Other pattern domains and their primitives

Marc Plantevit

Université Claude Bernard Lyon 1 – LIRIS CNRS UMR5205

ENS Lyon, April 2018

Outline

Outside the itemset domain

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{\psi \in \mathcal{L} \mid \mathcal{C}(\psi, \mathcal{D}) \text{ is true}\}$$

- Pattern domain: (itemset, sequences, graphs, dynamic graphs, etc.)
- Constraints: How to efficiently push them?

Outside the itemset domain

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{\psi \in \mathcal{L} \mid \mathcal{C}(\psi, \mathcal{D}) \text{ is true}\}$$

- Pattern domain: (itemset, sequences, graphs, dynamic graphs, etc.)
- Constraints: How to efficiently push them?

Considering more sophisticated pattern domain is more challenging!

- Some anti-monotonic properties do not hold:
  - freeness for sequence.
  - support within a single graph.
- Some pessimistic results (non derivability outside itemset domain)

Outside the itemset domain

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{\psi \in \mathcal{L} \mid \mathcal{C}(\psi, \mathcal{D}) \text{ is true}\}$$

- Pattern domain: (itemset, sequences, graphs, dynamic graphs, etc.)
- Constraints: How to efficiently push them?

Considering more sophisticated pattern domain is more challenging!

- Some anti-monotonic properties do not hold:
  - freeness for sequence.
  - support within a single graph.
- Some pessimistic results (non derivability outside itemset domain)

But it makes it possible to capture more meaningful patterns.
☞ it's worth it!

Outline

Sequence mining

Key notions

- $\mathcal{I} = \{i_1, i_2 \dots i_m\}$ the items.
  - $\mathcal{I} = \{a, b, c, d, e\}$.
- *itemset*
  - (a, b).
- *A sequence is an ordered list of itemsets*
  - $\langle (a, b)(b)(a, c) \rangle$.
- the set of all possible sequences $\mathcal{I}$ is denoted $\mathbb{T}(\mathcal{I})$.
- Relation between sequences:

  Inclusion $\preceq$
  - $\langle (b)(c) \rangle \preceq \langle (a, b)(b)(a, c) \rangle$,
  - $\langle (c)(a) \rangle \not\preceq \langle (a, b)(b)(a, c) \rangle$.

Mining Task

Sequence database $\mathcal{D}$: a collection of pairs (SID, T), SID is an id and T is a sequence $\mathbb{T}(\mathcal{I})$.

SDB $\mathcal{D}$

S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)\rangle$

R. Agrawal and R. Srikant, 1996.

Mining Task

Sequence database $\mathcal{D}$: a collection of pairs (SID, T), SID is an id and T is a sequence $\mathbb{T}(\mathcal{I})$.

SDB $\mathcal{D}$

S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)\rangle$

Frequency

$Support(S, \mathcal{D}) = |\{(SID, T) \in \mathcal{D} | S \preceq T\}|$.

R. Agrawal and R. Srikant, 1996.

Mining Task

Sequence database $\mathcal{D}$: a collection of pairs (SID, T), SID is an id and T is a sequence $\mathbb{T}(\mathcal{I})$.

SDB $\mathcal{D}$

S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)\rangle$

Frequency

$Support(S, \mathcal{D}) = |\{(SID, T) \in \mathcal{D} | S \preceq T\}|.$

Relative Frequency

$freq_S^{\mathcal{D}} = \frac{Support(S,\mathcal{D})}{|\mathcal{D}|}.$

R. Agrawal and R. Srikant, 1996.

Mining Task

Sequence database $\mathcal{D}$: a collection of pairs (SID, T), SID is an id and T is a sequence $\mathbb{T}(\mathcal{I})$.

SDB $\mathcal{D}$

S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)\rangle$

Frequency

$Support(S, \mathcal{D}) = |\{(SID, T) \in \mathcal{D} | S \preceq T\}|.$

Relative Frequency

$freq_S^{\mathcal{D}} = \frac{Support(S,\mathcal{D})}{|\mathcal{D}|}.$

Sequence Pattern Mining Problem

$$FSeqs(\mathcal{D}, \sigma) = \{S \mid freq_S^{\mathcal{D}} \geq \sigma\}$$

R. Agrawal and R. Srikant, 1996.

Main Algorithms

Based on A Priori

- Candidate generation.
- Levelwise or depthfirst enumeration.
- GSP, SPAM, PSP, SPADE, etc.

Pattern-Growth

- No candidate generation.
- Depthfirst enumeration.
- Prefixspan.
- Key concept of projected database

Main Algorithms

Based on A Priori

- Candidate generation.
- Levelwise or depthfirst enumeration.
- GSP, SPAM, PSP, SPADE, etc.

Pattern-Growth

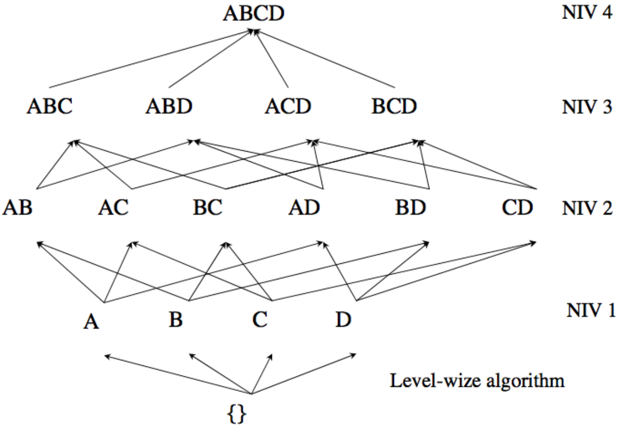
- No candidate generation.
- Depthfirst enumeration.
- Prefixspan.
- Key concept of projected database

SDB $\mathcal{D}$

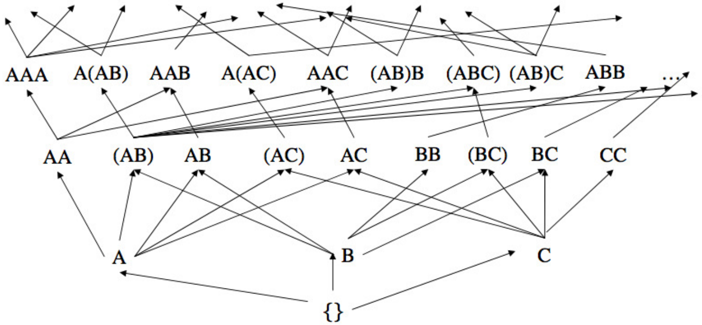
S_1	$\langle(a)(b)(c)(d)(a)(b)(c)\rangle$
S_2	$\langle(a)(b)(c)(b)(c)(d)(a)(b)(c)(d)\rangle$
S_3	$\langle(a)(b)(b)(c)(d)(b)(c)(c)(d)(b)(c)(d)\rangle$
S_4	$\langle(b)(a)(c)(b)(c)(b)(b)(c)(d)()\rangle$
S_5	$\langle(a)(c)(d)(c)(b)(c)(a)()\rangle$
S_6	$\langle(a)(c)(d)(a)(b)(c)(a)(b)(c)()\rangle$
S_7	$\langle(a)(c)(c)(a)(c)(b)(b)(a)(e)(d)()\rangle$
S_8	$\langle(a)(c)(d)(b)(c)(b)(a)(b)(c)()\rangle$

$\mathcal{D}_{|\langle(a)(b)(d)\rangle}$: the suffixes of the first occurrence of $\langle(a)(b)(d)\rangle$ in each data sequence.

Itemset: Search Space = A lattice



Sequential patterns: the search space



Antimonotonicity holds

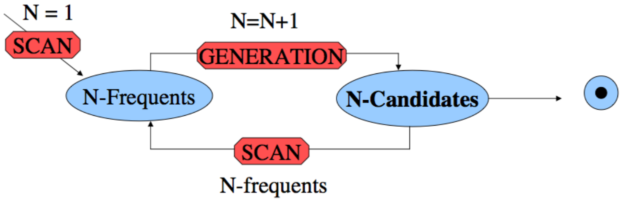
Important property [Agrawal et Srikant, 94]

- If S is not a frequent sequence
- Then none of its super-sequences is frequent
- Ex : $\langle hb \rangle$ not frequent then $\langle hab \rangle$ $\langle (ah)b \rangle$ cannot be frequent.

Seq. ID	Séquence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

minsupp =2

General Approach : generate and test



Candidate generation: 2 types of extension

S-Extension

a novel itemset to the sequence

- $\langle(a, b)(c)\rangle \rightarrow \langle(a, b)(c)(d)\rangle$

I-Extension

a novel item into an existing itemset of the sequence.

- $\langle(a, b)(c)\rangle \rightarrow \langle(a, b)(c, d)\rangle$

GSP (Agrawal et Srikant): based on Apriori

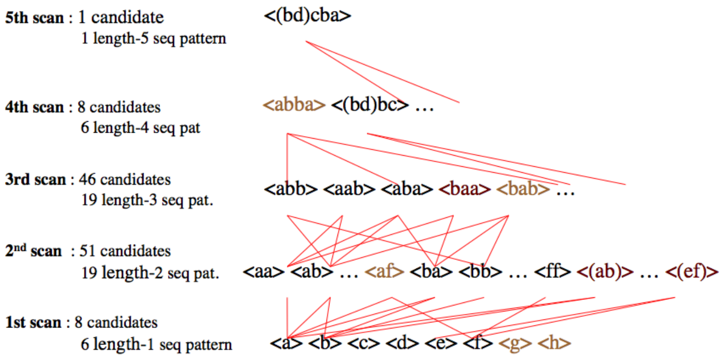
```
L=1
While (Result_L != NULL)
    Candidate Generate
    Prune
    Test
    L=L+1
```

One scan over the DB to discover the sequences that contain a single item.

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Cand	Sup
<a>	**3**
<b>	**5**
<c>	**4**
<d>	**3**
<e>	**3**
<f>	**2**
~~<g>~~	1
~~<h>~~	1

The whole process



Candidate generation of length 2

S-Extension

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

51 2-Candidates

I-Extension

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

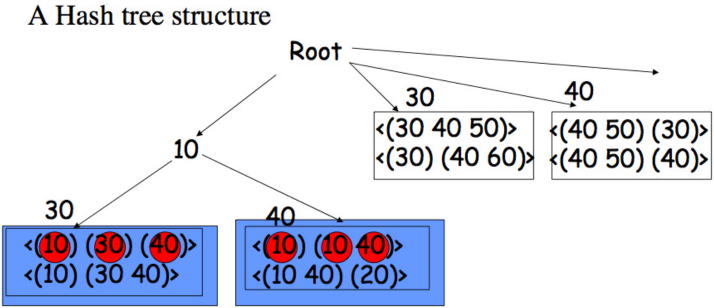
Without the antimonotonic property 8*8+8*7/2=92 candidates

Candidate support computation

the most costly step

- candidates are stored in central memory.
- limit the disk accesses.
- load the DB if possible.

How to efficiently stored the candidates?

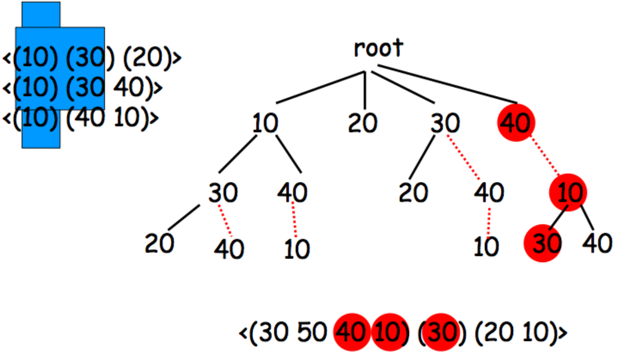


$S = \langle (10)(30)(10,40) \rangle$

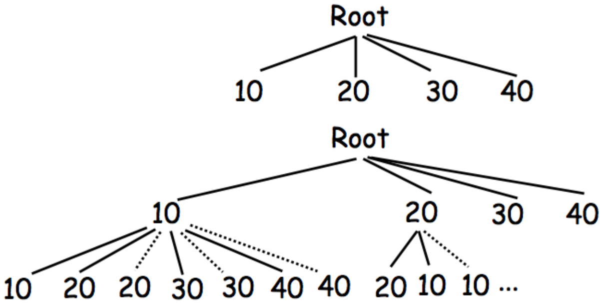
PSP (Prefix Tree for SP) [Masseglia et al. 98]

A more efficient structure based on prefix tree.

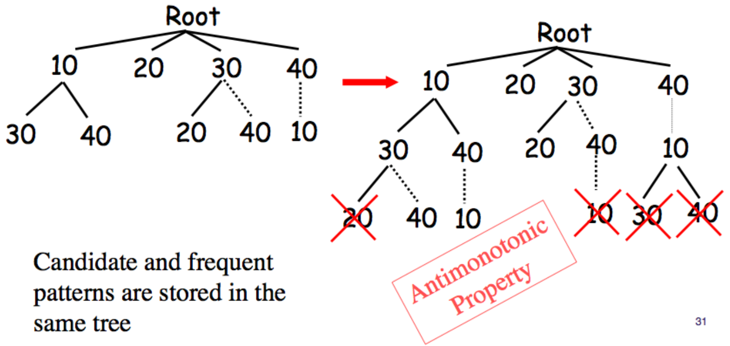
- 2 types of edges



PSP: generation of 2-candidates



PSP: generation ok k-candidates ($k > 2$)



Candidate and frequent patterns are stored in the same tree

Algorithm SPADE

SPADE (Sequential PAttern Discovery using Equivalent Class) [Zaki 2001]

- vertical representation of data
- DB: set of triples (item, SID,EID)

SPADE: more efficient support computation

SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

a		b		···
SID	EID	SID	EID	···
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

ab			ba			···
SID	EID (a)	EID(b)	SID	EID (b)	EID(a)	···
1	1	2	1	2	3	
2	1	3	2	3	4	
3	2	5				
4	3	5				

aba				···
SID	EID (a)	EID(b)	EID(a)	···
1	1	2	3	
2	1	3	4	

Limitation of generate-and-test methods

- An over-generation of candidates:
  - For 1000 frequent 1-sequences, $1000 \times 1000 \times \frac{1000 \times 999}{2} = 1,499,500$ 2 candidates are generated.
  - Multiples scans on the whole DB.
  - High memory consumption
- Discovery of long sequences is impossible:
  - an exponential number of candidates sub-sequences generated
  - for a sequence of length 100 : $2^{100} - 1 \approx 10^{30}$

Approaches "pattern growth"

- No candidate generation
- extraction of frequent items in projected DB.
- a greedy approach (in a depthfirst enumeration)

Prefix and suffix

- $\langle a \rangle$, $\langle aa \rangle$ and $\langle a(abc) \rangle$ are **prefix** of $\langle a(abc)(ac)d(cf) \rangle$

Prefix and suffix

- $\langle a \rangle$, $\langle aa \rangle$ and $\langle a(abc) \rangle$ are **prefix** of $\langle a(abc)(ac)d(cf) \rangle$
- Given $\langle a(abc)(ac)d(cf) \rangle$

Prefix and suffix

- $\langle a \rangle$, $\langle aa \rangle$ and $\langle a(abc) \rangle$ are **prefix** of $\langle a(abc)(ac)d(cf) \rangle$
- Given $\langle a(abc)(ac)d(cf) \rangle$

Préfixe	Suffixe (Prefix-Based Projection)
<a>	<(abc)(ac)d(cf)>
<aa>	<(_bc)(ac)d(cf)>
<ab>	<(_c)(ac)d(cf)>

- Some particular items : (_b)

Sequential pattern mining with prefix projections

- Step 1 : extraction of frequent 1-sequences:

$$\langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle, \langle f \rangle$$

- Step 2 : the complete set of frequent sequences can be partitioned into 6 sub-sets:
  - w.r.t. to prefix $\langle a \rangle$,
  - w.r.t. to prefix $\langle b \rangle$,
  - w.r.t. to prefix $\langle c \rangle$,
  - w.r.t. to prefix $\langle d \rangle$,
  - w.r.t. to prefix $\langle e \rangle$,
  - w.r.t. to prefix $\langle f \rangle$.

Finding sequences of prefix $\langle a \rangle$

Simply consider the projections according to $\langle a \rangle$:

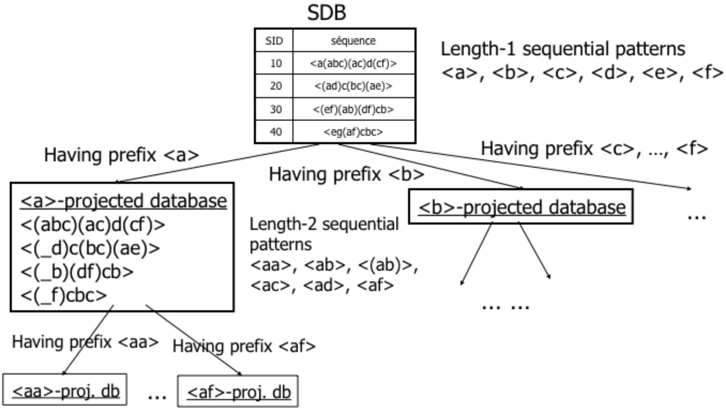
- $< (abc)(ac)d(cf) >$,
- $< (\_d)c(bc)(ae) >$,
- $< (\_b)(df)cb >$,
- $< (\_f)cbc >$

2-sequences of prefix $\langle a \rangle$:

- $< aa >$,
- $< ab >$,
- $< (ab) >$,
- $< ac >$,
- $< ad >$,
- $< af >$

new partition into 6 subsets

Completness of PrefixSpan

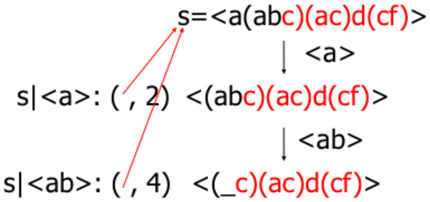


Efficiency PrefixSpan

- No candidate generation
- The size of projected DB decreases with the enumeration.
- The main cost of PrefixSpan: building the projected DB.
  - improved with **pseudo projections**

Pseudo projection

- If the DB can be load in main memory, use of pointers for the projections
- Pointers on the sequences
- Offset on the suffix



Closed Sequential Patterns

Definition ?

Closed Sequential Patterns

Definition ?

Motivations

- Redundancy
- Efficiency.

Closed Sequential Patterns

Definition ?

Motivations

- Redundancy
- Efficiency.

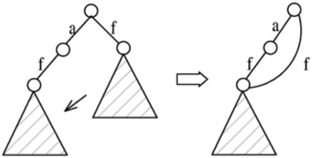
2 approaches:

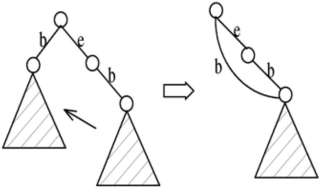
- CloSpan (Yan et al. 2003)
- Bide (Wang et al. 2007)

Clospan

Avoid to scan several times the same projected DB

"Using Backward Subpattern and Backward Superpattern pruning to prune redundant search space"





Algorithm

Limitations of managing a set of closed candidates

- post-processing : $O(n^2)$

Stop the management of candidates

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \ldots, s_g \rangle$:

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \ldots, s_g \rangle$:

1. forward inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \dots, s_g \rangle$:

1. forward inter itemset $S' = \langle s_1, s_2, \dots, s_g, \{e'\} \rangle$
2. forward intra itemset $S' = \langle s_1, s_2, \dots s_g \cup \{e'\} \rangle$

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \ldots, s_g \rangle$:

1. forward inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. forward intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. backward inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \ldots, s_g \rangle$:

1. forward inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. forward intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. backward inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$
4. backward intra itemset $S' = \langle s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g \rangle$

BIDE: Main idea

Extension of g-k-prefix sequence $\langle s_1, s_2, \ldots, s_g \rangle$:

1. forward inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. forward intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. backward inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$
4. backward intra itemset $S' = \langle s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g \rangle$

Closed sequence:

- If there is no extension that preserve the support of the sequence

Intervals of $\langle s_1, s_2, \ldots, s_g \rangle$

Exhibit items that occur in all i^{th} intervals

$$\underbrace{I_1}\, s_1, \underbrace{I_2}, s_2, \underbrace{I_3}, s_3, \ldots, s_{g-1} \underbrace{I_g}, s_g$$

Intervals of $\langle s_1, s_2, \dots, s_g \rangle$

Exhibit items that occur in all i^{th} intervals

$$\underbrace{}_{I_1} s_1, \underbrace{}_{I_2}, s_2, \underbrace{}_{I_3}, s_3, \dots, s_{g-1} \underbrace{}_{I_g}, s_g$$

Potentially several occurrences of a sequence within a data sequence

- Sequence $\langle (a, b)(a, c) \rangle$
- Data Sequence
  $\langle (a, b)(a, c)(a, b)(a, c)(a, b)(a, c)(a, b)(a, c) \rangle$

Intervals of $\langle s_1, s_2, \ldots, s_g \rangle$

Exhibit items that occur in all i^{th} intervals

$\underbrace{I_1}\, s_1, \underbrace{I_2}, s_2, \underbrace{I_3}, s_3, \ldots, s_{g-1} \underbrace{I_g}, s_g$

Potentially several occurrences of a sequence within a data sequence

- Sequence $\langle (a, b)(a, c) \rangle$
- Data Sequence $\langle (a, b)(a, c)(a, b)(a, c)(a, b)(a, c)(a, b)(a, c) \rangle$

Maximize these intervals

Intervals of $\langle s_1, s_2, \ldots, s_g \rangle$

Exhibit items that occur in all i^{th} intervals

$\underbrace{I_1}\, s_1, \underbrace{I_2}, s_2, \underbrace{I_3}, s_3, \ldots, s_{g-1} \underbrace{I_g}, s_g$

Potentially several occurrences of a sequence within a data sequence

- Sequence $\langle (a, b)(a, c) \rangle$
- Data Sequence $\langle (a, b)(a, c)(a, b)(a, c)(a, b)(a, c)(a, b)(a, c) \rangle$

Maximize these intervals

$I_1 : \langle (a, b)(a, c) \rangle$

$$\langle \underbrace{(a, b)(a, c)(a, b)(a, c)(a, b)(a, c)}(a, b)(a, c) \rangle$$

Intervals of $\langle s_1, s_2, \ldots, s_g \rangle$

Exhibit items that occur in all i^{th} intervals

$\underbrace{I_1} s_1, \underbrace{I_2}, s_2, \underbrace{I_3}, s_3, \ldots, s_{g-1} \underbrace{I_g}, s_g$

Potentially several occurrences of a sequence within a data sequence

- Sequence $\langle (a, b)(a, c) \rangle$
- Data Sequence $\langle (a, b)(a, c)(a, b)(a, c)(a, b)(a, c)(a, b)(a, c) \rangle$

Maximize these intervals

$I_2 : \langle (a, b)(a, c) \rangle$

$$\langle (a, b) \underbrace{(a, c)(a, b)(a, c)(a, b)(a, c)(a, b)} (a, c) \rangle$$

Constraints on sequences

Time constraints

- Window size,
- min gap,
- max gap

H Mannila, H Toivonen, AI Verkamo. Discovery of frequent episodes in event sequences. Data mining and knowledge discovery 1997.

Ramakrishnan Srikant, Rakesh Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. EDBT 1996.

M. Nanni and C. Rigotti. Extracting Trees of Quantitative Serial Episodes. KDID 2006.

Regular expressions

$\langle [a * a * bc * a] \rangle$

M. N. Garofalakis, R. Rastogi, and K. Shim. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. 1999.

Condensed representation

Much less condensed representation

- Closed patterns.
- Free/Generators.
- Non derivable pattern, impossible for data sequences.
  Raïssi et al, 2008.

Noise tolerant patterns: δ-free patterns.

More robust w.r.t. noise.

- the freeness is anti-monotone for itemset, not for sequences.

$\Rightarrow$ We have to define some introduce some other pruning properties.

P. Holat, M. Plantevit, C. Raïssi, N. Tomeh, T. Charnois, B. Crémilleux: Sequence Classification Based on Delta-Free Sequential Patterns. ICDM 2014

Outline

Graph Mining

In a graph collection

- Subgraph isomorphism test: NP Complete in the general case
- Canonical code base on DFS lexicographic order

X. Yan and J. Han. gSpan: graph-based substructure pattern mining. ICDM 2003.

Graph Mining

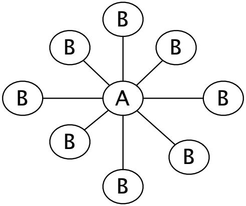
In a graph collection

- Subgraph isomorphism test: NP Complete in the general case
- Canonical code base on DFS lexicographic order

X. Yan and J. Han. gSpan: graph-based substructure pattern mining. ICDM 2003.

In a single graph

The usual definition of support is not anti-monotone:



T. Calders, J. Ramon, D. Van Dyck: Anti-monotonic Overlap-Graph Support Measures. ICDM 2008.

B. Bringmann, S. Nijssen: What Is Frequent in a Single Graph?. PAKDD 2008

Dense subgraph mining

Mining clique: cliqueness is antimonotone $\Rightarrow$ Just enumerate the nodes taking advantage of AM property.

Dense subgraph mining

Mining clique: cliqueness is antimonotone $\Rightarrow$ Just enumerate the nodes taking advantage of AM property.

What about quasi-clique mining?

Dense subgraph mining

Mining clique: cliqueness is antimonotone $\Rightarrow$ Just enumerate the nodes taking advantage of AM property.

What about quasi-clique mining?

Pb1

Let $\gamma \in]0,1]$, $C \subseteq V$ is a γ-quasi-clique if $\forall v \in C, deg(v, G[C]) \geq \gamma(|C|-1)$ where $deg(v, G[C])$ is the degree of v in $G[C]$

Guimei Liu, Limsoon Wong: Effective Pruning Techniques for Mining Quasi-Cliques. ECML/PKDD 2008

Pb2

Let $\gamma \in]0,1]$, $C \subseteq V$ is a pseudo-clique if $\frac{2\times|E[C]|}{|C|\times(|C|-1)} \geq \gamma$.

Takeaki Uno: An Efficient Algorithm for Solving Pseudo Clique Enumeration Problem. Algorithmica 2010

Dense subgraph mining

Mining clique: cliqueness is antimonotone $\Rightarrow$ Just enumerate the nodes taking advantage of AM property.

What about quasi-clique mining?

Pb1

Let $\gamma \in]0,1]$, $C \subseteq V$ is a γ-quasi-clique if $\forall v \in C, deg(v, G[C]) \geq \gamma(|C|-1)$ where $deg(v, G[C])$ is the degree of v in $G[C]$

Guimei Liu, Limsoon Wong: Effective Pruning Techniques for Mining Quasi-Cliques. ECML/PKDD 2008

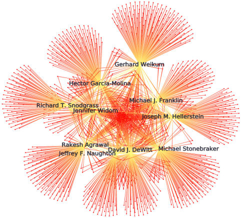
Pb2

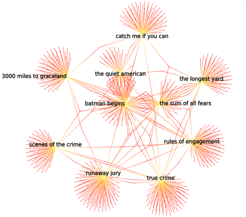
Let $\gamma \in]0,1]$, $C \subseteq V$ is a pseudo-clique if $\frac{2\times|E[C]|}{|C|\times(|C|-1)} \geq \gamma$.

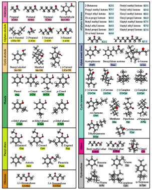
Takeaki Uno: An Efficient Algorithm for Solving Pseudo Clique Enumeration Problem. Algorithmica 2010

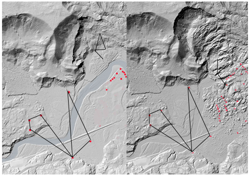
Question: Which Pb is the most difficult? Why?

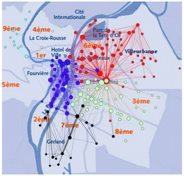
From Data to Augmented Graphs











- Graphs are often **dynamic** with **attributes** related to **vertices** and/or **edges**.

Mining Augmented Graphs

Analyzing large augmented graphs leads to many challenges:

- Working with network data is messy
  - Not just "wiring diagrams" but also dynamics and data (features, attributes) on nodes and edges
  - Computational issues
- Expressivity et genericity: to answer to questions from
  - Social sciences, Physics, Biology, Neurosciences, etc.

☛ How network structure and node attribute values relate and influence each other?

Mining Augmented Graphs

Analyzing large augmented graphs leads to many challenges:

- Working with network data is messy
  - Not just "wiring diagrams" but also dynamics and data (features, attributes) on nodes and edges
  - Computational issues
- Expressivity et genericity: to answer to questions from
  - Social sciences, Physics, Biology, Neurosciences, etc.

How network structure and node attribute values relate and influence each other?

Constraint-based pattern mining and the IDB framework

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{\varphi \in \mathcal{L} \mid \mathcal{C}(\varphi, \mathcal{D}) \text{ is true } \}$$

- $\mathcal{L}$: multiples pattern domains are possible
- $\mathcal{D}$: one or several graphs
- $\mathcal{C}$: (quasi)-clique, homogeneity, diameter, etc.

Boolean Attributed-Node Graph

- Attribute + Structure → Mining homogeneous dense subgraphs.
  F. Moser, R. Colak, A. Rafiey, M. Ester: Mining Cohesive Patterns from Graphs with Feature Vectors. SDM 2009

Boolean Attributed-Node Graph

- Attribute + Structure → Mining homogeneous dense subgraphs.

  F. Moser, R. Colak, A. Rafiey, M. Ester: Mining Cohesive Patterns from Graphs with Feature Vectors. SDM 2009

- Attribute + Structure → Mining homogeneous *collections* of dense subgraphs.

  P-N Mougel, C. Rigotti, M. Plantevit, O. Gandrillon: Finding maximal homogeneous clique sets. Knowl. Inf. Syst. 39(3), 2014

  P-N Mougel, C. Rigotti, O. Gandrillon Finding Collections of k-Clique Percolated Components in Attributed Graphs. PAKDD 2012

Boolean Attributed-Node Graph

- Attribute + Structure → Mining homogeneous dense subgraphs.

  F. Moser, R. Colak, A. Rafiey, M. Ester: Mining Cohesive Patterns from Graphs with Feature Vectors. SDM 2009
- Attribute + Structure → Mining homogeneous *collections* of dense subgraphs.

  P-N Mougel, C. Rigotti, M. Plantevit, O. Gandrillon: Finding maximal homogeneous clique sets. Knowl. Inf. Syst. 39(3), 2014

  P-N Mougel, C. Rigotti, O. Gandrillon Finding Collections of k-Clique Percolated Components in Attributed Graphs. PAKDD 2012
- Structural Correlation Pattern Mining:
  - Structural correlation: Probability of a vertex that has an attribute set S to be part of a correlated dense subgraph Q
  - Structural correlation pattern (S, Q): Correlated dense subgraph Q wrt S.

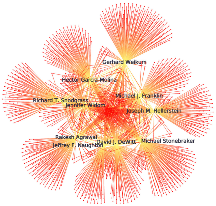
  A. Silva, W. Meira Jr., M. J. Zaki: Mining Attribute-structure Correlated Patterns in Large Attributed Graphs. PVLDB (2012)

Numeric Attributed-Node Graph: Topological Patterns

What are the node attributes that strongly co-vary with the graph structure

- $P = \{PVLDB^+, Betw^+\}$ $(Gr(P, E) \simeq 7)$
- Kendall's Tau Generalization:

$$Supp_\tau(P) = \frac{|\{(u,v) \in V^2 \mid \forall A^s \in P : A(u) \rhd_s A(v)\}|}{\binom{|V|}{2}}$$



if $s = +$, $\rhd_s$ is $<$, otherwise $>$.

- Computing $Supp_\tau$: $O(|V|^2)$
  - Computing a tight upper bound: $O(|V|)$
  - Index structure

A. Prado, M. Plantevit, C. Robardet, J-F. Boulicaut: Mining Graph Topological Patterns: Finding Covariations among Vertex Descriptors. IEEE Trans. Knowl. Data Eng. 25(9): 2090-2104 (2013)

Topological Patterns (2/2)

- For a centrality measure, what are the most impacting conferences?

Rank	DEG$^+$		BETWEEN$^+$	
	Publication	Factor	Publication	Factor
1	ECML/PKDD$^+$	2.5	PVLDB$^+$	5.67
2	IEEE TKDE$^+$	2.28	EDBT$^+$	5.11
3	PAKDD$^+$	2.21	VLDB J.$^+$	4.35
4	DASFAA$^+$	2.09	SIGMOD$^+$	4.25
5	ICDM$^+$	1.95	ICDE$^+$	3.42

- What are the most representative authors?

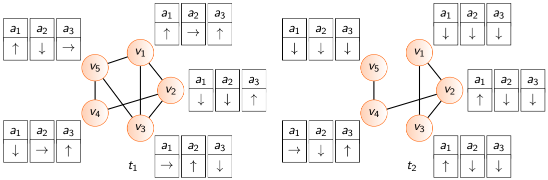
PRK$^+$ DEG$^+$ECML/PKDD$^+$	PRK$^+$ BETWEEN$^+$PVLDB$^+$
Christos Faloutsos	Gerhard Weikum
Jiawei Han	Jiawei Han
Philip S. Yu	David Maier
Bing Liu	Philip S. Yu
C. Lee Giles	Hector Garcia-Molina

Dynamic Attributed Graphs

A dynamic attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$ is a sequence over $\mathcal{T}$ of attributed graphs $G_t = (\mathcal{V}, E_t, A_t)$, where:

- $\mathcal{V}$ is a set of vertices that is fixed throughout the time,
- $E_t \in \mathcal{V} \times \mathcal{V}$ is a set of edges at time t,
- A_t is a vector of numerical values for the attributes of $\mathcal{A}$ that depends on t.

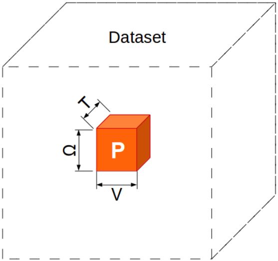
Example



Co-evolution Pattern

Given $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$, a co-evolution pattern is a triplet $P = (V, T, \Omega)$ s.t.:

- $V \subseteq \mathcal{V}$ is a subset of the vertices of the graph.
- $T \subset \mathcal{T}$ is a subset of not necessarily consecutive timestamps.
- Ω is a set of signed attributes, i.e., $\Omega \subseteq A \times S$ with $A \subseteq \mathcal{A}$ and $S = \{+, -\}$ meaning respectively a $\{\textit{increasing}, \textit{decreasing}\}$ trend.



Predicates

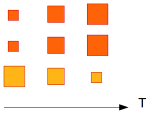
A co-evolution pattern must satisfy two types of constraints:

Constraint on the evolution:

- Makes sure attribute values co-evolve

☛ δ-**strictEvol**.

- $\forall v \in V$, $\forall t \in T$ and $\forall a^s \in \Omega$ then δ-$trend(v, t, a) = s$

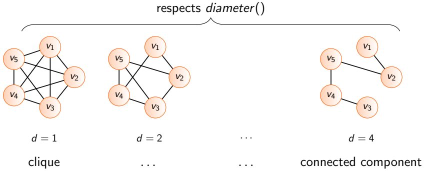


Constraint on the graph structure:

- Makes sure vertices are related through the graph structure.

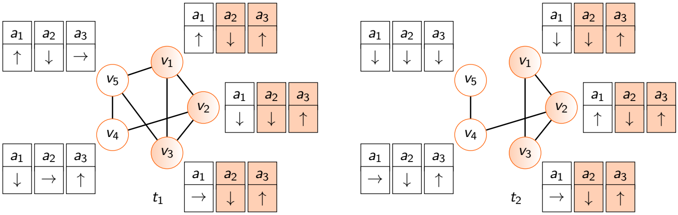
☛ **diameter**.

- Δ-$diameter\big(V, T, \Omega\big) = true \Leftrightarrow \forall t \in T \; diam_{G_t(V)} \leq \Delta$

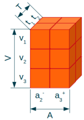


Example

$$P = \{(v_1, v_2, v_3)(t_1, t_2)(a_2^-, a_3^+)\}$$



- 1-Diameter(P) is true,
- 0-strictEvol(P) is true.

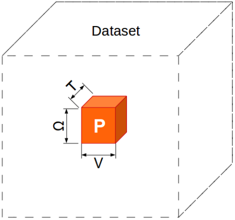


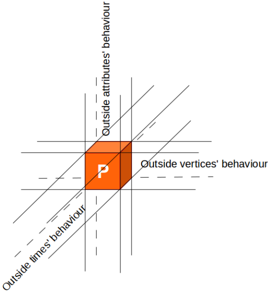
Density Measures

Intuition

Discard patterns that depict a behaviour supported by many other elements of the graph.

☛ **vertex specificity**, **temporal dynamic** and **trend relevancy**.

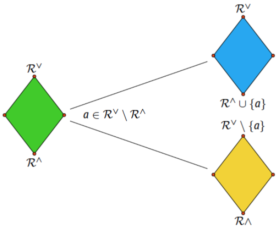




Algorithm

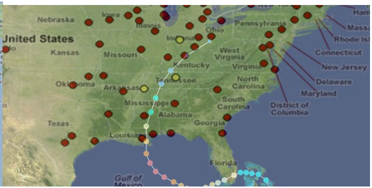
How to use the properties of the constraints to reduce the search space?

- Binary enumeration of the search space.
- Using the properties of the constraints to reduce the search space
  - Monotone, anti-monotone, piecewise (anti-)monotone, etc.
- Constraints are fully or partially pushed:
  - to prune the search space (i.e., stop the enumeration of a node),
  - to propagate among the candidates.



Cerf et al, ACM TKDD 2009

This algorithms aim to be complete but other heuristic search can be used in a straightforward way (e.g., beam-search) to be more scalable



Top temporal_dynamic trend dynamic sub-graph (in red)

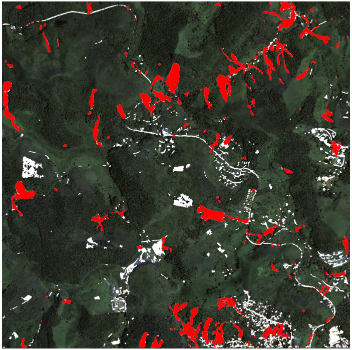
- 71 airports whose arrival delays increase over 3 weeks.
- *temporal_dynamic* $= 0$, which means that arrival delays never increased in these airports during another week.
- The hurricane strongly influenced the domestic flight organization.

Top trend_relevancy (Yellow)

- 5 airports whose number of departures and arrivals increased over the three weeks following Katrina hurricane.
- *trend_relevancy* value equal to 0.81
- Substitutions flights were provided from these airports during this period.
- This behavior is rather rare in the rest of the graph

	$\|V\|$	$\|T\|$	$\|A\|$	density
Katrina	280	8	8	5×10^{-2}

Brazil landslides

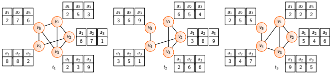


	$\|V\|$	$\|T\|$	$\|A\|$	density
Brazil landslide	10521	2	9	0.00057

Discovering lanslides

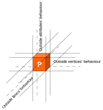
- Taking into account expert knowledge, focus on the patterns that involve NDVI$^+$.
- Regions involved in the patterns: true landslides (red) and other phenomena (white).
- Compare to previous work, much less patterns to characterize the same phenomena (4821 patterns vs millions).

Overview



Co-evolution patterns

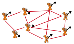
Interestingness Measures



(Desmier et al., ECML/PKDD 2013)

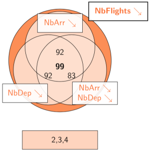
Experimental results

DBLP US flights Brazil landslides

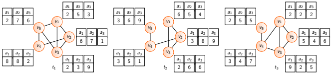


- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized

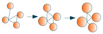
Desmier et al, IDA 2014



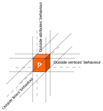
Overview



Co-evolution patterns



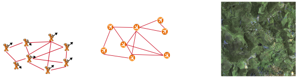
Interestingness Measures



(Desmier et al., ECML/PKDD 2013)

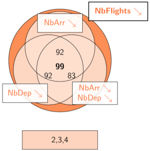
Experimental results

DBLP US flights Brazil landslides



- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized

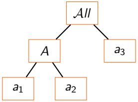
Desmier et al, IDA 2014



Hier. co-evolution patterns

Take benefits from a hierarchy over the vertex attributes to :

- return a more concise collection of patterns;
- discover new hidden patterns;



Issue

☛ We need to mine *contextualized* trajectories.

- What about the data?

Contexts	✓
Trajectories	only 2 points

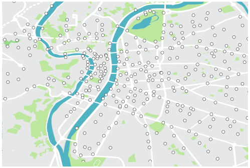
- How to have a good view of the demographic flows with only 2-point-trajectories?

Our idea:

☛ Taking benefit from the **crowd** with an attributed graph based approach.

- Individual trajectories are aggregated into weighted graphs;
- We look for *exceptional sub-graph*

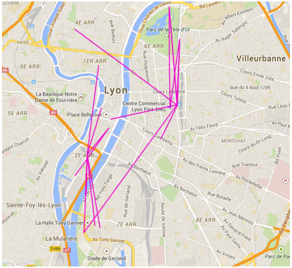
Example: The Velo'v network[1]



- 348 Velov stations across the city of Lyon.
- The dataset contains movement data collected in a 2 year period (Jan. 2011– Dec. 2012)
- Each movement (edge) includes both bicycle stations (vertices) and timestamps for departure and arrival, as well as some basic demographics about the user of the bike (context).
- Customers described by nominal attributes (gender, type of membership card, ZIP code and country of residence) and a numerical one (year of birth).
- $50,601$ customers.
- $2,000,000$ contextualized edges in total.

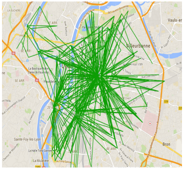
[1] http://www.velov.grandlyon.com/

Examples of Demographic and contextualized Specific Routes



$\text{YoB} \geq 1968, \text{ZIP} = 42400$

- identifies people born after 1968, living in a city (Saint Chamond) located approximately 50km from Lyon.
- the edges involve the **two main train stations of Lyon**: Perrache (south-west) and Part-Dieu (center), from which users take bicycles to areas that are not easily reached by metro or tram, such as the 1^{st} and 4^{th} districts.



$\text{YoB} \geq 1962, \text{CAT} = \text{OURA}$

Pb Formalization: Key concepts

A **context** aims to characterize a subset of movements/trajectories.

Aggregate graph G_C

- Given a context C, G_C is a weighted graph involving all edges that satisfy C.
- The weight of an edge is the number of movements involving the two vertices that hold for C.

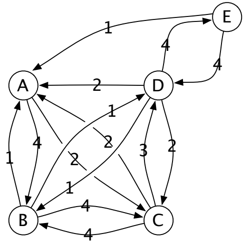
Operations on G_C

Differential comparison with G_*:

- Adequacy of an edge to a context assessed by a χ^2 test.
- Some quality measures to "quantify" the attraction of the edges for a context: $q(e, C)$.

Example

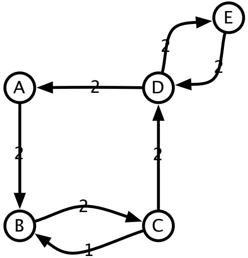
	Contexts			Trajectories
User	Gender	Age	Time	Travels
u_1	F	20	Day	(A,C), (B,A), (C,B)
u_1	F	20	Night	(D,C),(D,E),(E,A),(E,D)
u_2	M	23	Day	(A,B),(B,C),(C,A),(C,B)
u_2	M	23	Night	(A,B),(B,C),(C,B)(C,D),(D,C),(D,E),(E,D)
u_3	F	45	Day	(A,B),(B,C),(C,D),(D,A),(D,E),(E,D)
u_3	F	45	Night	(B,D),(D,B)
u_4	M	50	Day	(A,B),(B,C),(C,B),(C,D),(D,A),(D,E),(E,D)
u_4	M	50	Night	(A,C),(C,A)



$G_\star$

Example

	Contexts			Trajectories
User	Gender	Age	Time	Travels
u_1	F	20	Day	(A,C), (B,A), (C,B)
u_1	F	20	Night	(D,C),(D,E),(E,A), (E,D)
u_2	M	23	Day	(A,B),(B,C),(C,A), (C,B)
u_2	M	23	Night	(A,B),(B,C),(C,B) (C,D),(D,C),(D,E), (E,D)
u_3	F	45	Day	(A,B),(B,C),(C,D), (D,A),(D,E),(E,D)
u_3	F	45	Night	(B,D),(D,B)
u_4	M	50	Day	(A,B),(B,C),(C,B), (C,D),(D,A),(D,E), (E,D)
u_4	M	50	Night	(A,C),(C,A)



C= $(Gender = \star, Age \in [45, 50], Time = Day)$

Demographic and Contextualized Specific Route pattern

A pair (C, G') where

- C is a context
- G' is a subgraph of G_C such that:
  - $\forall e \in G'$, e fulfils the χ^2 test and $q(e, C) > 0$,
  - G' is connected.

The Mining Task

- No threshold to avoid related issues.
- ☛Some measures to be maximized by the patterns:
  - density of G', #edges, #vertices, several aggregations of the quality measure.

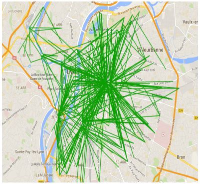
Mining Task:

Given a set of measures (user-preferences) M, our goal is to compute the Pareto-front of the Demographic and Contextualized Specific Route patterns according to M.

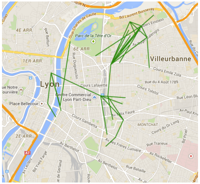
Algorithm in a nutshell

- Enumeration of the possible contexts in a depth-first fashion.
- Several upper-bounds to early prune unpromising candidates:
  - on the χ^2 for each edge (see Sese and Morishita, PKDD'04)
  - on the other measures;

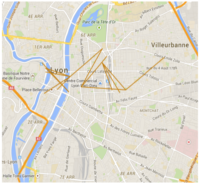
XP



(i) YoB ≥ 1962, CAT $=$ OURA



(ii) YoB ≥ 1980, TYP $=$ standard



(iii) YoB ≥ 1992, ZIP $=$ 69003

i The edges of pattern (i) radiate from all of Lyon's train stations, not only the major ones. Its description refers to holders of a regional train subscription (monthly or yearly).

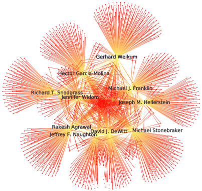
ii It involves users born in or after 1980:

- 3 main areas: the scientific campus in the north, the Presqu'île and its pubs, and the shopping area in the center of Lyon.

iii Young people that live in the 3rd district use bicycles to move around in their area.

- ground truth in real-world data: the ZIP code of users aligns with the area where the bicycles are used!

Some other inductive queries for augmented graphs

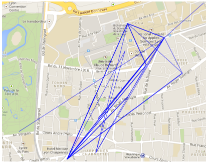


- What are the node attributes that strongly co-vary with the graph structure?
  - Co-authors that published at ICDE with a high degree and a low clustering coefficient.

  Prado et al., IEEE TKDE 2013

- Which are the node attribute temporal combination that impact the graph structure ?
  - *dynamic attributed graph*

  M. Kaytoue et al. Social Netw. Analys. Mining (2015)



- For a given population, what is the most related subgraphs (i.e., behavior)? For a given subgraph, which is the most related subpopulation?
  - *edge-attributed graph*
  - People born after 1979 are over represented on the campus.

Outline

Conclusion

$$Th(\mathcal{L}, \mathcal{D}, \mathcal{C}) = \{\psi \in \mathcal{L} \mid \mathcal{C}(\psi, \mathcal{D}) \text{ is true}\}$$

- Pattern domains: (itemset, sequences, graphs, dynamic graphs, etc.)
- Constraints: How to efficiently push them?

Research Avenues

- Still new pattern domains and and their related primitives have to be defined.
- Accept to lose the completeness in some cases (next course).
- Integration of domain knowledge.
- Interactivity: replace the user in the center of the KDD process.
  - User preference learning
  - Inductive query recommendation
- Describing a phenomena with the simplest pattern language!

Thanks

End