  $k$  in  $\text{range}(3, K_{max})$  do
17:      $T[i, k] = T[i, k - 1] + D[i][k]$ 
18:   end for
19: end for
20:  $C = \text{COLUMNWISE\_SUM}(T)$ 
21: for  $k$  in  $\text{range}(2, K_{max})$  do
22:    $A.\text{append}(C[k] / (|I[F]| \times (k + 1)))$ 
23: end for
24:  $d \leftarrow \text{KNEE\_METHOD}(A)$ 
25: return  $d$ 
```

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

```
1: Initialize  $P_k \leftarrow \emptyset, I_k \leftarrow \emptyset, H \leftarrow \emptyset$ 
2: for  $c$  in  $C_k$  do
3:    $B_{pattern} \leftarrow c[0 : k-1], L_f \leftarrow c[k-1]$ 
4:   Initialize  $I_k[c] \leftarrow \emptyset, H[c] \leftarrow \emptyset$ 
5:    $H[c] \leftarrow \{f : \emptyset \text{ for } f \text{ in } c\}$ 
6:    $I_{base} \leftarrow I_{k-1}[B_{pattern}]$ 
7:   for  $key$  in  $I_{base}$  do
8:      $N \leftarrow \emptyset$ 
9:     for  $id$  in  $key$  do
10:      if not  $N$  then
11:         $N \leftarrow \text{NEIGHBOR}(S[id], F_{info}[L_f])$ 
12:      else
13:         $N \leftarrow N \cap \text{NEIGHBOR}(S[id], F_{info}[L_f])$ 
14:      end if
15:    end for
```

```
16:    for  $i$  in  $I_{base}[key]$  do
17:       $n = N \cap \text{NEIGHBOR}(S[i], F_{info}[L_f])$ 
18:      if  $n$  then
19:         $key_{new} = key.append(i)$ 
20:         $I_k[c][key_{new}] \leftarrow n$ 
21:        for  $j \in key_{new}$  do
22:           $f \leftarrow \text{GET\_FEATURE\_ID}(j)$ 
23:           $H[c][f].add(j)$ 
24:        end for
25:         $H[c][L_f].update(n)$ 
26:      end if
27:    end for
28:  end for
29:   $PR \leftarrow \emptyset$ 
30:  for  $f$  in  $c$  do
31:     $PR.append(|H[c][f]|/F_{info}[f].count)$ 
32:  end for
33:   $PI = \min(PR)$ 
34:   $P_k.append(c)$  if  $PI \geq \theta$ 
35: end for
36: return  $P_k, I_k$ 
```
