Spatial Association Rules

nanopoulos@ismll.de,
buza@ismll.de

Outline

1. Motivation (examples for frequent patterns and association rules)
2. Association rule mining
3. Mining spatial association rules

Examples of frequent patterns

1. Products typically bought together in a supermarket
2. Co-occurring words in texts
3. Recurrent parts (motifs) in time series
4. Tags used together in social tagging systems
5. Diseases appearing together
6. Animals/plants living in symbiosis
7. …

Textual Patterns

Lars ***is from*** *Germany. Alex* ***is from*** *Greece. They both like* ***reading books****. Tomas comes from Slovakia, he also likes* ***reading books****. Do you know someone else, who enjoys* ***reading books****?*

...

Do you like Malgorzata from Poland? She must know Tomas, because Poland is adjacent to Slovakia and Tomas ***is from*** *Slovakia.*

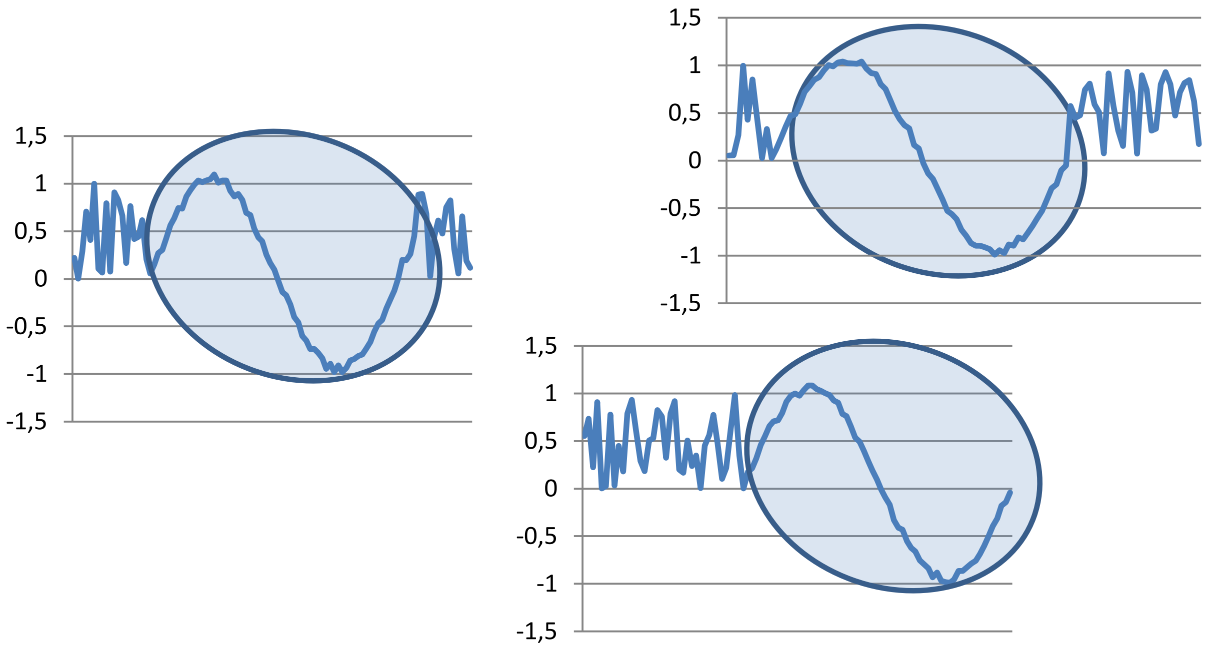
Application: Information extraction

Person	ComesFrom
Lars	Germany
Alex	Greece
Tomas	Slovakia

Motifs in time series

Motif: approximately repeated local pattern in time series

Application: e.g. medical diagnosis



Symbiotic species

Nile Crocodile &
Egyptian Plover

Images from
www.wikipedia.org

Example: Association rules

$\{ \text{Diaper} \} \rightarrow \{ \text{Beer} \}$

$\{ \text{Milk, Cheese} \} \rightarrow \{ \text{Bread, Sausage} \}$

Outline

1. Motivation (examples for frequent patterns and association rules)
2. Association rule mining
3. Mining spatial association rules

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} → {Beer},
{Milk, Bread} → {Eggs, Coke},
{Beer, Bread} → {Milk}

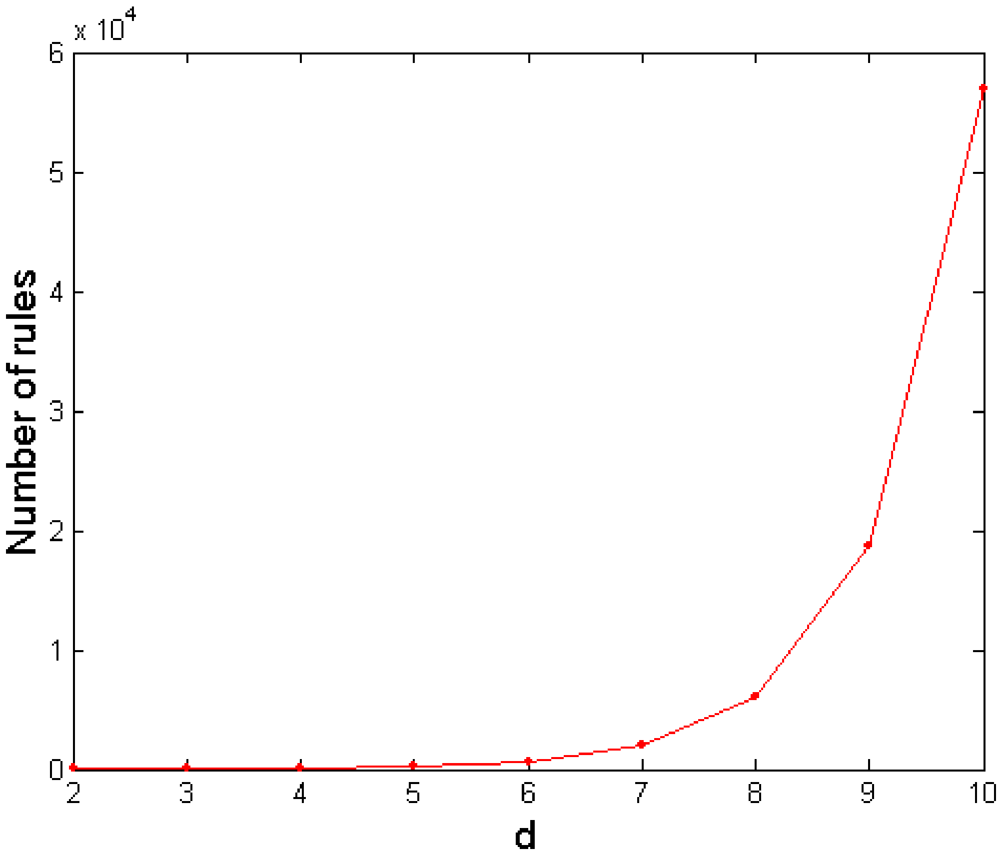
Implication means co-occurrence, not causality!

Many possible rules!

Given d unique items:

Total number of sets of items = 2^d

Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

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 $X \rightarrow Y$, where X and Y are itemsets
  - Example:
    $\{Milk, Diaper\} \rightarrow \{Beer\}$

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**

- 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

Example:

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

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

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

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

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} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {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

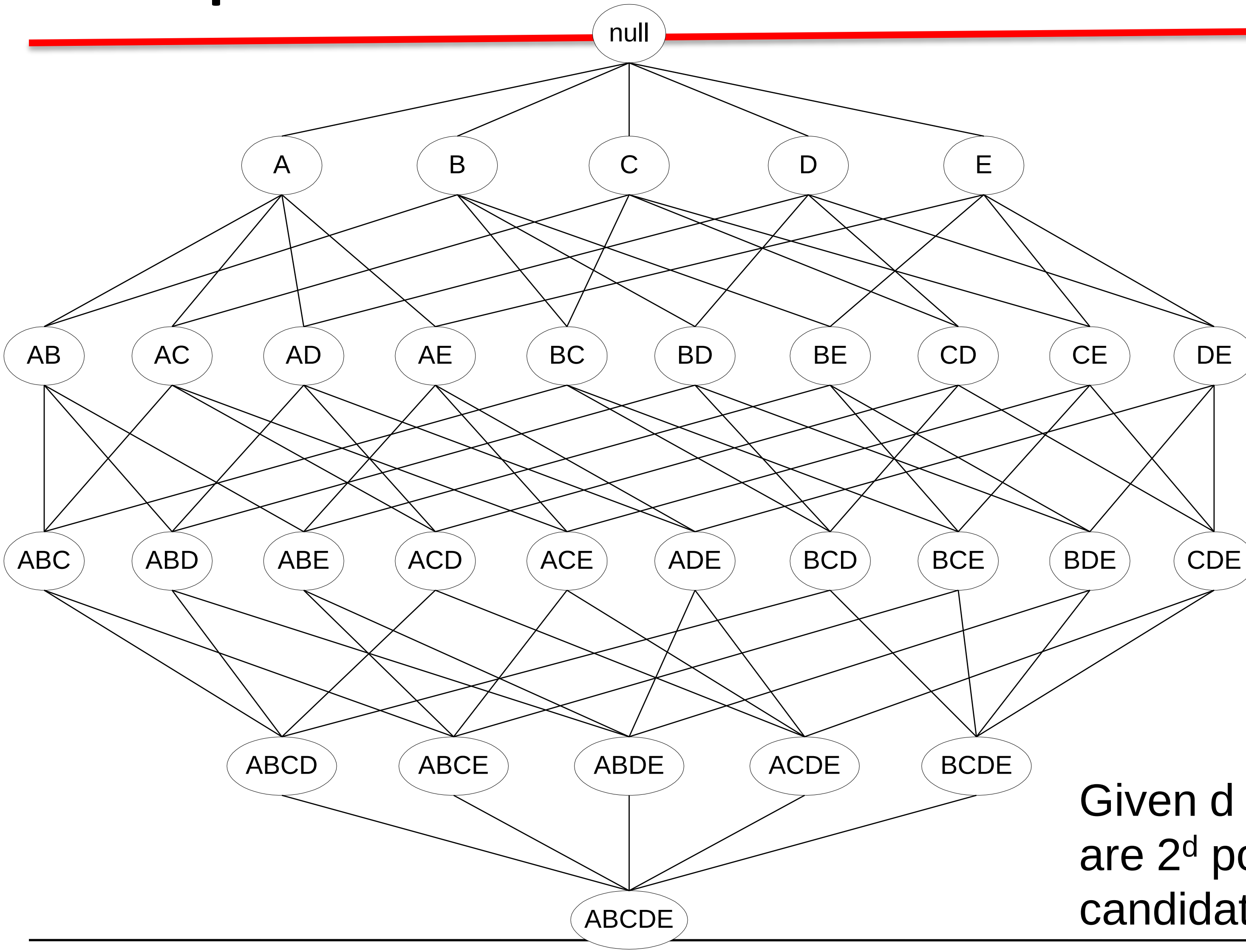
2. Rule Generation
   - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent itemset generation is the most computationally expensive

Generating Frequent Itemsets: Naive algorithm

```
d ← |I|
N ← |D|
for each subset x of I do
    σ(x) ← 0
    for each transaction T in D do
        if x is a subset of T then
            σ(x) ← σ(x) + 1
    if minsup <= σ(x)/N then
        add s to frequent subsets
```

The powerset of an itemset



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

Analysis of naive algorithm

$O(2^d)$ subsets of *I*

Scan n transactions for each subset

$O(2^d n)$ tests of s being subset of T

Growth is exponential in the number of items!

Can we do better?

Frequent Itemset Generation Strategies

Reduce the number of candidates (M)

- Complete search: $M=2^d$
- Use pruning techniques to reduce M

Reduce the number of comparisons (NM)

- Use efficient data structures to store the candidates or transactions
- No need to match every candidate against every transaction

Reducing Number of Candidates

Apriori principle:

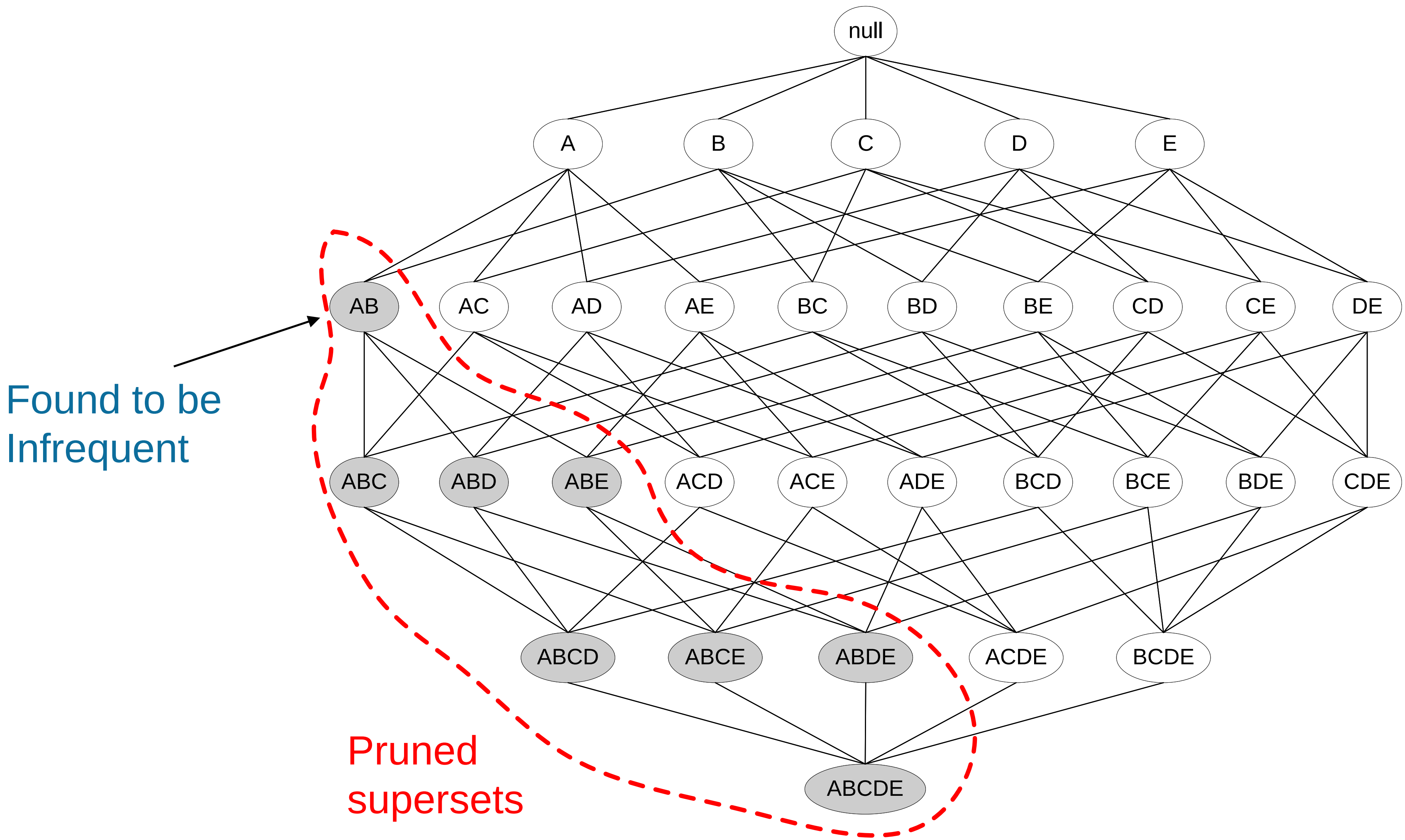
- If an itemset is frequent, then all of its subsets must also be frequent

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)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
$^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
$6 + 6 + 1 = 13$

The Apriori Algorithm

Join Step: C_k is generated by joining L_{k-1} with itself

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Pseudo-code:

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

```
L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) do begin
    C_k+1 = candidates generated from L_k;
   for each transaction t in database do
         increment the count of all candidates in C_k+1
       that are contained in t
   L_k+1  = candidates in C_k+1 with min_support
   end
return ∪_k L_k;
```

Example of Generating Candidates

L_3={*abc, abd, acd, ace, bcd*}

Self-joining: L_3*L_3

abcd from *abc* and *abd*

acde from *acd* and *ace*

Pruning:

acde is removed because *ade* is not in L_3

C_4={*abcd*}

The Apriori Algorithm — Example

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D ←

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

←

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D →

L_3

itemset	sup
{2 3 5}	2

Another example

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

minsup (count) >= 2

k=1, 2

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

1-Itemsets	Sup-count
1	6
2	7
3	6
4	2
5	2

2-Itemsets	Sup-count
1, 2	4
1, 3	4
1, 4	1
1, 5	2
2, 3	4
2, 4	2
2, 5	2
3, 4	0
3, 5	1
4, 5	0

Frequent 2-Itemsets	Sup-count
1, 2	4
1, 3	4
1, 5	2
2, 3	4
2, 4	2
2, 5	2

k=3

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Frequent 2-Itemsets	Sup-count
1, 2	4
1, 3	4
1, 5	2
2, 3	4
2, 4	2
2, 5	2

Frequent 3-Itemsets	Sup-count
1, 2, 3	2
1, 2, 5	2

Factors Affecting Complexity

- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
  - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - more space is needed to store support count of each item
  - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database
  - since Apriori makes multiple passes, run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
  - This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

Generating rules (2nd sub-problem)

Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement

If {A,B,C,D} is a frequent itemset, candidate rules:

ABC $\rightarrow$D,	ABD $\rightarrow$C,	ACD $\rightarrow$B,	BCD $\rightarrow$A,
A $\rightarrow$BCD,	B $\rightarrow$ACD,	C $\rightarrow$ABD,	D $\rightarrow$ABC
AB $\rightarrow$CD,	AC $\rightarrow$ BD,	AD $\rightarrow$ BC,	BC $\rightarrow$AD,
BD $\rightarrow$AC,	CD $\rightarrow$AB,		

If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Rule Generation with anti-monotone property

How to efficiently generate rules from frequent itemsets?

In general, confidence does not have an anti-monotone property

$c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$

But confidence of rules generated from the same itemset has an anti-monotone property

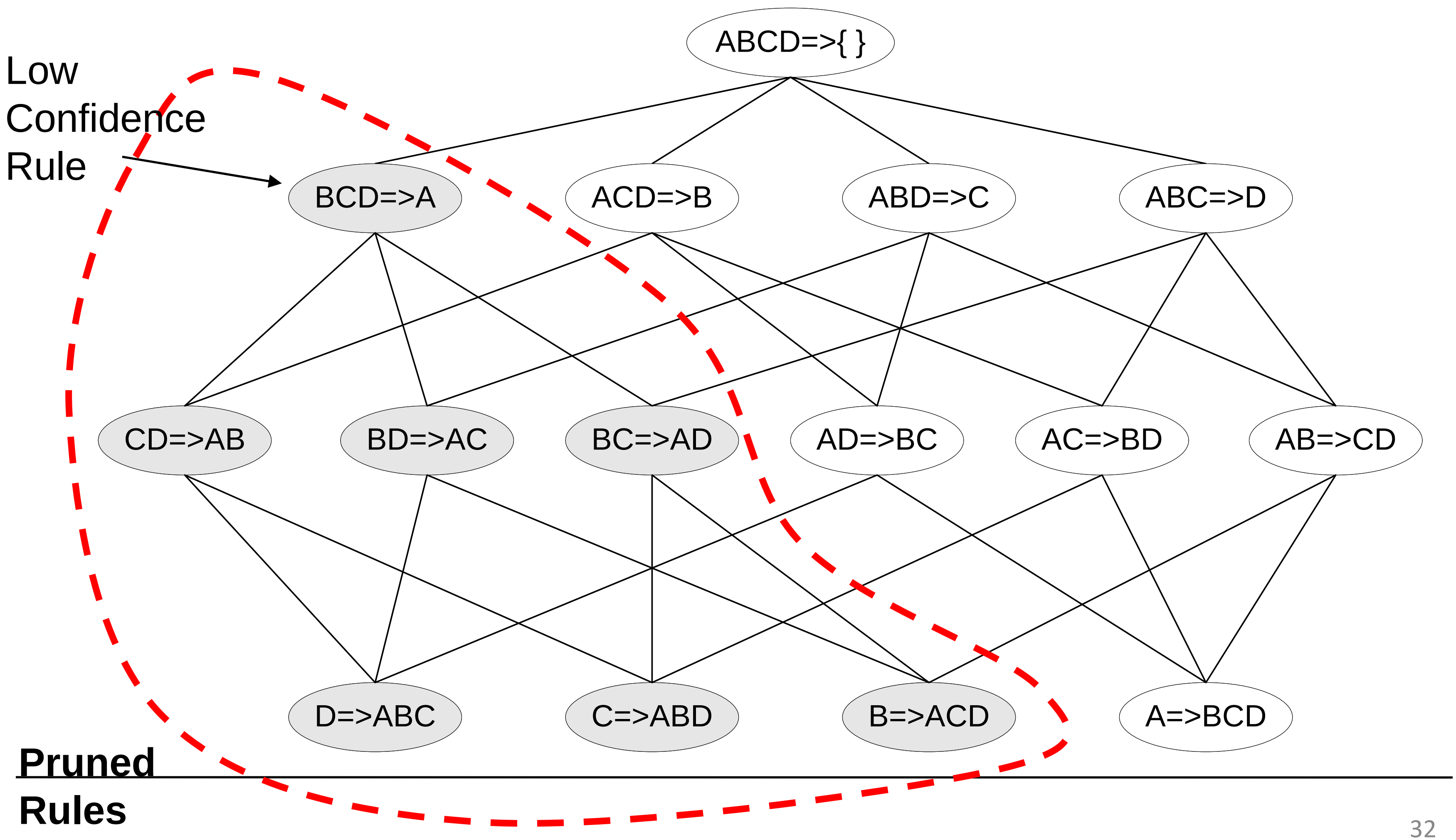
e.g., $L = \{A,B,C,D\}$:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

Rule Generation: example of anti-monotonicity

Lattice of rules



Example with confidence

3 association rules:

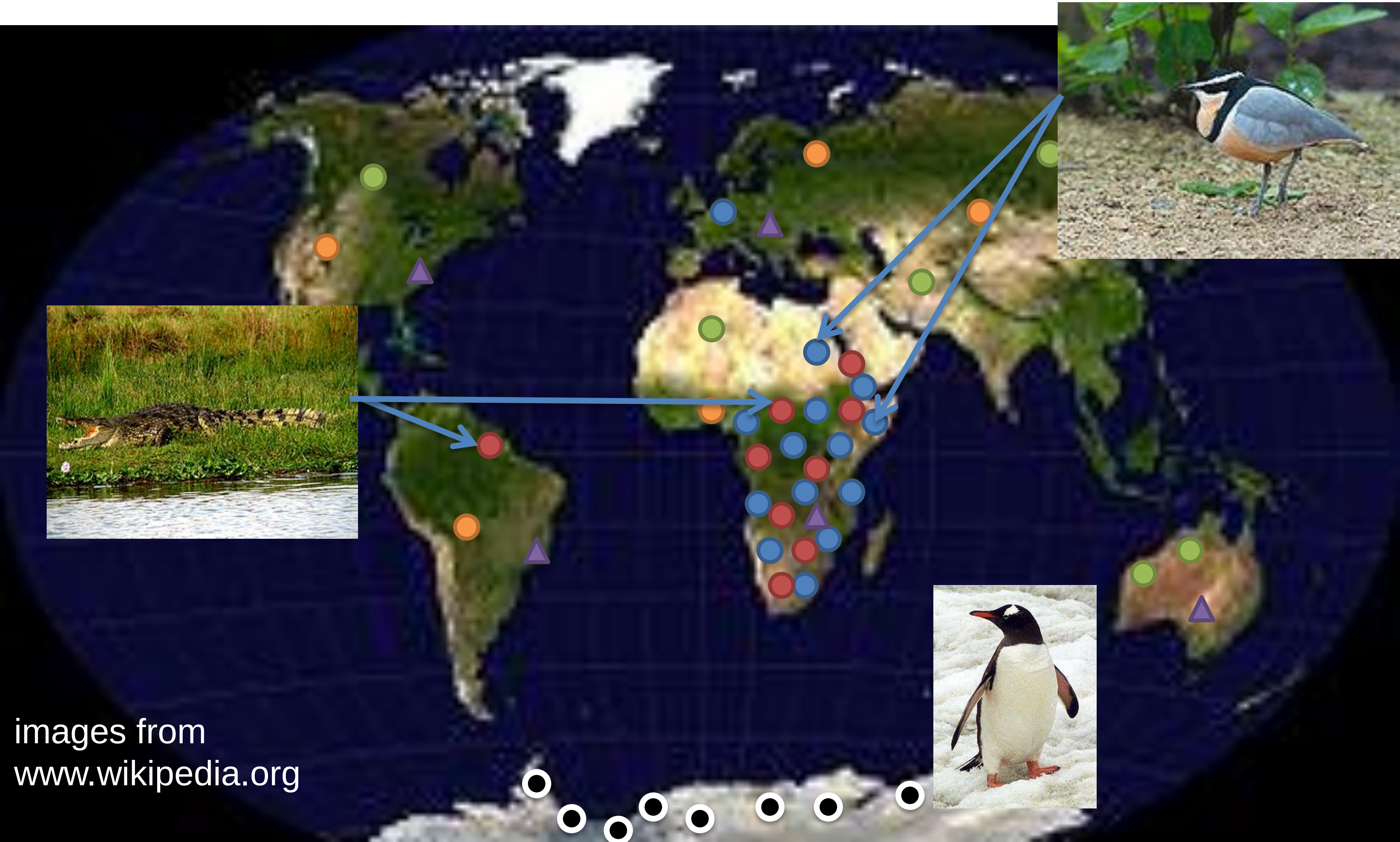
- {p} => {q} with confidence C1
- {p} => {q, r} with confidence C2
- {p, r} => {q} with confidence C3.

If C1, C2, C3 are unequal, give possible relation (inequalities) between them. Which one is bigger?

Outline

1. Motivation (examples for frequent patterns and association rules)
2. Association rule mining
3. Mining spatial association rules

Co-locations, Spatial association rules

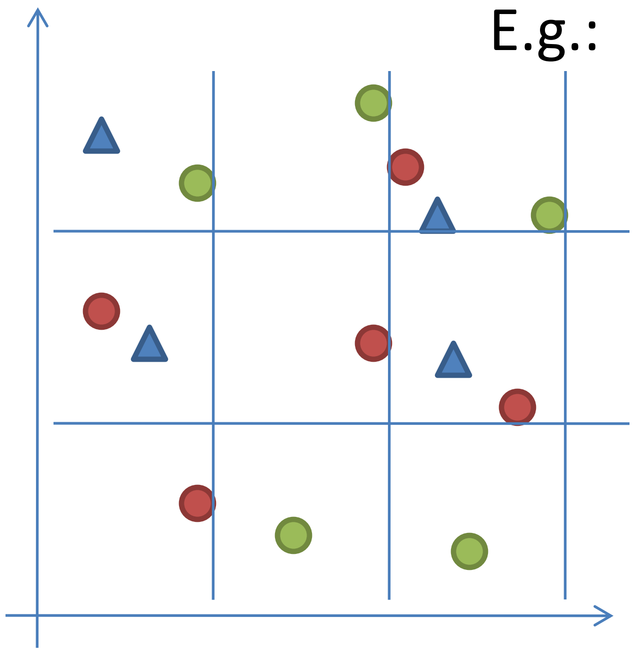


Approaches for finding co-location rules

- Spatial statistics
- Data Mining
  - Clustering-based
  - Association rule-based
    - Transaction-based
    - Distance-based

Transaction-based approaches

Project spatial data to a transactional database and apply frequent itemset mining



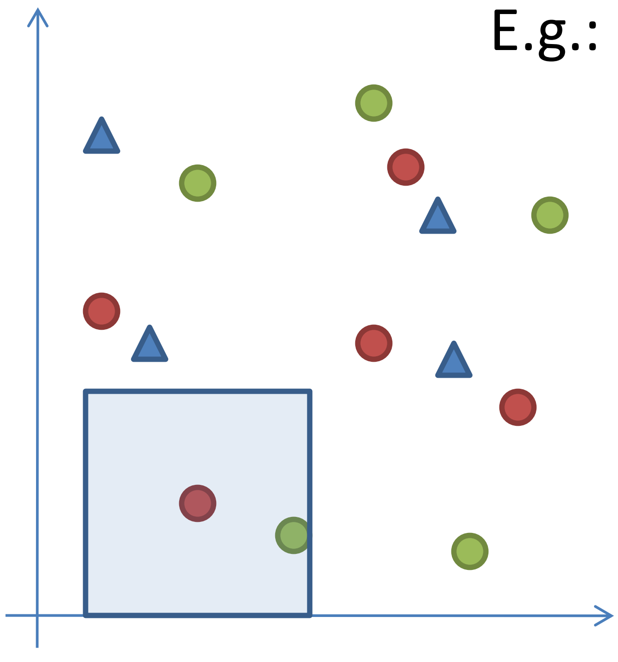
E.g.:

- disjoint windowing (according to a grid)
- reference feature centric model
- transactions for all instances
- …

Problems: over-counting, under-counting, rules for only one feature only

Transaction-based approaches

Project spatial data to a transactional database and apply frequent itemset mining



E.g.: - disjoint windowing (according to a grid)

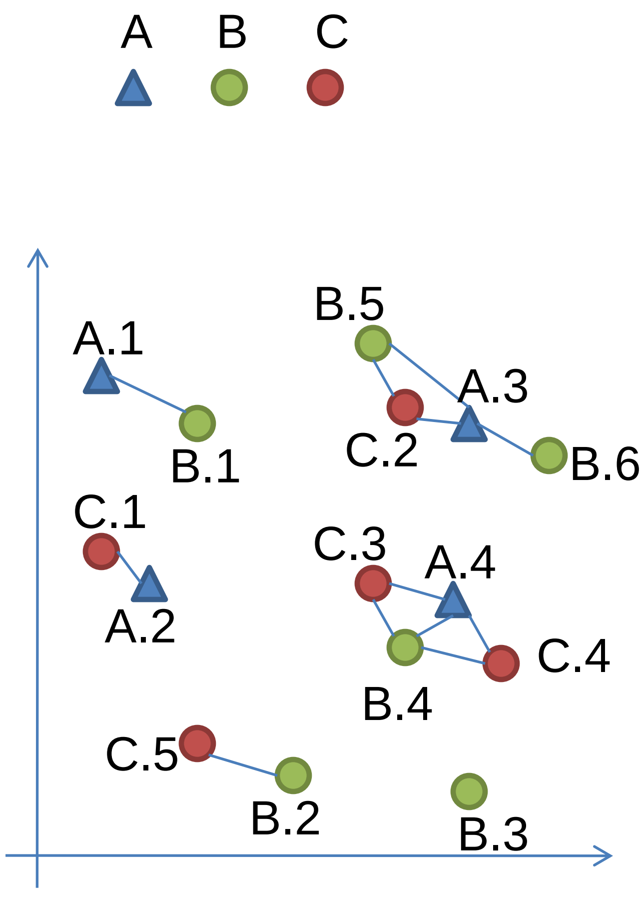
- reference feature centric model

- transactions for all instances

- ...

Problems: over-counting, under-counting, rules for only one feature only

Distance-based approach



Given:

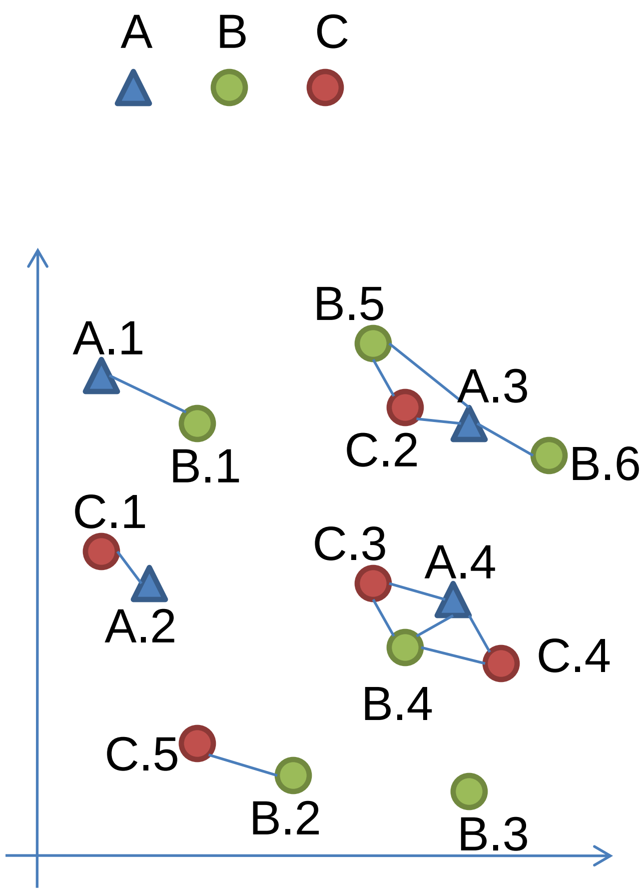
1) set T spatial feature types: T = {A,B,C,…}

2) their instances $I = \{i_1, i_2, \ldots i_N\}$ each instance is a vector: (id, type, location)

3) reflexive and symmetric neighbor relation R over instances in I

Task: find co-located spatial features (subsets and rules)

Distance-based approach



Co-location *c* is a **subset** of feature types, e.g. {B,C}.

Row instance of co-location {B,C}:

{B.5, C.2}

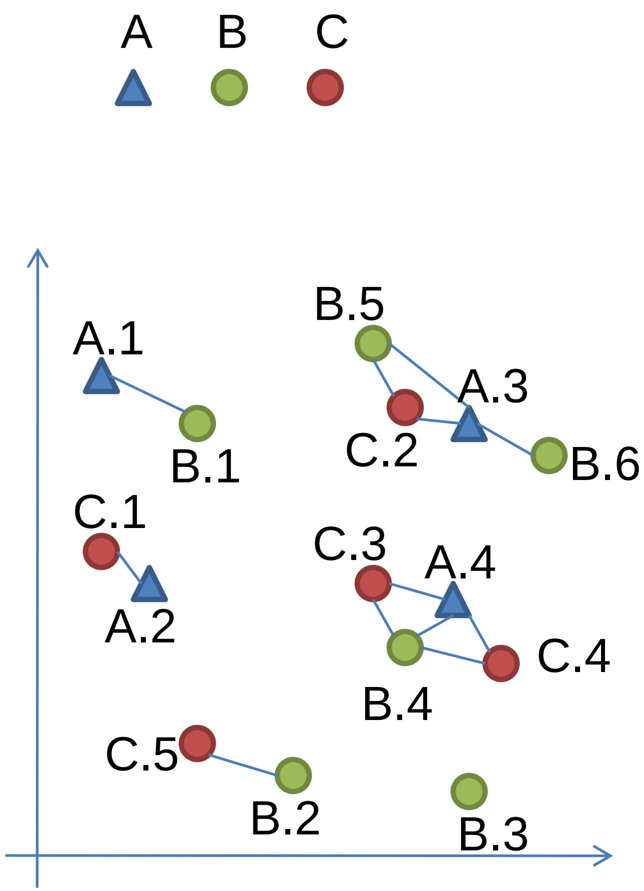
Table instance of co-location {B,C}:

table_instance({B,C}) = { {B.5, C.2}, {B.2,C.5}, {B.4, C.3}, {B.4, C.4} }

Projection with duplicate elimination:

π_B(table_instance({B,C})) = {B.2, B.5, B.4}

Distance-based approach



Participation ratio: $pr(c, F) =$

$|\pi_F(\text{table\_instance}(c))| \;/\; |\text{table\_instance}(F)|$

Participation index: c = {A,B,...}

$pi(c) = min\{pr(c,A), pr(c,B) \ldots \}$

Conditional probability:

$cp(c_1 \rightarrow c_2) =$

$|\pi_{c_1}(\text{table\_instance}(c_1 \cup c_2))| \;/$

$|\text{table\_instance}(c_1)|$

Distance-based approach

Co-Location Mining Algorithm

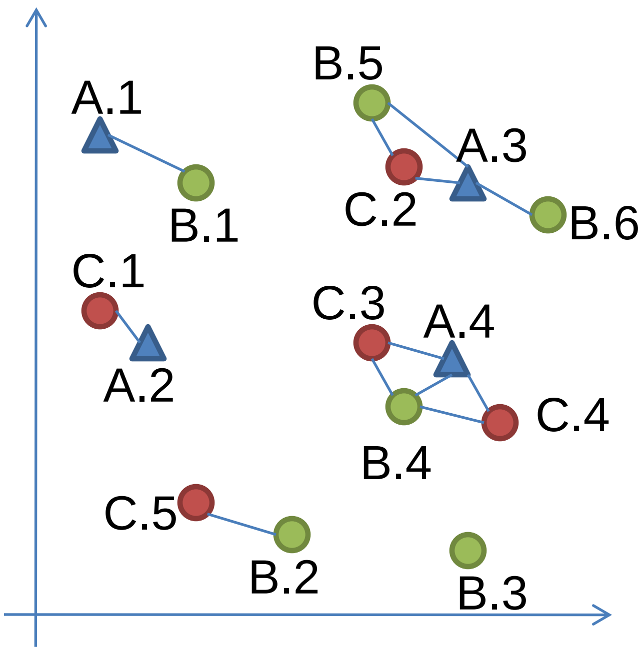
1. Apriori-based, but there are differences
2. Participation index is used as support,
   conditional probability as confidence
3. All co-location of size 1 are frequent
   (participation index is 1 for all co-location of size 1)
4. Iteration steps
   1. Generation of candidate co-locations
   2. Generation of table-instances of candidate co-locations
   3. Pruning of infrequent co-locations
   4. Generation of co-location rules

Generation of table-instances candidate co-locations

Join table-instances of previously found frequent co-locations

Join constraints: 1. All features are equal, but last one

2. Neighbor relation R



table_inst({A,B}) =
{ {A.1, B.1},
{A.3, B.5},
{A.3, B.6},
{A.4, B.4} }

table_inst({A,C}) =
{ {A.2, C.1},
{A.3, C.2},
{A.4, C.3},
{A.4, C.4} }

table_inst({A,B}) =
{ {A.3, B.5, C.2},
{A.4, B.4, C.4} }

Literature

http://www.spatial.cs.umn.edu/paper_list.html

Y. Huang, S. Shekhar: *Discovering Co-location Pattern from Spatial Datasets: A General Approach*, IEEE TKDE, 2004

S. Shekhar, Y. Huang: *Discovering Spatial Co-location Patterns: A Summary of Results*, 7th Int'l. Symp. on Spatial and Temporal Databases, 2001