HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Spatial Data Mining
Co-location rules

Antti Leino ‹antti.leino@cs.helsinki.fi›

Department of Computer Science

Background: frequent sets

- Frequently co-occurring items in transaction data
  - Finite set of disjoint transactions
  - E.g. customer data derived from supermarket cash registers
  - Well-known problem since the early 1990's
- Next step: association rules
  - $\{A_1,\ldots,A_n\} \Rightarrow B$
  - Confidence: $\hat{\mathrm{P}}(B|\{A_i\}) = \frac{|\{A_i,B\}|}{|\{A_i\}|}$
  - Support: $\hat{\mathrm{P}}(\{A_i,B\}) = \frac{|\{A_i,B\}|}{|\mathscr{R}|}$

Frequent sets: Apriori

- Classic algorithm for finding frequent sets
  - Two independent formulations in 1993–94
- Start with all pairs of items that are sufficiently frequent
- As long as there are sets of size $n-1$,
  - Generate as candidates those sets of size n whose subsets of size $n-1$ are frequent
  - Accept as frequent those candidates that are in fact frequent

Apriori: example

- Transaction data

```
baby_food beer milk
baby_food beer mustard sausage
baby_food bread butter
baby_food bread butter cigarettes milk
baby_food bread diapers milk sausage
baby_food bread milk
baby_food butter candy cigarettes diapers
baby_food candy diapers mustard
beer bread butter mustard sausage
beer bread candy
beer bread milk mustard sausage
beer butter sausage
candy cigarettes
```

Apriori: example

- Limit: frequency ≥ 0.2
- 1st iteration: frequent items
  - {baby_food:0.62, beer:0.46, mustard:0.31, bread:0.54, butter:0.38, candy:0.31, cigarettes:0.23, diapers:0.23, milk:0.38, sausage:0.38}
- 2nd iteration: pairs
  - Candidates: all pairs of the above
  - Frequent: {(baby_food,bread):0.31, (baby_food,diapers):0.23, (baby_food,milk):0.31, (beer,bread):0.23, (beer,mustard):0.23, (beer,sausage):0.31, (bread,butter):0.23, (bread,milk):0.31, (bread,sausage):0.23, (mustard,sausage):0.23}

Apriori: example

- 3rd iteration: triplets
  - Candidates: {(baby_food,bread,milk), (beer,bread,sausage), (beer,mustard,sausage)}
  - Frequent: {(baby_food,bread,milk):0.23, (beer,mustard,sausage):0.23}
- 4th iteration: quadruplets
  - No more candidates

Association rules

- The example discovered some frequent sets
- Association rules can be derived from those
  - Sets (beer,mustard,sausage):0.23 and (beer,sausage):0.31
  - Rule (beer,sausage) $\Rightarrow$ mustard
    - Support: 0.23
    - Confidence: $\frac{0.23}{0.31} \approx 0.7$
  - Sets (baby_food,diapers):0.23 and (diapers):0.23
  - Rule diapers $\Rightarrow$ baby_food
    - Support: 0.23
    - Confidence: 1

From transactions to spatial data

- Transactions are disjoint
- Spatial co-location is not
- Something must be done
- Three main options
  1. Divide the space into areas and treat them as transactions
  2. Choose a reference point pattern and treat the neighbourhood of each of its points as a transaction
  3. Treat all point patterns as equal

Window-centric co-location mining

- Divide the space into areas
  - Create a uniform grid that covers the space
  - See which phenomena occur in each grid cell
  - Treat grid cells as transactions
- Easy: just use transaction-based algorithms
- Useful for large-scale co-location rules
  - Correlations between the distributions of the different phenomena on e.g. national scale
- Not very useful for small-scale co-locations
  - Noise level increases as the size of grid cells decreases

Reference feature centric co-location mining

- Choose one point pattern as the reference
- Find the neighbourhood of each point in the reference pattern
- Treat these as transactions
- Again, relatively easy to use transaction-based algorithms
- Useful for applications where there is an obvious choice for the reference phenomenon
- Not very useful when there is no such candidate

Event-centric co-location mining

- Large number of different point patterns
  - Each describe the existence of a phenomenon
  - These phenomena are considered equal
- Transaction-based algorithms not immediately applicable
- More general than the other two approaches
- Still, only binary phenomena
  - Each point describes the existence of something
  - More detailed properties – e.g. temperature scale – must be discretised as a preprocessing step

Mining without transactions

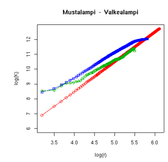
- Possible to adapt Apriori for event-centric co-location mining
  - Needed: a measure for co-occurrence
  - Apriori uses frequency of (A,B)
- Find co-occurring pairs
- Use an Apriori-derivative to find larger sets

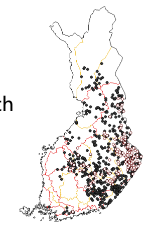
Measuring spatial attraction

- Spatial statistics: the K function
- In its basic form, for a single point pattern, $\lambda K(h) =$ E(number of points within radius h of a random point)
  - If no spatial correlation, $K(h) = \pi h^2$
  - Attraction: $K(h) > \pi h^2$
  - Repulsion: $K(h) < \pi h^2$
- Correlation between two point patterns: $\lambda_2 K_{12}(h) =$ E(number of points of type 2 within radius h of a random point of type 1

Combining K and Apriori

- Calculate the K_{12} function for each pair of point patterns
- Use these as the measure for co-occurrence
  - Accept those sets where K_{12} for each pair exceeds a set limit
- Example: two place names with significant attraction
  Mustalampi 'Black Pond'
  Valkealampi 'White Pond'





Apriori and the K function: example

- Raw data: Finnish lake names
  - Preprocessing: select those with ≥ 20 occurrences
  - This gives 331 names and 19 230 lakes
- Criterion: $K_{12}(1\,000) > 20\,000\,000\pi$ (units: metres)

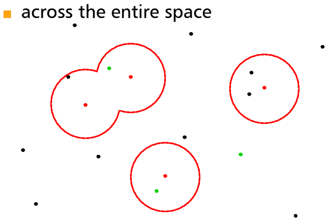
Set size	Number of sets	Distinct pairs
4	2	12
3	104	255
2	638	638
2–4	744	903

Apriori and the K function: results

- Some interesting co-location patterns:
  - (*Myllyjärvi* 'Mill Lake', *Kirkkojärvi* 'Church Lake')
  - (*Kaitajärvi* 'Narrow Lake', *Hoikkajärvi* 'Thin Lake')
  - (*Mäntyjärvi* 'Pine Lake', *Mäntylampi* 'Pine Pond')
  - (*Iso Haukilampi* 'Greater Pike Pond', *Pieni Haukilampi* 'Lesser Pike Pond')
  - (*Ahvenlampi* 'Perch Pond', *Haukilampi* 'Pike Pond')
  - (*Alalampi* 'Low Pond', *Keskilampi* 'Middle Pond', *Ylilampi* 'High Pond')
  - Also a lot of noise
- Several co-location patterns can be interpreted in terms of linguistics
- Insight into properties of the name system and the name-giving process

Co-locations without K

- K function is
  - statistically justifiable
  - computationally expensive
- Simpler method: frequency of points
  - in the neighbourhood of points in another pattern
  - across the entire space



Points in a neighbourhood

- If point patterns A and B are independent,
  - The neighbourhood of the A points is a random sample of B points
  - The number of B points ~ Poisson(λ), where $\lambda =$ the number of all points in the neighbourhood × the overall frequency of B points
- For larger sets, select those points of type B whose neighbourhood contains points $A_i, \forall i$
  - If the point patterns are independent, this is still a random sample of B
  - This gives an association rule of $A_i \Rightarrow B$
- Assumptions
  - All point patterns (A,B,...) fundamentally similar
  - The point patterns do not have internal spatial correlation

Apriori and neighbourhoods

- Again, possible to adapt an Apriori-like algorithm
- Compute co-location pairs
- As long as there are co-location rules of size $n - 1$,
  - Generate candidates of size n
  - Accept those candidates that fulfill the criteria
- Problem: checking the neighbourhoods
  - Spatial operations are expensive

Minimising spatial operations

- In a database environment, spatial queries can be expensive
- Fortunately, they are not required all the time
- Sufficient to compute neighbourhoods once
  - Create a new database table that contains
    - Point-id
    - Which point pattern this one belongs to
    - Which point patterns have instances in the neighbourhood of this point
  - This table is sufficient for checking the candidates
  - Not necessary to do spatial queries in all iterations

Further development

- This is just a starting point for co-location mining
- Further optimisations are possible
  - Fine-tuning of Apriori-based algorithms
  - Different approaches
- The next three sessions will touch on these issues

Revised schedule
Week 12

19.3. Huang & al. 2004: Discovering Colocation Patterns from Spatial Data Sets: A General Approach
Joona Lehtomäki

- Salmenkivi 2006: Efficient Mining of Correlation Patterns in Spatial Point Data
Daniela Hellgren

22.3. Yoo & al. 2006: A Joinless Approach for Mining Spatial Colocation Patterns
(TBD)

- Huang & al. 2005: Can We Apply Projection Based Frequent Pattern Mining Paradigm to Spatial Colocation Mining?
Zoltán Bójás

Revised schedule
Week 13

26.3. Xiong & al. 2004: A Framework for Discovering Co-location Patterns in Data Sets with Extended Spatial Objects
Paula Silvonen

- Yoo & al. 2006: Discovery of Co-evolving Spatial Event Sets
Timo Nurmi

29.3. Introduction: spatial clustering

Revised schedule
Week 14

2.4. Tung & al. 2001: Spatial Clustering in the Presence of Obstacles
Milan Magdics

- Wang & Hamilton 2003: DBRS: A Density-Based Spatial Clustering Method with Random Sampling
Bence Novák

- Easter break

Revised schedule
Week 16

16.4. Introduction: spatial modelling

19.4. Kavouras 2001: Understanding and Modelling Spatial Change
Sandeep Puthan Purayil

- Kazar & al. 2004: Comparing Exact and Approximate Spatial Auto-Regression Model Solutions for Spatial Data Analysis
Magnus Udd

Revised schedule
Week 17

23.4. Shekhar & al.2003: A Unified Approach to Detecting Spatial Outliers
Pekka Maksimainen

- Hyvönen & al. (forthcoming): Multivariate Analysis of Finnish Dialect Data – an overview of lexical variation
Hanna Tikkanen

26.4. Summary