Lecture 6: Association Rules and Colocations

Spatial Data Mining

Instructor: Yiqun Xie

Attribution: Slides modified based on lecture notes from Dr. Xun Zhou (UI) and Dr. Shashi Shekhar (UMN)

Outline

- Association Rule Mining
  - Apriori algorithm
- Spatial cross-k function
- Co-location Pattern Mining

What is Association?

- Conceptual example
  - If we observe A, we also expect to observe B
  - We say occurrences of A and B are associated

A foundational pattern in data mining

Oracle example,
Some others…

- Who may be interested in this?

- Real-world examples
  - Customers who purchase A also purchase B
  - Netflix users who watch A also watch B
  - Patients who have symptom A are diagnosed with disease B
  - …

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Market-Basket transactions

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example of Association Rules

{Diaper} $\rightarrow$ {Beer},
{Milk, Bread} $\rightarrow$ {Eggs,Coke},
{Beer, Bread} $\rightarrow$ {Milk},

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Definition: Association Rule

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}

- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *minsup* and *minconf* thresholds

  ⇒ Computationally prohibitive!

Mining Association Rules

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Example of Rules:

{Milk,Diaper} $\rightarrow$ {Beer} (s=0.4, c=0.67)
{Milk,Beer} $\rightarrow$ {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} $\rightarrow$ {Milk} (s=0.4, c=0.67)
{Beer} $\rightarrow$ {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} $\rightarrow$ {Milk,Beer} (s=0.4, c=0.5)
{Milk} $\rightarrow$ {Diaper,Beer} (s=0.4, c=0.5)

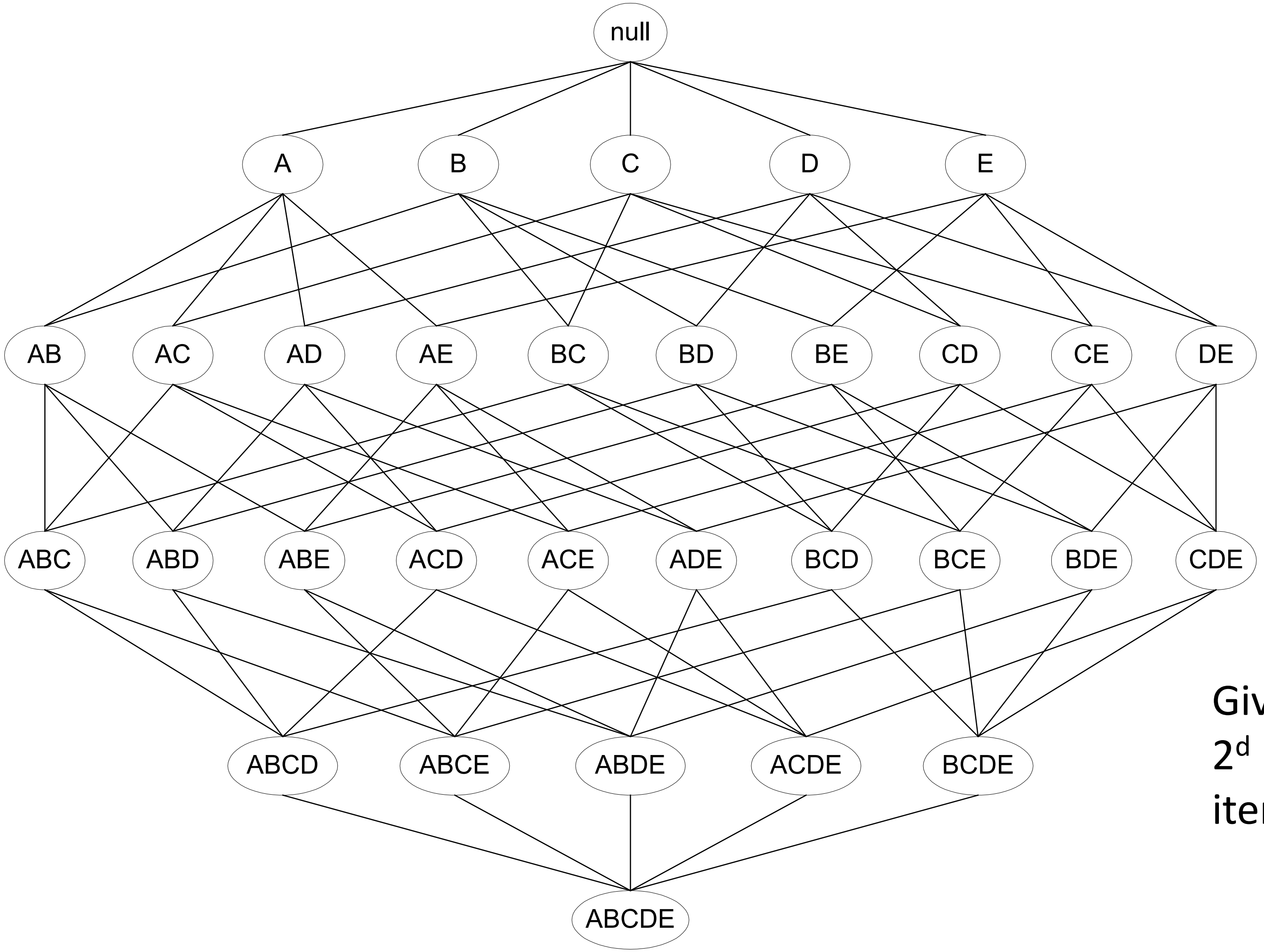
Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

Mining Association Rules

- Two-step approach:
  1. Frequent Itemset Generation
     – Generate all itemsets whose support $\geq$ minsup
  2. Rule Generation
     – Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

Frequent Itemset Generation



Given d items, there are 2^d possible candidate itemsets

Frequent Itemset Generation

- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

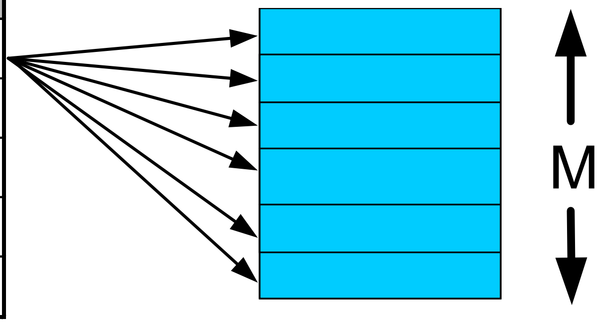
Transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

N

w

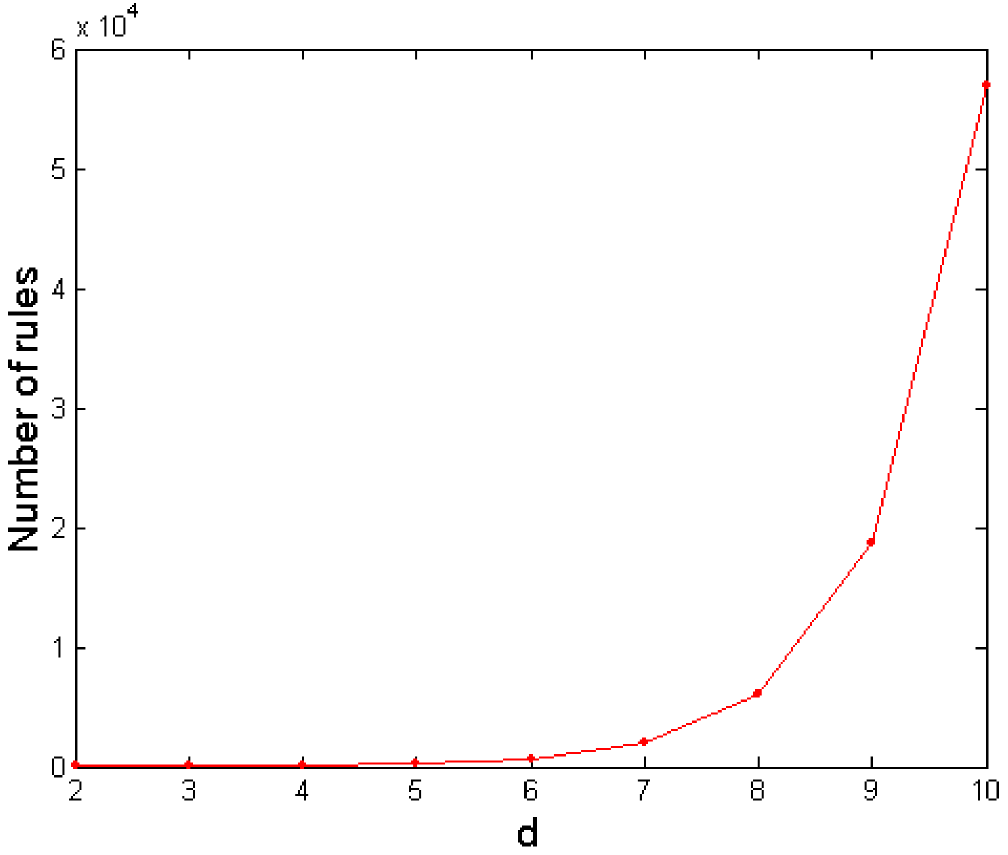
List of Candidates



  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



If d=6, R = 602 rules

If d=10, R = 57,002 rules

If d=100, R = 5×10^{47} rules

Even if one can calculate 1 million of these per second, it will take 1.6×10^{34} years to complete

Reducing Number of Candidates

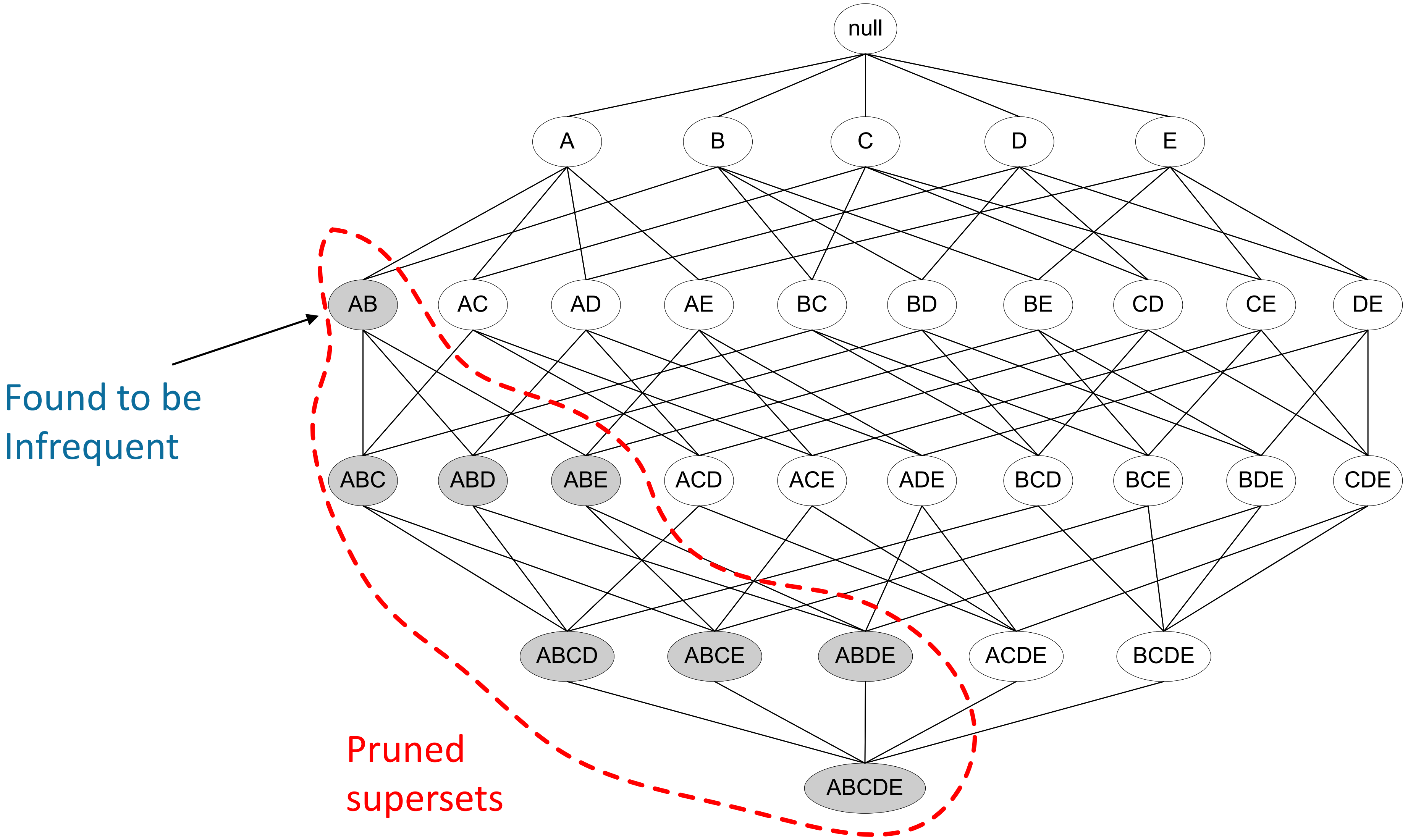
- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent

- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

Illustrating Apriori Principle



Illustrating Apriori Principle

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

Minimum Support Count = 3

If every subset is considered,
41
With support-based pruning,
13

Apriori Algorithm

- Method:
  - Let k=1
  - Generate frequent itemsets of length 1
  - Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Apriori Algorithm

- Suppose “apple” does not have enough support

Association and Causality

- If two events A and B have high association
  - i.e., when we observe A we also observe B
  - Can we say A causes B?
  - Can we say A causes B, or, B causes A?

Real Examples

- Case 1: Energy drink and sleeping in class

- Case 2: Drinking and health

nature communications

Explore content ⌄ Journal information ⌄ Publish with us ⌄

nature › nature communications › articles › article

Article | Open Access | Published: 07 January 2021

Genome-wide analyses of behavioural traits are subject to bias by misreports and longitudinal changes

Angli Xue, Longda Jiang, Zhihong Zhu, Naomi R. Wray, Peter M. Visscher, Jian Zeng & Jian Yang

Nature Communications **12**, Article number: 6450 (2021) | Cite this article

2799 Accesses | **86** Altmetric | Metrics

"There were 15,889 individuals (8.3%) choosing illness or doctor's advice as the primary reason for reducing drinking, and their mean disease count was nearly twice that of all other current drinkers (Table 2)."

A website with some funny and "a bit extreme" examples:
http://www.tylervigen.com/spurious-correlations

Association and Causality

- If two events A and B have high association
  - i.e., when we observe A we also observe B
  - Can we say A causes B?
  - Can we say A causes B, or, B causes A?
- Association does not mean causality!

Colocation Pattern and Examples

- Colocation: a set of *spatial features* that frequently occur in *together*
- Example:

Spatial association rule mining

  - Ecology: symbiotic relationship in animals or plants
  - Public health: environmental factors and cancers
  - Public safety: crime generators and crime events
  - Business: {Kum&Go, Casey's}, {Walmart, Subway}

Nile Crocodiles and Egyptian Plover
http://www.alamy.com/

Gobbies and Pistol Shrimps
fragbox.ca, blog.wakatobi.com

Bar closing events and crimes
http://www.startribune.com/

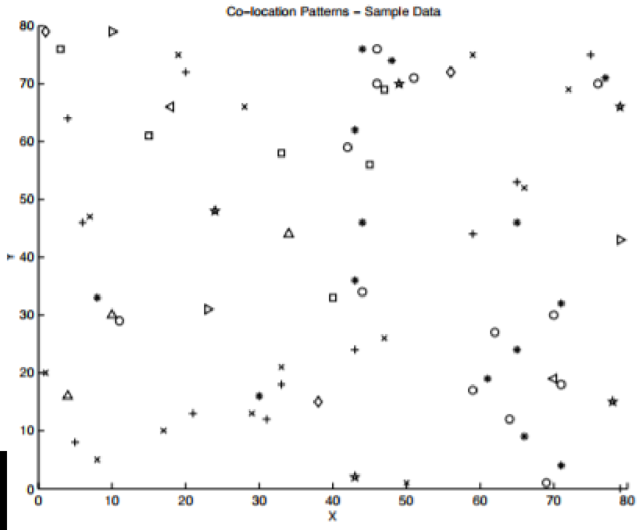
Basic Concepts

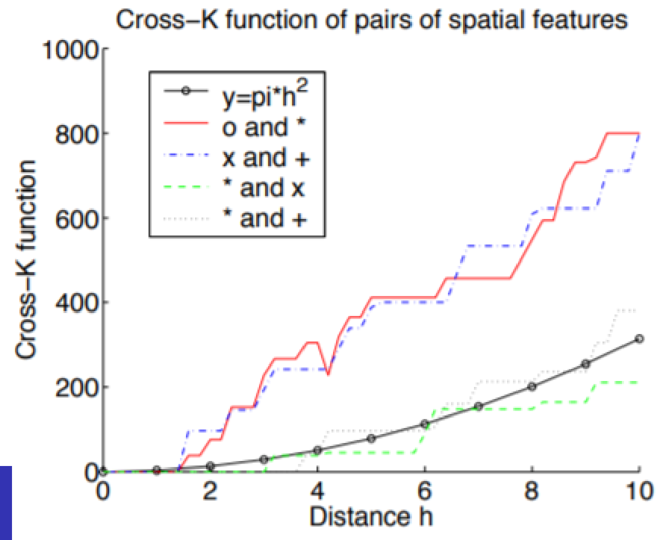
- Spatial event type
  - Example: Bar closing, drunk driving
- Spatial event instance
  - Belong to an event type, associated with a location
  - Example: one specific drunk driving event
- Colocation pattern c:
  - A subset of spatial event types: (bar closing, drunk driving)
  - Instances of these event types *frequently occur together*

Cross-K Function

- An extension of Ripley's K function

- Test if of two types of events tend to cluster together
  - H_0: event types i and j are independent
  - H_1: event types i and j tend to cluster together
  - Test statistic:
    - $K_{ij}(d) = \lambda_j{}^{-1} E(\#\ of\ points\ of\ type\ j\ within\ d\ of\ a\ point\ i)$
    - Under H_0, $K_{ij}(d) = \pi d^2$





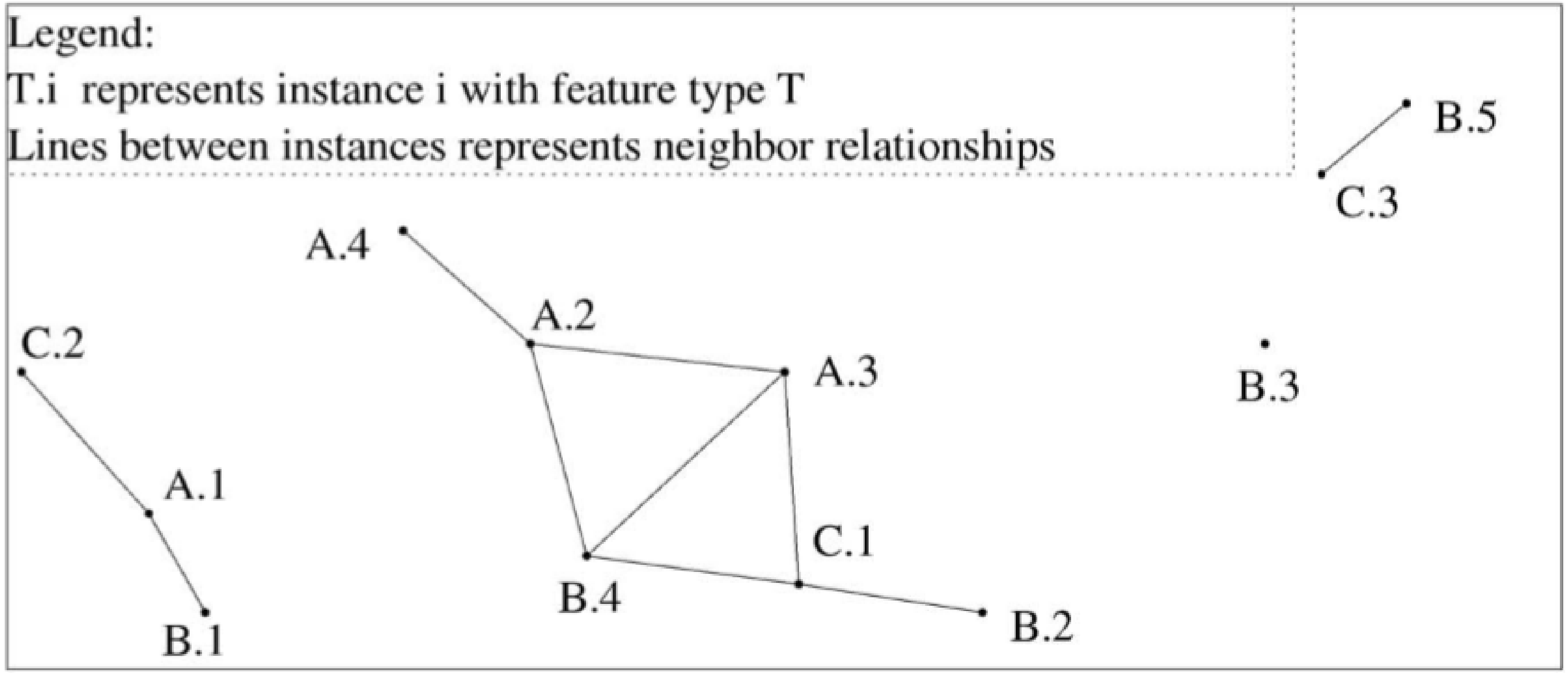
Co-location and Cross-K function

- Cross-k function
  - Each pair of events is tested separately
  - Potential duplicated counting
    - May make it less reflective of spatial distribution
- Co-location pattern mining
  - More general relationships among events (similar to association)
  - More efficient algorithms
  - Distribution matters more

Basic Concepts

- Neighbor relationship R
  - Connect two event instances if they are neighbors → Neighbor Graph
  - Determined by a distance threshold or adjacency
- R-proximity neighborhood
  - A clique of multiple event instances
  - Any pair of instances are neighbors, according to R
- Row instance of a colocation pattern c
  - An R-proximity neighborhood
  - Each event type in c appears only once → Examples in the next slide
- Table instance of a colocation pattern c
  - Collection of all row instances of c

Basic Concept Example



Question: Table instance of (A, B, C)?

Spatial event types
A, B, C

Neighbor relationship (solid line)
(A.1, B.1), (A.1, C.2) …

Spatial event instances
A.1, A.2, A.3, … ...

Candidate Colocation
(A, B), (B, C) …

Table Instance 1

(A,B)
(A.1, B.1)
(A.2, B.4)
(A.3, B.4)

Table Instance 2

(A,B,C)
(A.3, B.4, C.1)

How about (A.2, B.4, C.1)?

Interest Measure (Score Function)

- Participation ratio pr
  - Given colocation pattern $c = (f_1, f_2, \dots, f_k)$
  - $pr(candidate\ c, event\ type\ f_i) = \frac{Number\ of\ f_i\ instances\ participating\ in\ c}{Number\ of\ f_i\ instances}$
- Participation index pi
  - $pi(c) = min_i\{pr(c, f_i)\}$
- Example:

T1

A
A.1
A.2
A.3
A.4

T2

B
B.1
B.2
B.3
B.4
B.5

T3

C
C.1
C.2
C.3

T7

(A,B,C)
(A.3, B.4, C.1)

$pr((A,B,C),A) = \frac{1}{4}$

$pr((A,B,C),B) = \frac{1}{5}$

$pr((A,B,C),C) = \frac{1}{3}$

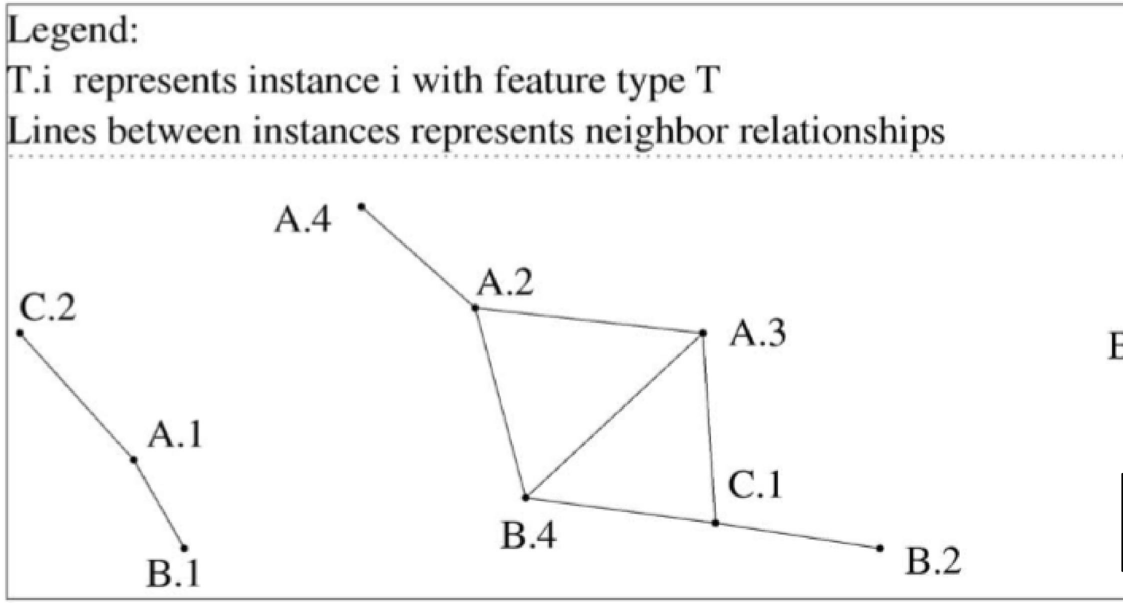
$pi((A,B,C),C) = \frac{1}{5}$

Problem Definition

- Input:
  - A set of spatial event types $(f_1, f_2, \dots, f_k)$
  - A table instance for each event type
  - Spatial neighbor graph
  - A participation index threshold δ
- Find:
  - All colocation patterns c such that $pi(c) \geq \delta$

Problem Example

Input:



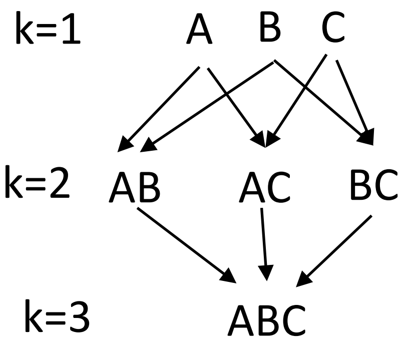
$\delta = 0.5$

Output:

{A,C} with $pi(\{A, C\}) = 0.5$
{B,C} with $pi(\{B, C\}) = 0.6$

Colocation Mining Algorithm: Baseline

- Starting with $k = 1$
- Iterate until no prevalent pattern
  - Generate size k colocation patterns $\{c_k\}$
  - Generate table instance of each c_k
  - Compute each $pi(c_k)$, add to result if prevalent
  - $k = k + 1$



T1	T2	T3
A	B	C
A.1	B.1	C.1
A.2	B.2	C.2
A.3	B.3	C.3
A.4	B.4	
	B.5	

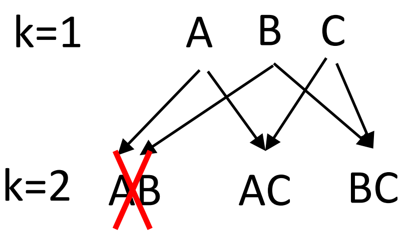
T4	T5	T6
A B	A C	B C
A.1, B.1	A.1, C.2	B.2, C.1
A.2, B.4	A.3, C.1	B.4, C.1
A.3, B.4		B.5, C.3
$pi = 0.4$	$pi = 0.5$	$pi = 0.6$

T7
A B C
A.3, B.4, C.1

$pi = 0.2$

Prevalence-based Pruning

- Lemma (apriori property):
  - If a colocation pattern c_k is not prevalent, then any superset of c_k is also not prevalent
- Example



T1	T2	T3
A	B	C
A.1	B.1	C.1
A.2	B.2	C.2
A.3	B.3	C.3
A.4	B.4	
	B.5	

T4	T5	T6
~~A B~~	A C	B C
~~A.1, B.1~~	A.1, C.2	B.2, C.1
~~A.2, B.4~~	A.3, C.1	B.4, C.1
~~A.3, B.4~~		B.5, C.3

$pi = 0.4$ $pi = 0.5$ $pi = 0.6$

Don't need to check (A,B,C)

Reference

[1] Huang, Yan, Shashi Shekhar, and Hui Xiong. "Discovering colocation patterns from spatial data sets: a general approach." *IEEE Transactions on Knowledge and data engineering* 16.12 (2004): 1472-1485.

Other Patterns

- Outliers/Anomaly
- Cascading (spatio-temporal)
- Teleconnection
- Change detection
- …

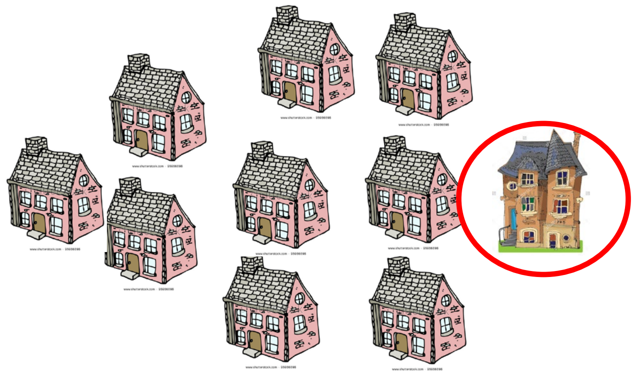
Other Patterns: Optional

- Spatio-Temporal Cascading Patterns
  - Generalization of colocation with time
- Outliers
  - Global vs. Spatial

- The following slides are optional

What is Spatial Outlier?

- Global outlier (anomaly)
  - Data samples different from other samples in population
  - Defined based on global distribution
- Spatial outlier (anomaly)
  - Locations where samples different from their neighbors
  - Defined based on neighborhood context



Global outlier

Spatial outlier

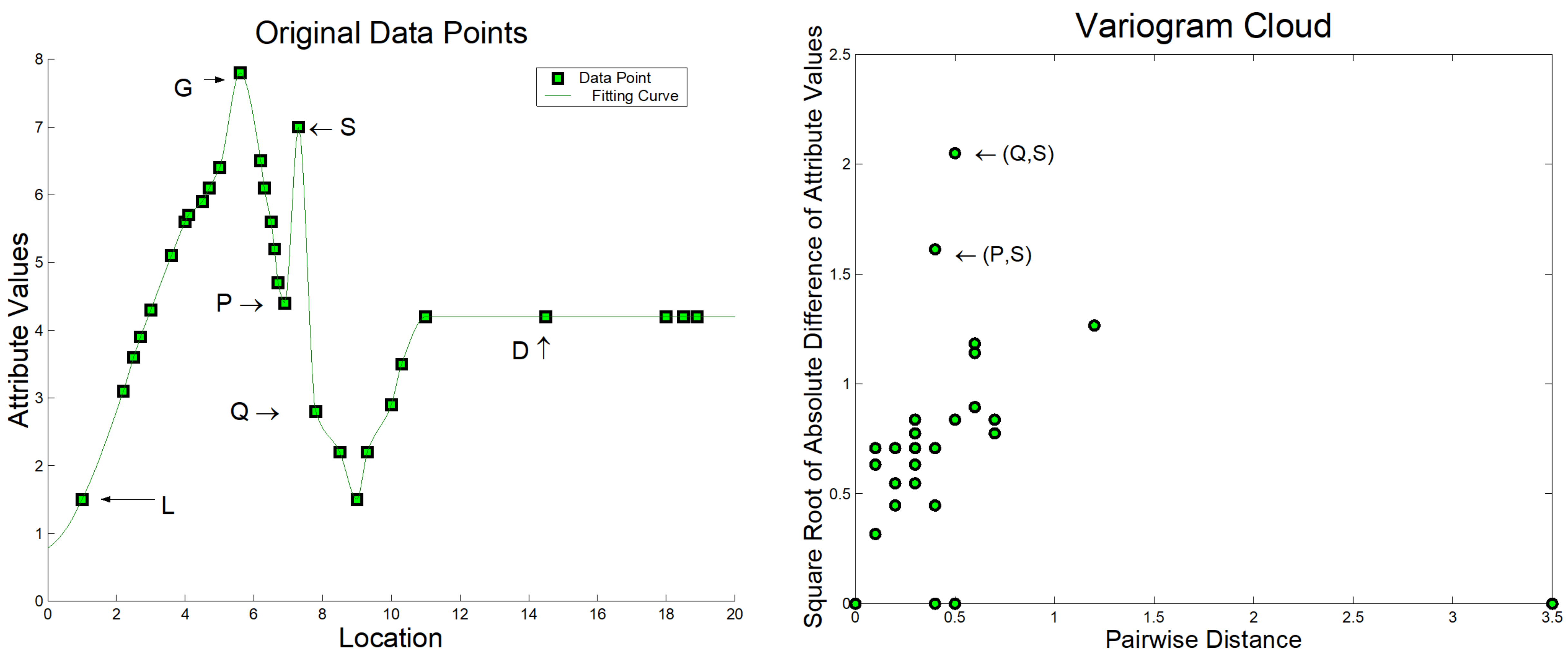
What is Spatial Outlier?

- Spatial attribute
  - Attribute related to object location and footprint
  - Coordinates (latitude, longitude), extent
- Spatial neighborhood relationship
  - Determined based on spatial attribute
  - Distance threshold, touch, network topology
- Non-spatial attributes
  - House age, color, income
- Spatial outlier
  - Object whose non-spatial attributes differ significantly from their neighbors’

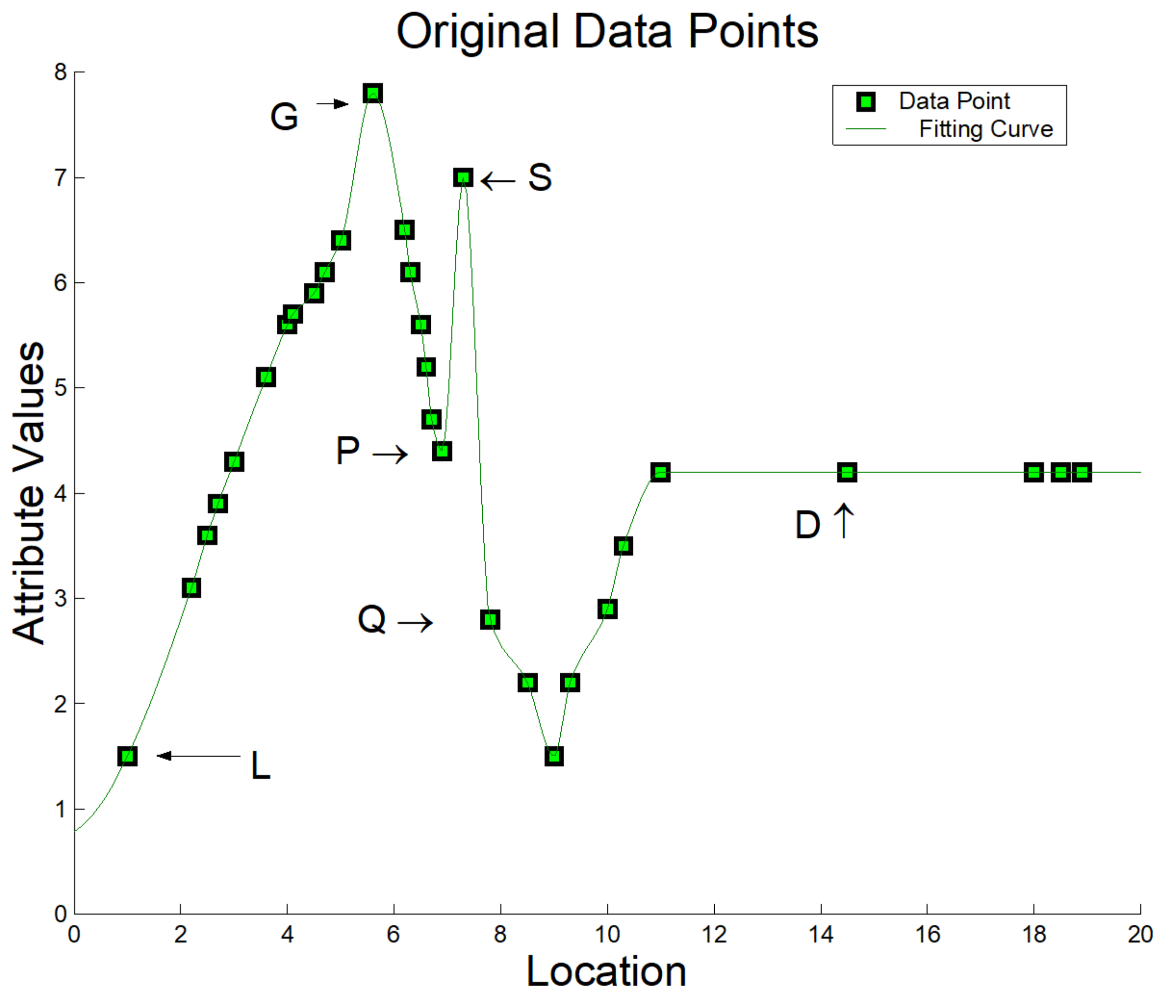
How to Detection Spatial Outliers

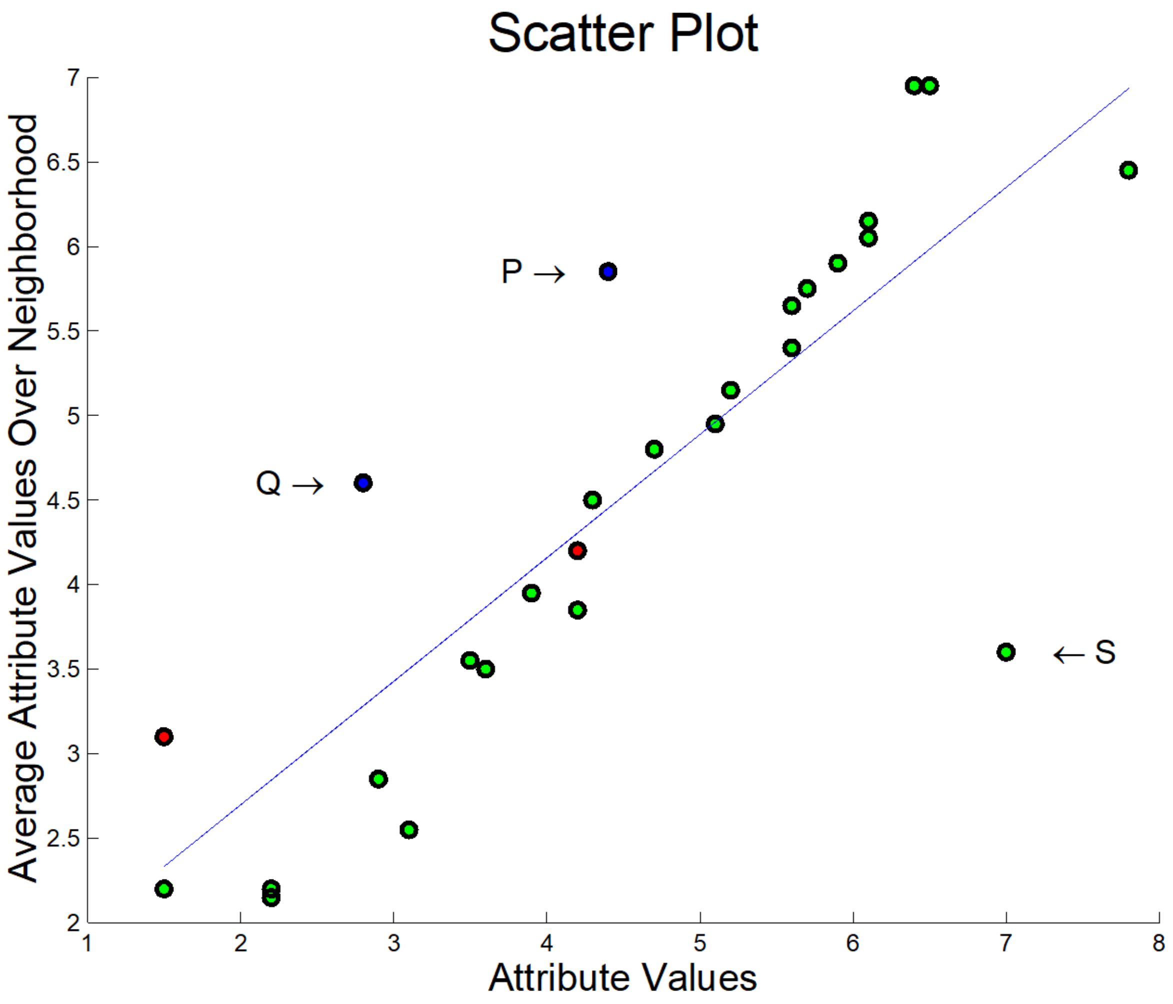
- Visualization approach
  - Variogram cloud
  - Moran scatterplot
- Neighborhood approach
  - Distance-based neighbors
  - Graph-based neighbors

Spatial Outlier Detection: Variogram Cloud



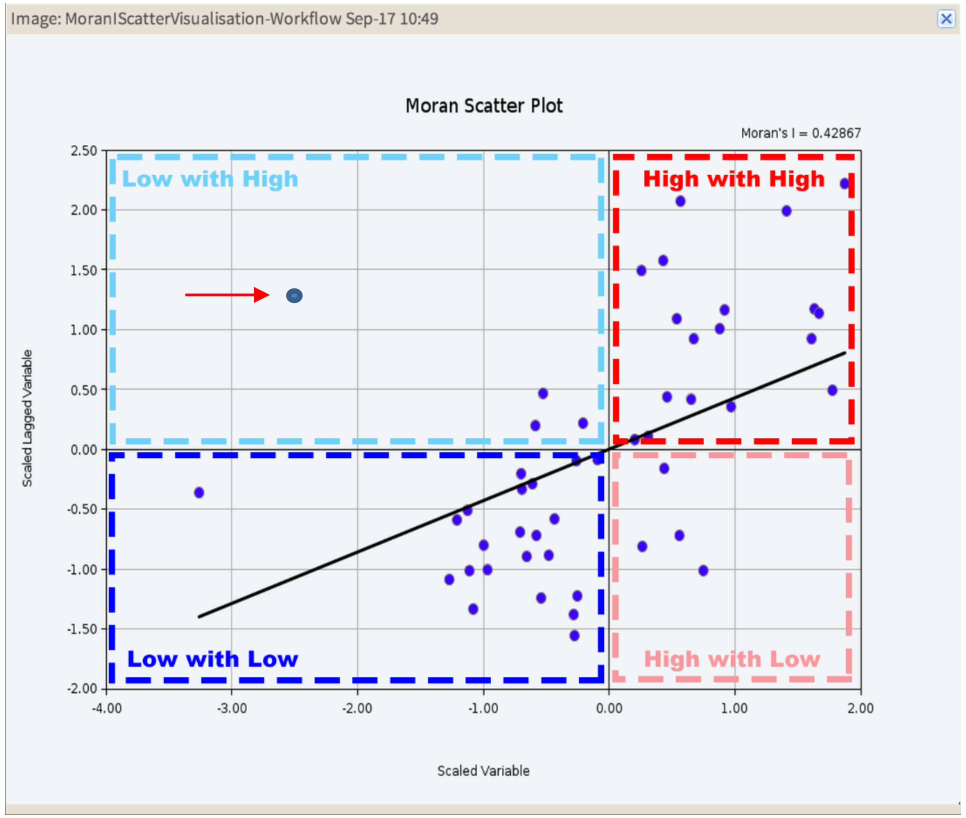
Spatial Outlier Detection: Moran Scatterplot





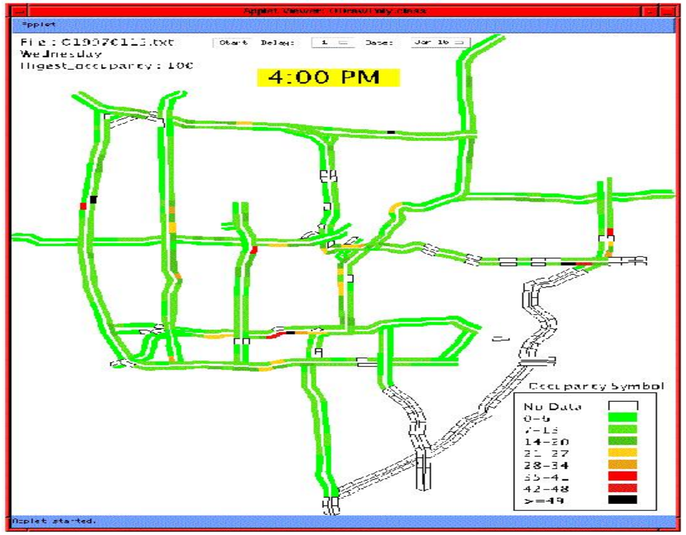
Spatial Outlier Detection: Graphical Methods

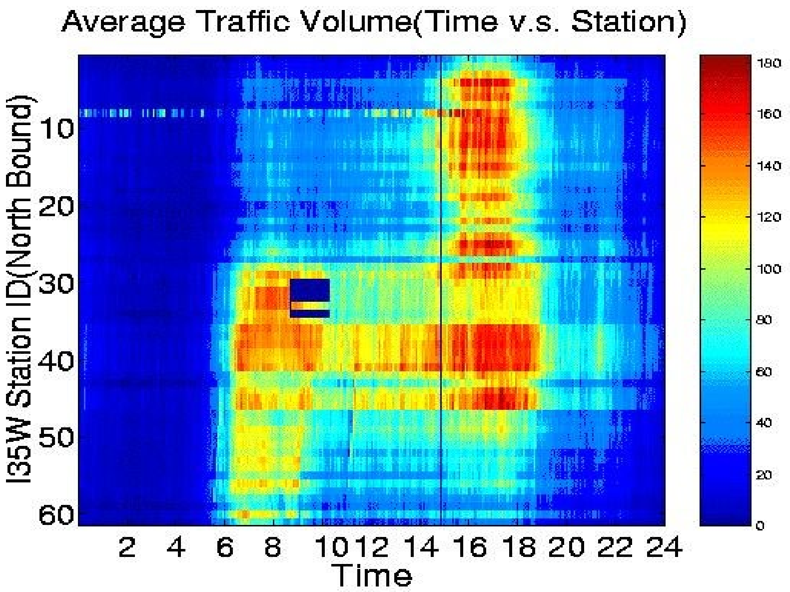
- High Value surrounded by high values: Hotspot
- Low value surrounded by low values: Cold spot
- Moran Scatter Plot
  - X-axis: z-score of a location
  - Y-axis: weighted avg neighborhood z-score
  - Slope of fitted line: Global Moran's I
- Interpretation:
  - Quadrant II and IV: Outliers/transitions



Spatial Outlier Detection: Neighborhood Approach

- $Z_{S(x)} = \frac{S(x) - \mu_S}{\delta_S}$
- where $S(x)$ is difference between one observation and its neighborhood average, μ_S is expectation of $S(x)$, δs is standard deviation of $S(x)$
- Assuming Gaussian distribution
- If $Z_s(x)$ >= 3.0 (top 0.5%) of the entire dataset, then x is a spatial outlier





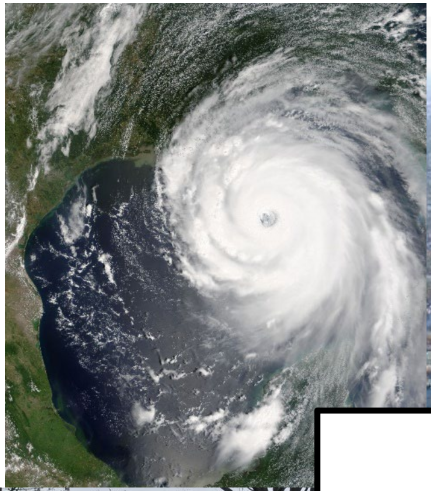
Spatial outliers

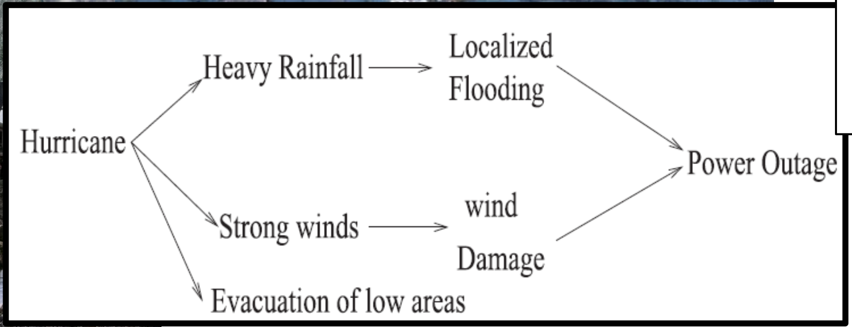
Spatio-Temporal Cascading Patterns

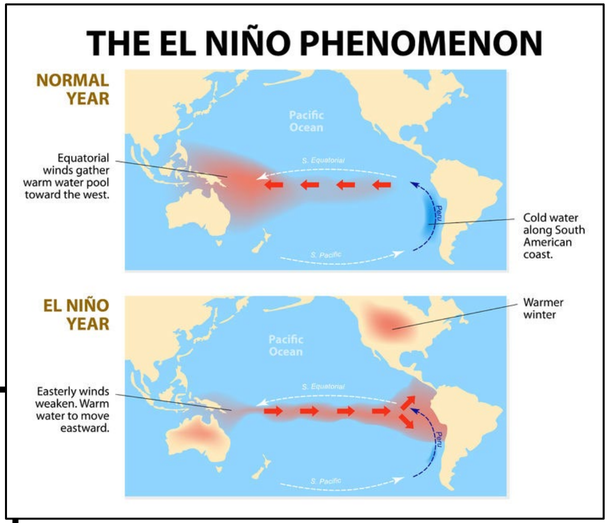
- Generalization of colocation with time

Motivation (Optional starting from this slide)

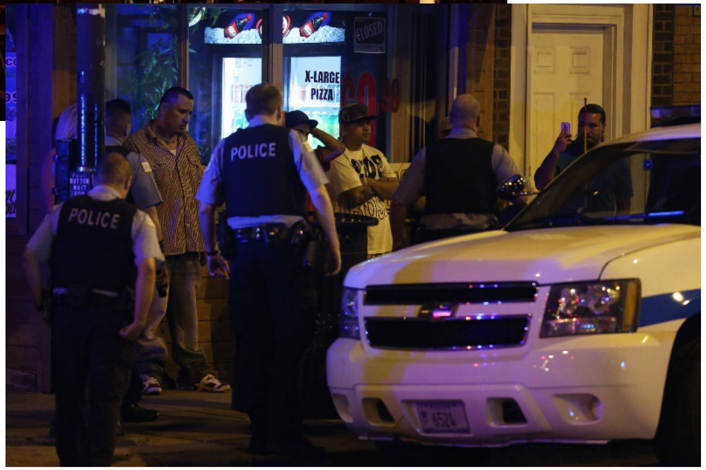
- Cascading spatiotemporal patterns are valuable in many fields:
  - Natural disaster prediction
  - Crime analysis



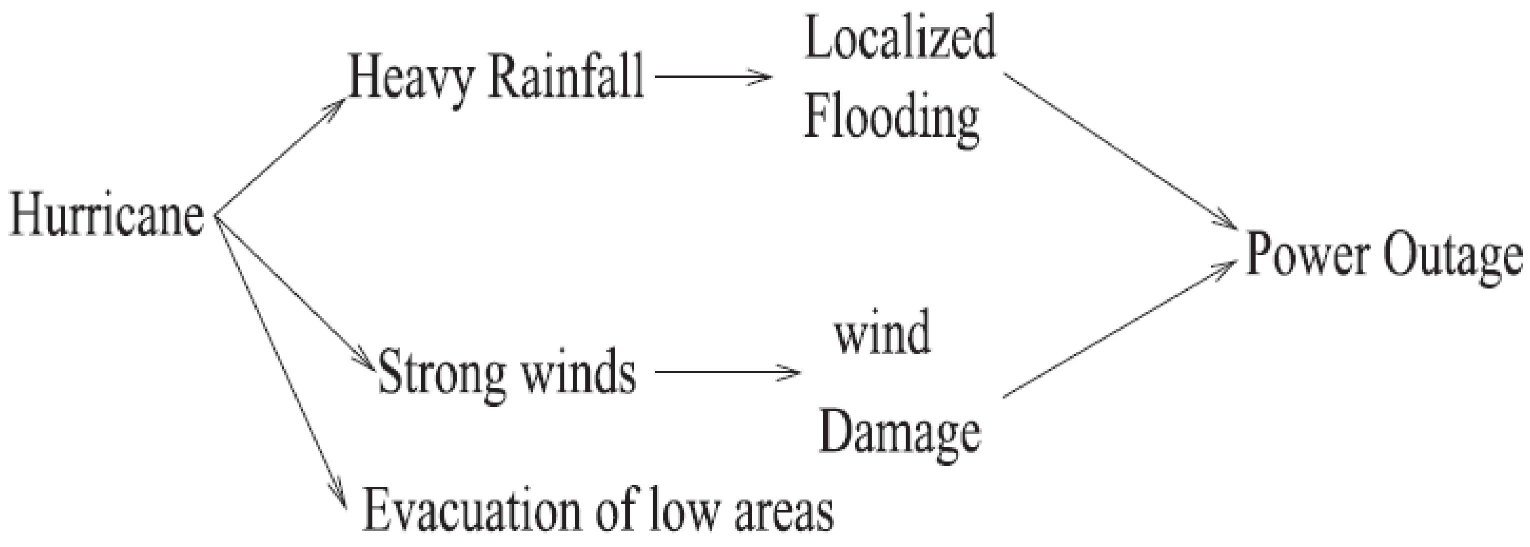


www.cbs5az.com

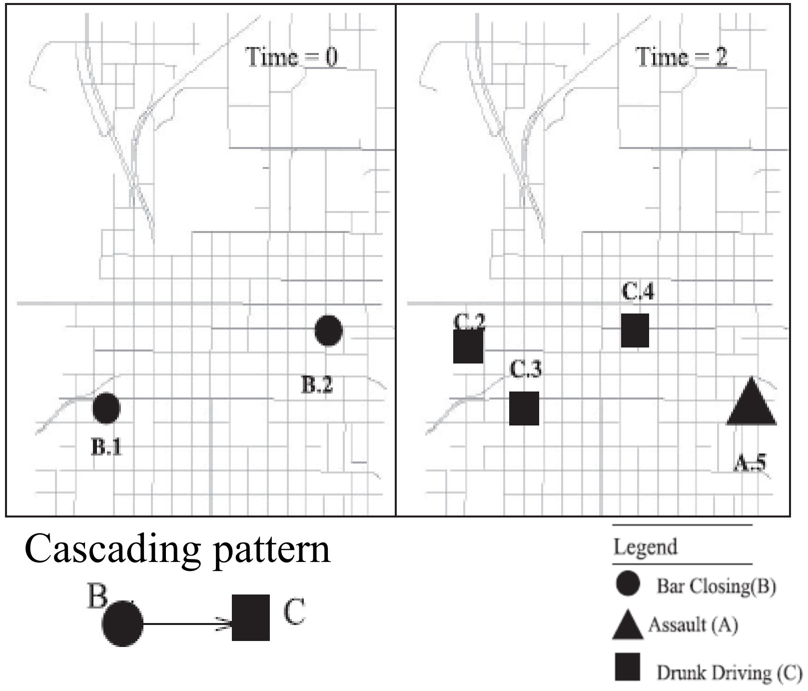
Cascading spatiotemporal pattern (CSTP)

- An acyclic directed graph of features G’=<N’, E’>;
- G’ is a connected graph;



Problem Definition: Differences

- Input
  - A set of geo-located feature instances
  - An interaction distance interval [$d1$, $d2$]
  - An interaction time interval: [$t1$, $t2$]
  - An interest measure threshold r
- Output
  - Cascading spatiotemporal patterns
- Constraint
  - CSTP is acyclic



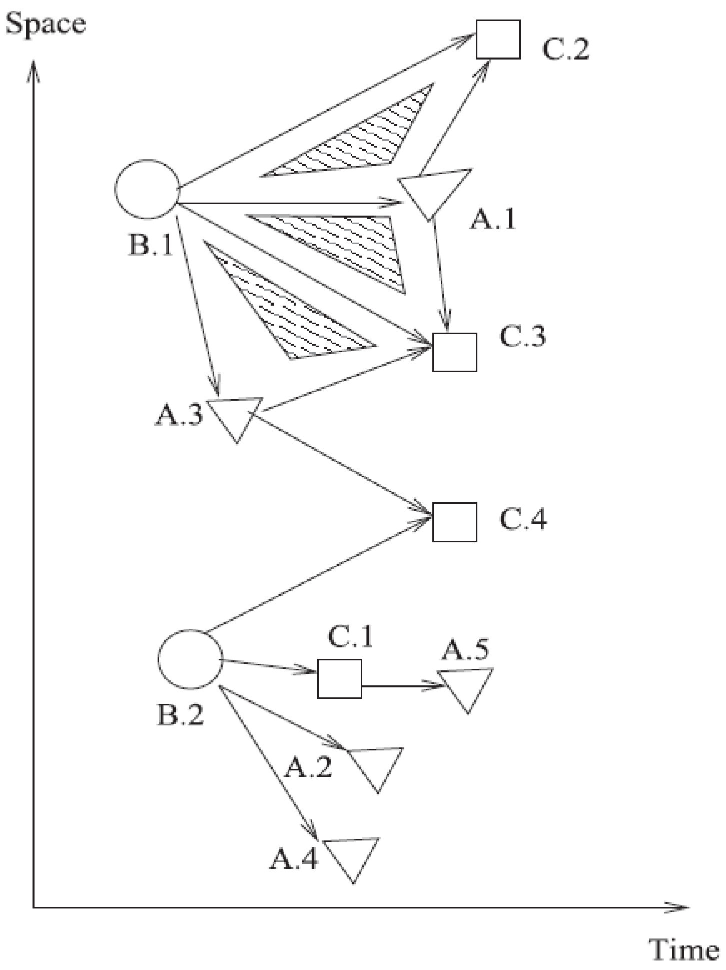
Building Blocks for CSTP

- Spatiotemporal neighbor graph

- New interest measure
  - Cascading spatiotemporal index
  - Cascading spatiotemporal ratio

Spatiotemporal Neighbor Graph

- Data: spatiotemporal (ST) neighbor graph
  - G=<N, E>
  - A graph defined on a set of point instances Point(x, y, time, eventType);
  - Neighbor relations are modeled by two thresholds:
    - Spatial distance interval [d0, d1]
    - Time distance interval [t0, t1]
  - For point P1 and P2, a directed edge (P1->P2) is added if distance(P1, P2) ∈ [d0, d1] and time(P1, P2) ∈ [t0, t1], and P1.time < P2.time.



Cascading Participation Index

- Interest measure: Cascading participation index (CPI)

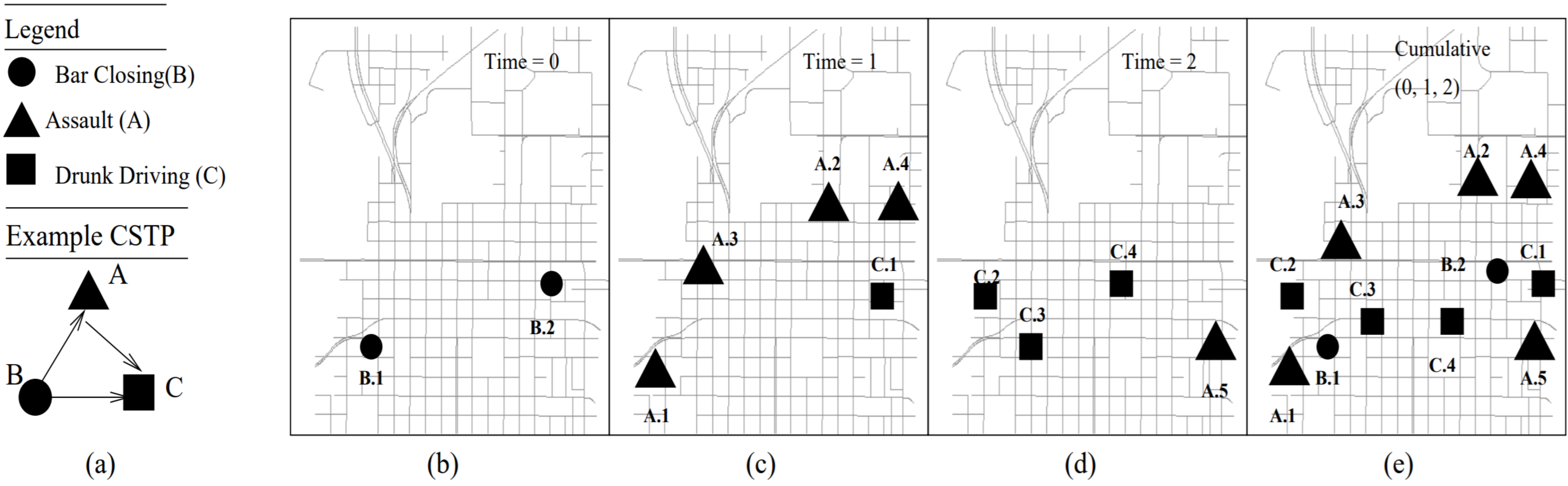
$$CPR(CSTP, M) = \frac{\#instances\ (M)\ participating\ in\ CSTP}{\#instances\ (M)\ in\ DataSet},$$

$$CPI = min\{CPR(CSTP, M)\}$$

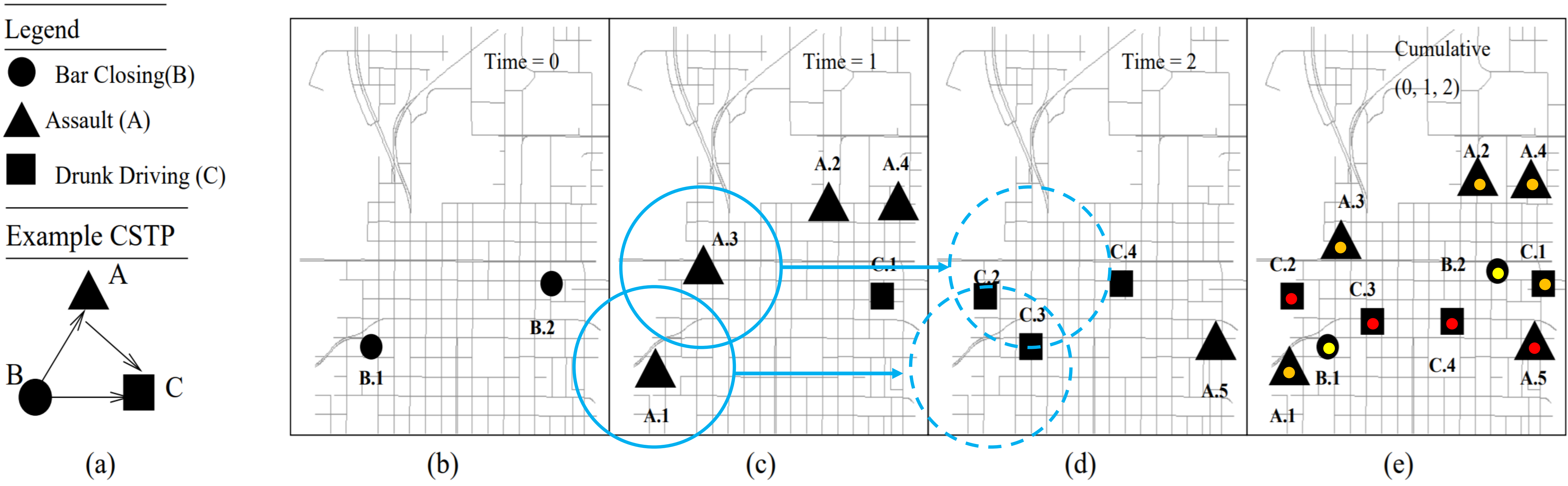
where M is a participating event type in the CSTP.

- Anti-monotonicity: If $CSTP_1 \subset CSTP_2$, $CPI(CSTP_1) \geq CPI(CSTP_2)$
  - Do association rule and colocation mining have the same property?

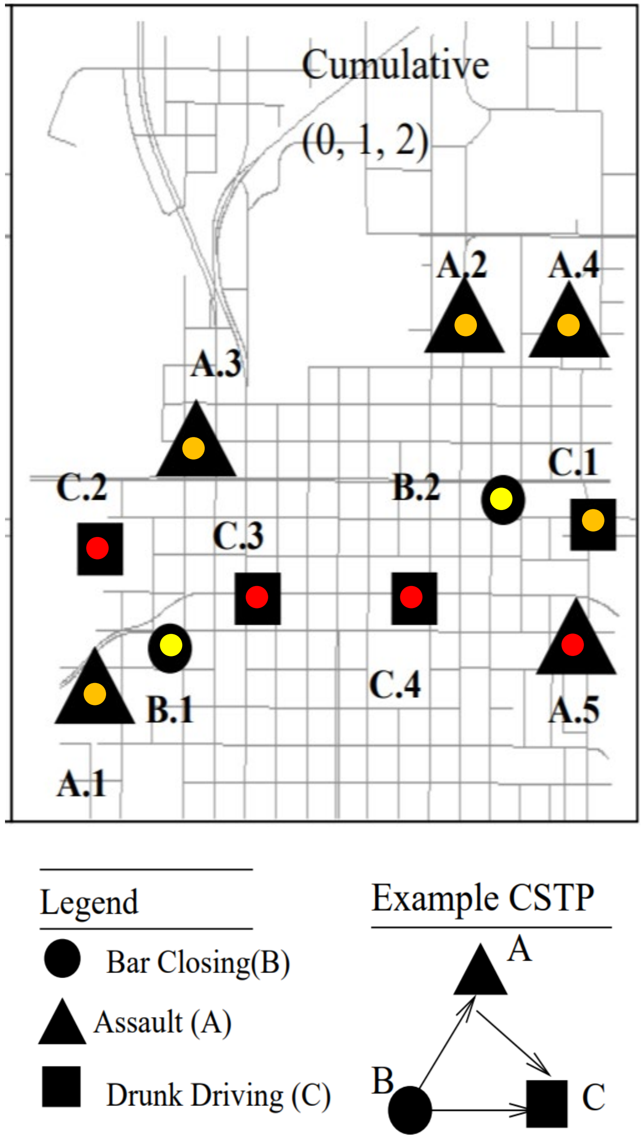
Cascading Example

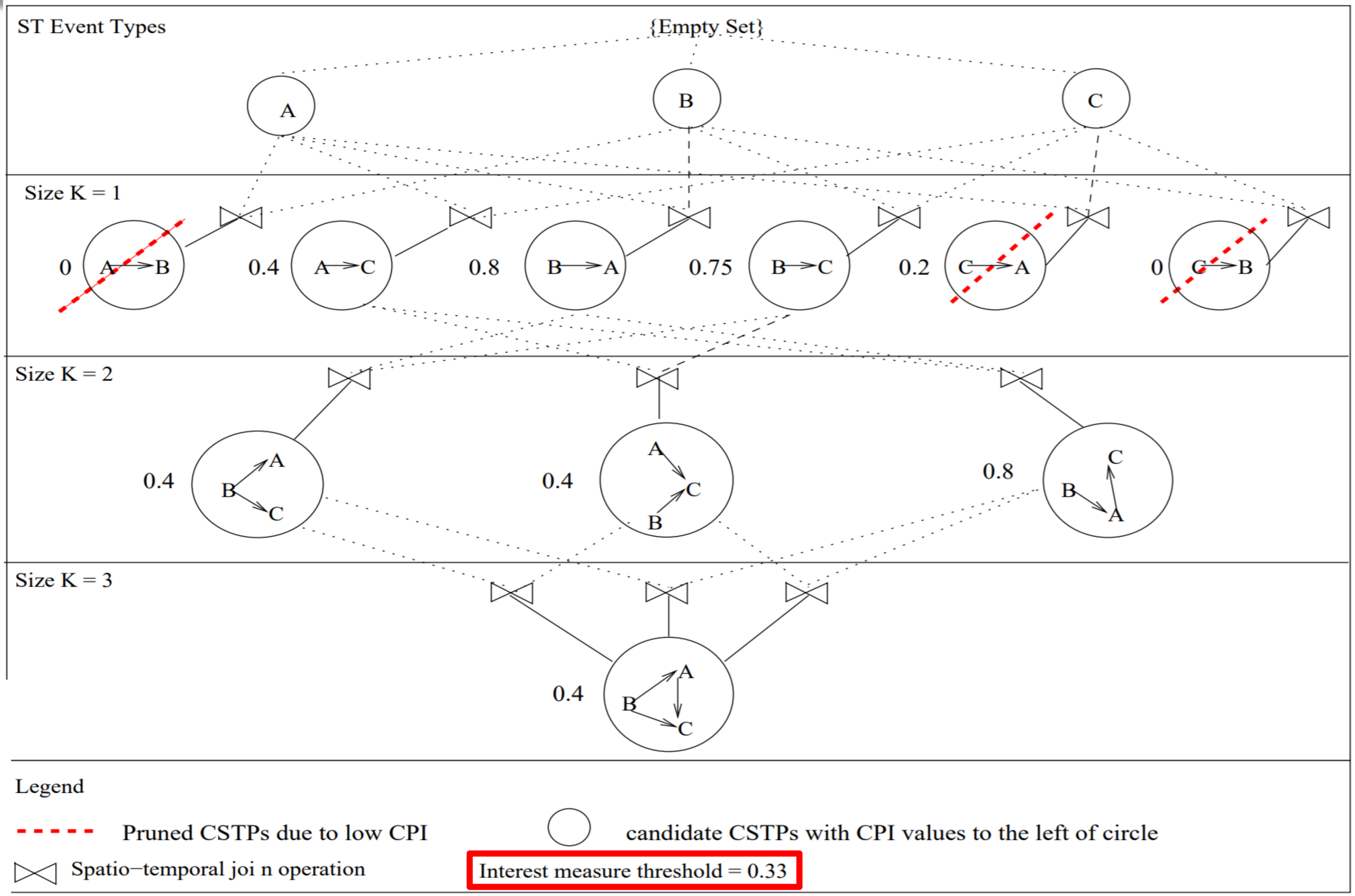


Cascading Example

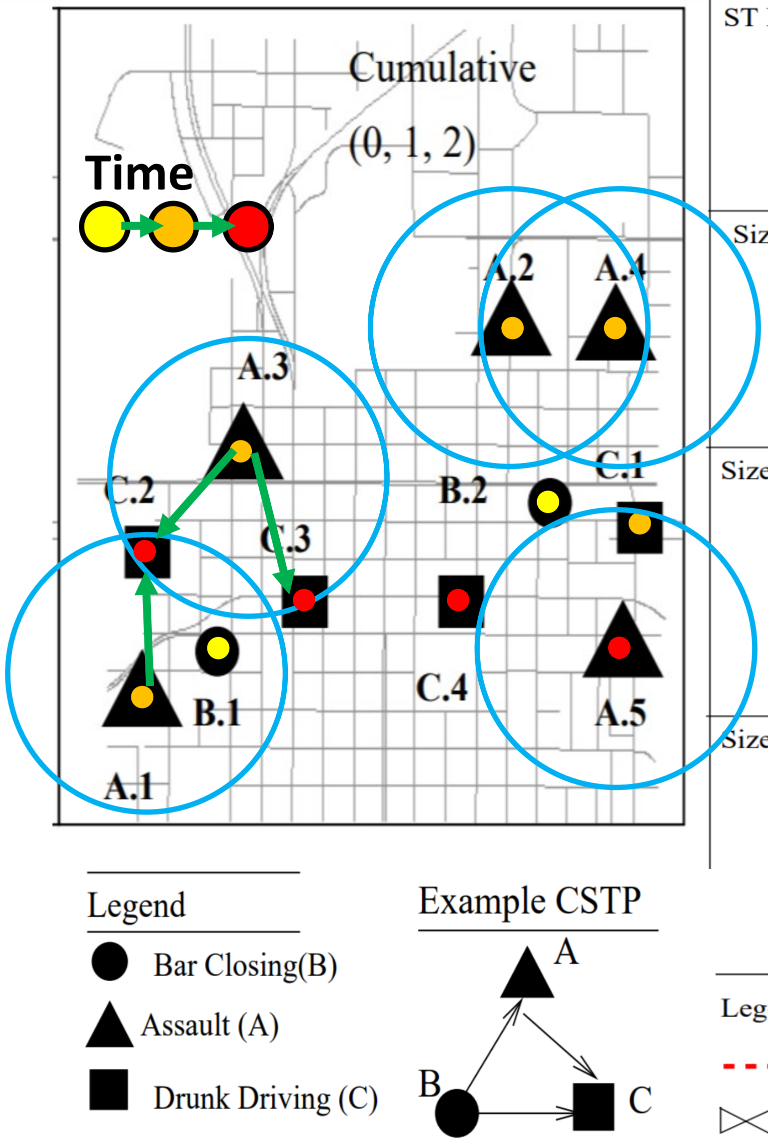


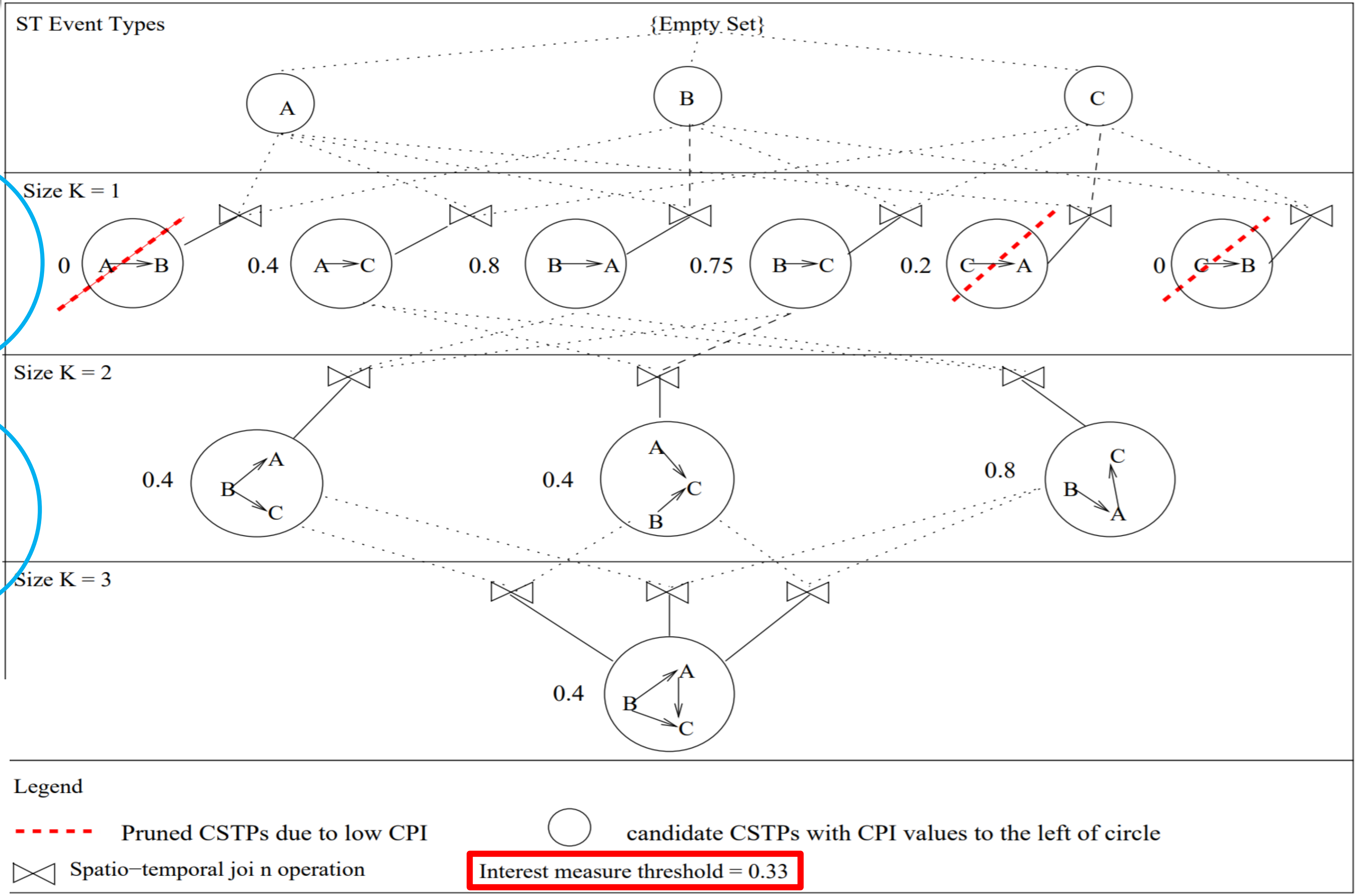
Cascading Example





Cascading Example





Cascading Example