Constraint-Based Pattern Mining

From classic pattern domains to more sophisticated ones

Marc Plantevit

ENS Lyon, March 2016
Université Claude Bernard Lyon 1 – LIRIS CNRS UMR5205

* Slides from the éEGC lecture.

About me.

marc.plantevit@univ-lyon1.fr or marc.plantevit@liris.cnrs.fr
Associate Professor
Computer Science Dept.
University Claude Bernard Lyon 1.

Lab: **LIRIS UMR 5205**

Team: **Data Mining & Machine Learning**

Interest: **Fondations of constraint-based pattern mining, sequences, augmented graphs.**

Before: **Ph.D from University Montpellier II (LIRMM),**
Post-Doc at Univ. Caen (GREYC).

Outline

Evolution of Sciences

Before 1600: Empirical Science

- Babylonian mathematics: 4 basis operations done with tablets and the resolution of practical problems based on words describing all the steps. $\Rightarrow$ that worked and they manage to solve 3 degree equations.
- Ancient Egypt: No theorization of algorithms. We give only examples made empirically, certainly repeated by students and scribes. Empirical knowledge, transmitted as such, and not a rational mathematical science.
- Aristotle also produced many biological writings that were empirical in nature, focusing on biological causation and the diversity of life. He made countless observations of nature, especially the habits and attributes of plants and animals in the world around him, classified more than 540 animal species, and dissected at least 50.
- ...

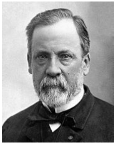
Wikipedia

1600-1950s: Theoretical Science

Each discipline has grown a theoretical component. Theoretical models often motivate experiments and generalize our understanding.

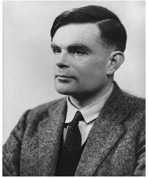
- Physics: Newton, Max Planck, Albert Einstein, Niels Bohr, Schrödinger
- Mathematics: Blaise Pascal, Newton, Leibniz, Laplace, Cauchy, Galois, Gauss, Riemann
- Chemistry: R. Boyle, Lavoisier, Dalton, Mendeleev,
- Biology, Medecine, Genetics: Darwin, Mendel, Pasteur





1950s–1990s, Computational Science

- Over the last 50 years, most disciplines have grown a third, computational branch (e.g. empirical, theoretical, and computational ecology, or physics, or linguistics.)
- Computational Science traditionally meant simulation. It grew out of our inability to find closed form solutions for complex mathematical models.



The Data Science Era

1990's-now, Data Science

- The flood of data from new scientific instruments and simulations
- The ability to economically store and manage petabytes of data online
- The Internet and computing Grid that makes all these archives universally accessible
- Scientific info. management, acquisition, organization, query, and visualization tasks scale almost linearly with data volumes.

The Fourth Paradigm: Data-Intensive Scientific Discovery

Data mining is a major new challenge!

The Fourth Paradigm. Tony Hey, Stewart Tansley, and Kristin Tolle. Microsoft Research, 2009.

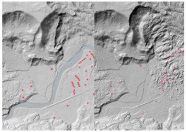
Evolution of Database Technology

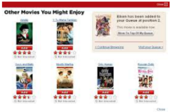
- 1960s: Data collection, database creation, IMS and network DBMS
- 1970s : Relational data model, relational DBMS implementation
- 1980s: RDBMS, advanced data models (extended-relational, OO, deductive, etc.), application-oriented DBMS (spatial, scientific, engineering, etc.)
- 1990s: Data mining, data warehousing, multimedia databases, and Web databases
- 2000s: Stream data management and mining, Data mining and its applications, Web technology (XML, data integration) and global information systems, NoSQL, NewSQL.

Why Data Mining?

- The Explosive Growth of Data: from terabytes to petabytes
  - Data collection and data availability
  - Automated data collection tools, database systems, Web, computerized society
- Major sources of abundant data
  - Business: Web, e-commerce, transactions, stocks, ...
  - Science: Remote sensing, bioinformatics, scientific simulation,...
  - Society and everyone: news, digital cameras, social network, ...
  - “We are drowning in data, but starving for knowledge!” – John Naisbitt, 1982 –

Applications



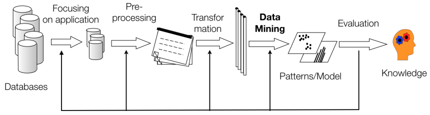


- Human mobility (ANR VEL'INNOV 2012–2016)
- Social media (GRAISearch - FP7-PEOPLE-2013-IAPP, Labex IMU project RESALI 2015–2018)
- Soil erosion (ANR Foster 2011–2015)
- Neuroscience (olfaction)
- Chemoinformatics
- Fact checking (ANR ContentCheck 2016 – 2019)
- Industry (new generation of product, failure detection)
- ...

What is Data Mining

- Data mining (knowledge discovery from data)
    - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data
- Alternative names:
    - KDD, knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- Watch out: Is everything “data mining”?
    - simple search or query processing
    - (Deductive) expert systems

KDD Process



Iterative and Interactive Process

Fayad et al., 1996

Data Mining

- Core of KDD
- Search for knowledge in data

Functionalities

- **Descriptive data mining** vs Predictive data mining
- **Pattern mining**, classification, clustering, regression
- Characterization, discrimination, association, classification, clustering, outlier and trend analysis, etc.

Major Issues In Data Mining

- Mining methodology
  - Mining different kinds of knowledge from diverse data types, e.g., bio, stream, Web.
  - Performance: efficiency, effectiveness, and scalability
  - Pattern evaluation: the interestingness problem
  - Incorporation of background knowledge.
  - Handling noise and incomplete data
  - Parallel, distributed and incremental mining methods.
  - Integration of the discovered knowledge with existing one: knowledge fusion.
  - Completeness or not.
- User interaction
  - Data mining query languages and ad-hoc mining.
  - Expression and visualization of data mining results.
  - Interactive mining of knowledge at multiple levels of abstraction
- Applications and social impacts
  - Domain-specific data mining & invisible data mining
  - Protection of data security, integrity, and privacy.

Where to Find References? DBLP, Google Scholar

- Data Mining and KDD
  - Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
  - Journals: Data Mining and Knowledge Discovery, ACM TKDD
- Database Systems
  - Conferences: : ACM-SIGMOD, ACM-PODS, (P)VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
  - Journals: IEEE-TKDE, ACM-TODS/TOIS, JIIS, J. ACM, VLDB J., Info. Sys., etc.
- AI & Machine Learning
  - Conferences: Int. Conf. on Machine learning (ICML), AAAI, IJCAI, COLT (Learning Theory), CVPR, NIPS, etc
  - Journals: Machine Learning, Artificial Intelligence, Knowledge and Information Systems, IEEE-PAMI, etc.
- Web and IR
  - Conferences: SIGIR, WWW, CIKM, etc
  - Journals: WWW: Internet and Web Information Systems,

Recommended Books

- U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996
- J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2nd ed., 2006
- D. J. Hand, H. Mannila, and P. Smyth, Principles of Data Mining, MIT Press, 2001
- P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Wiley, 2005
- Charu C. Aggarwal, Data Mining, Springer, 2015.
- Mohammed J. Zaki, Wagner Meira, Jr. Data Mining and Analysis Fundamental Concepts and Algorithms. Cambridge University Press, 2014.

Roadmap

We will focus on **descriptive data mining** especially on Constraint-based Pattern Mining with an **inductive database vision**.

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

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

Imielinski and Mannila: Communications of the ACM (1996).

Outline

Itemset: definition

Definition

Given a set of attributes $\mathcal{A}$, an *itemset* X is a subset of attributes, i. e., $X \subseteq \mathcal{A}$.

Input:

$$\begin{array}{c|cccc} & a_1 & a_2 & \dots & a_n \\ \hline o_1 & d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ o_2 & d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_m & d_{m,1} & d_{m,2} & \dots & d_{m,n} \end{array}$$

where $d_{i,j} \in \{\text{true,false}\}$

Question

How many itemsets are there?

Itemset: definition

Definition

Given a set of attributes $\mathcal{A}$, an *itemset* X is a subset of attributes, i. e., $X \subseteq \mathcal{A}$.

Input:

$$\begin{array}{c|cccc} & a_1 & a_2 & \dots & a_n \\ \hline o_1 & d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ o_2 & d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_m & d_{m,1} & d_{m,2} & \dots & d_{m,n} \end{array}$$

where $d_{i,j} \in \{\text{true,false}\}$

Question

How many itemsets are there? $2^{|\mathcal{A}|}$.

Transactional representation of the data

Relational representation: $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

$$
\begin{array}{c|cccc}
 & a_1 & a_2 & \dots & a_n \\
\hline
o_1 & d_{1,1} & d_{1,2} & \dots & d_{1,n} \\
o_2 & d_{2,1} & d_{2,2} & \dots & d_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
o_m & d_{m,1} & d_{m,2} & \dots & d_{m,n}
\end{array}
$$

where $d_{i,j} \in \{\text{true,false}\}$

Transactional representation: $\mathcal{D}$ is an array of subsets of $\mathcal{A}$

$$
\begin{array}{c}
t_1 \\
t_2 \\
\vdots \\
t_m
\end{array}
$$

where $t_i \subseteq \mathcal{A}$

Example

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

t_1	a_1, a_2, a_3
t_2	a_1, a_2
t_3	a_2
t_4	a_3

Frequency: definition

Definition (absolute frequency)

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, the absolute *frequency* of an itemset $X \subseteq \mathcal{A}$ in the dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ is $|\{o \in \mathcal{O} \mid \{o\} \times X \subseteq \mathcal{D}\}|$.

Definition (relative frequency)

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, the relative *frequency* of an itemset $X \subseteq \mathcal{A}$ in the dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ is $\frac{|\{o \in \mathcal{O} \mid \{o\} \times X \subseteq \mathcal{D}\}|}{|\mathcal{O}|}$.

The relative frequency is a joint probability.

Frequent itemset mining

Problem Definition

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, listing every itemset having a frequency above a given threshold $\mu \in \mathbb{N}$.

Input:

$$\begin{array}{c|cccc} & a_1 & a_2 & \dots & a_n \\ \hline o_1 & d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ o_2 & d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_m & d_{m,1} & d_{m,2} & \dots & d_{m,n} \end{array}$$

and a minimal frequency $\mu \in \mathbb{N}$.

where $d_{i,j} \in \{\text{true,false}\}$

R. Agrawal; T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in Large Databases, SIGMOD, 1993.

Frequent itemset mining

Problem Definition

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, listing every itemset having a frequency above a given threshold $\mu \in \mathbb{N}$.

Output: *every* $X \subseteq \mathcal{A}$ such that there are at least μ objects having all attributes in X.

R. Agrawal; T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in Large Databases, SIGMOD, 1993.

Frequent itemset mining: illustration

Specifying a minimal absolute frequency $\mu = 2$ objects (or, equivalently, a minimal relative frequency of 50%).

	a_1	a_2	a_3
o_1	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

Frequent itemset mining: illustration

Specifying a minimal absolute frequency $\mu = 2$ objects (or, equivalently, a minimal relative frequency of 50%).

	a_1	a_2	a_3
o_1	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

The frequent itemsets are: $\emptyset$ (4), $\{a_1\}$ (2), $\{a_2\}$ (3), $\{a_3\}$ (2) and $\{a_1, a_2\}$ (2).

Completeness

Both the clustering and the classification schemes *globally* model the data: every object influences the output. That is the fundamental reason for these tasks to be solved in an *approximate* way.

In contrast, *local* patterns, such as itemsets, describe "anomalies" in the data and all such anomalies usually can be *completely* listed.

Inductive database vision

Querying data:

$$\{d \in \mathcal{D} \mid q(d, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a dataset (tuples),
- q is a query.

Inductive database vision

Querying patterns:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is the dataset,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is the dataset,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a subset of $\mathcal{O} \times \mathcal{A}$, i. e., objects described with Boolean attributes,
- P is the pattern space,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a subset of $\mathcal{O} \times \mathcal{A}$, i. e., objects described with Boolean attributes,
- P is $2^{\mathcal{A}}$,
- $\mathcal{Q}$ is an inductive query.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a subset of $\mathcal{O} \times \mathcal{A}$, i. e., objects described with Boolean attributes,
- P is $2^{\mathcal{A}}$,
- $\mathcal{Q}$ is $(X, \mathcal{D}) \mapsto |\{o \in \mathcal{O} \mid \{o\} \times X \subseteq \mathcal{D}\}| \geq \mu$.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a subset of $\mathcal{O} \times \mathcal{A}$, i. e., objects described with Boolean attributes,
- P is $2^{\mathcal{A}}$,
- $\mathcal{Q}$ is $(X, \mathcal{D}) \mapsto f(X, \mathcal{D}) \geq \mu$.

Inductive database vision

Querying the frequent itemsets:

$$\{X \in P \mid \mathcal{Q}(X, \mathcal{D})\}$$

where:

- $\mathcal{D}$ is a subset of $\mathcal{O} \times \mathcal{A}$, i. e., objects described with Boolean attributes,
- P is $2^{\mathcal{A}}$,
- $\mathcal{Q}$ is $(X, \mathcal{D}) \mapsto f(X, \mathcal{D}) \geq \mu$.

Listing the frequent itemsets is NP-hard.

Naive algorithm

Input: $\mathcal{O}, \mathcal{A}, \mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}, \mu \in \mathbb{N}$
Output: $\{X \subseteq \mathcal{A} \mid f(X, \mathcal{D}) \geq \mu\}$
for all $X \subseteq \mathcal{A}$ **do**
  if $f(X, \mathcal{D}) \geq \mu$ **then**
    output(X)
  end if
end for

Question

How many itemsets are enumerated? $2^{|\mathcal{A}|}$.

Transactional representation of the data

Relational representation:
$\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

	a_1	a_2	$\dots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\dots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\dots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\dots$	$d_{m,n}$

where $d_{i,j} \in \{\text{true,false}\}$

Transactional representation: $\mathcal{D}$ is an array of subsets of $\mathcal{A}$

$$\begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_m \end{array}$$

where $t_i \subseteq \mathcal{A}$

Transactional representation of the data

Relational representation: $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

	a_1	a_2	$\ldots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\ldots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\ldots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\ldots$	$d_{m,n}$

where $d_{i,j} \in \{\text{true,false}\}$

Transactional representation: $\mathcal{D}$ is an array of subsets of $\mathcal{A}$

$$\begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_m \end{array}$$

where $t_i \subseteq \mathcal{A}$

For a linear time verification of "X being a subset of t_i", the transactions are sorted (arbitrary order on $\mathcal{A}$) in a pre-processing step and any enumerated itemset X respects this order.

Transactional representation of the data

Relational representation:
$\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

	a_1	a_2	$\ldots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\ldots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\ldots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\ldots$	$d_{m,n}$

where $d_{i,j} \in \{\text{true,false}\}$

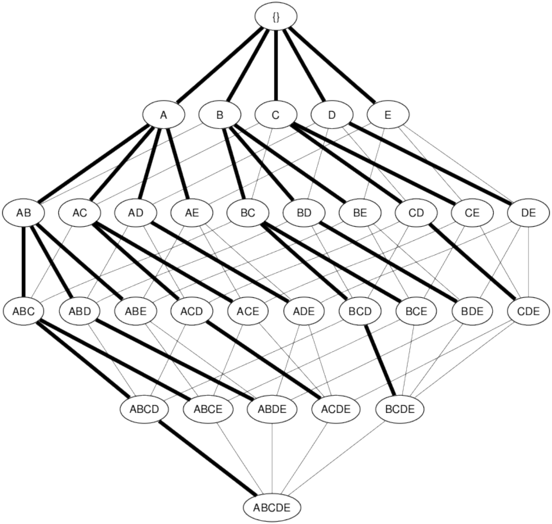
Transactional representation: $\mathcal{D}$ is an array of subsets of $\mathcal{A}$

$$\begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_m \end{array}$$

where $t_i \subseteq \mathcal{A}$

For a linear time verification of "X being a subset of t_i", the transactions are sorted (arbitrary order on $\mathcal{A}$) in a pre-processing step and any enumerated itemset X respects this order.

Prefix-based enumeration



Complexity of the naive approach

Question

How many itemsets are enumerated? $2^{|\mathcal{A}|}$.

Question

What is the worst-case complexity of computing $f(X, \mathcal{D})$?

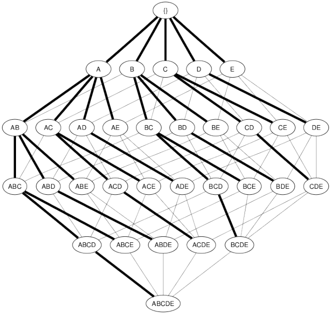
Question

What is the worst-case complexity of computing $f(X, \mathcal{D})$? $O(|\mathcal{O} \times \mathcal{A}|)$.

Question

What is the worst-case complexity of the naive approach? $O(2^{|\mathcal{A}|}|\mathcal{O} \times \mathcal{A}|)$.

How to efficiently mine frequent itemsets?



Taking advantage of an important property

- *Anti-monotonicity of the frequency*
- in a levelwise enumeration (e.g. Apriori)
  - R. Agrawal; T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in Large Databases, SIGMOD, 1993.
- in a depthfirst enumeration (e.g. Eclat)
  - Mohammed J. Zaki, Scalable Algorithms for Association Mining. IEEE TKDE, 2000.

Anti-monotonicity of the frequency

Theorem

Given a dataset $\mathcal{D}$ of objects described with Boolean attributes in $\mathcal{A}$:

$$\forall (X, Y) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}, X \subseteq Y \Rightarrow f(X, \mathcal{D}) \geq f(Y, \mathcal{D}) \ .$$

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

$$\begin{aligned} f(\emptyset, \mathcal{D}) &= 4 \\ f(\{a_1\}, \mathcal{D}) &= 2 \\ f(\{a_1, a_2\}, \mathcal{D}) &= 2 \\ f(\{a_1, a_2, a_3\}, \mathcal{D}) &= 1 \end{aligned}$$

Anti-monotonicity of the frequency

Theorem

Given a dataset $\mathcal{D}$ of objects described with Boolean attributes in $\mathcal{A}$:

$$\forall (X, Y) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}, X \subseteq Y \Rightarrow f(X, \mathcal{D}) \geq f(Y, \mathcal{D}) \ .$$

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

$$\begin{aligned} f(\emptyset, \mathcal{D}) &= 4 \\ f(\{a_3\}, \mathcal{D}) &= 2 \\ f(\{a_1, a_3\}, \mathcal{D}) &= 1 \\ f(\{a_1, a_2, a_3\}, \mathcal{D}) &= 1 \end{aligned}$$

Anti-monotonicity of the frequency

Corollary

Given a dataset $\mathcal{D}$ of objects described with Boolean attributes in $\mathcal{A}$ and a minimal frequency $\mu \in \mathbb{N}$:

$$\forall (X, Y) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}, X \subseteq Y \Rightarrow \left(f(Y, \mathcal{D}) \geq \mu \Rightarrow f(X, \mathcal{D}) \geq \mu \right) .$$

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

$$\begin{aligned} f(\emptyset, \mathcal{D}) &= 4 \\ f(\{a_3\}, \mathcal{D}) &= 2 \\ f(\{a_1, a_3\}, \mathcal{D}) &= 1 \\ f(\{a_1, a_2, a_3\}, \mathcal{D}) &= 1 \end{aligned}$$

Anti-monotonicity of the frequency

Corollary

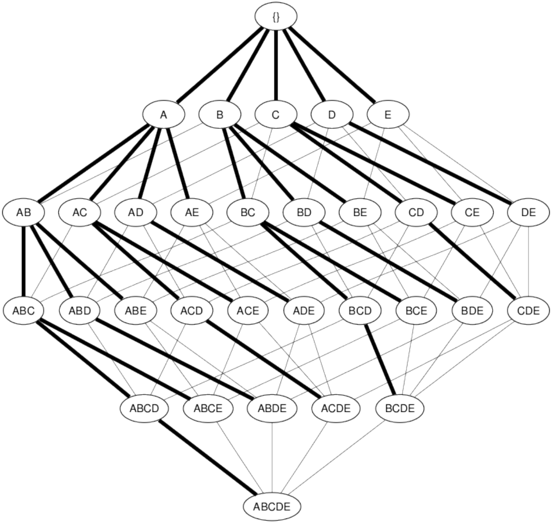
Given a dataset $\mathcal{D}$ of objects described with Boolean attributes in $\mathcal{A}$ and a minimal frequency $\mu \in \mathbb{N}$:

$$\forall (X, Y) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}, X \subseteq Y \Rightarrow \left(f(X, \mathcal{D}) < \mu \Rightarrow f(Y, \mathcal{D}) < \mu \right) .$$

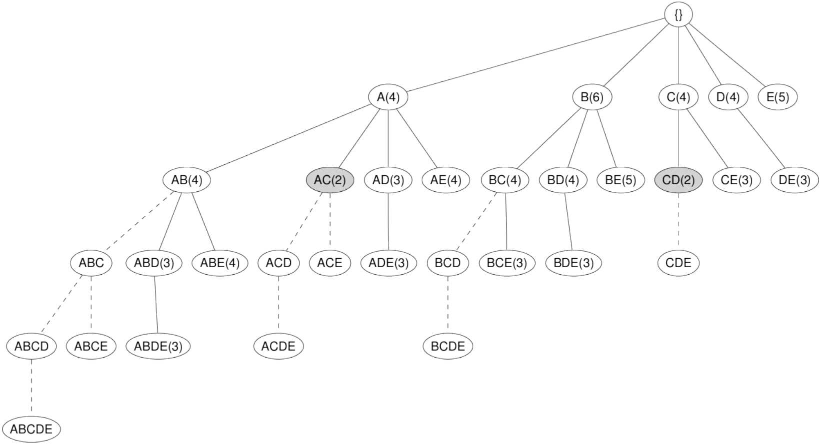
	a_1	a_2	a_3
o_1	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

$$\begin{aligned} f(\emptyset, \mathcal{D}) &= 4 \\ f(\{a_3\}, \mathcal{D}) &= 2 \\ f(\{a_1, a_3\}, \mathcal{D}) &= 1 \\ f(\{a_1, a_2, a_3\}, \mathcal{D}) &= 1 \end{aligned}$$

Pruning the enumeration tree ($\mu = 3$)



Pruning the enumeration tree ($\mu = 3$)



APriori enumeration

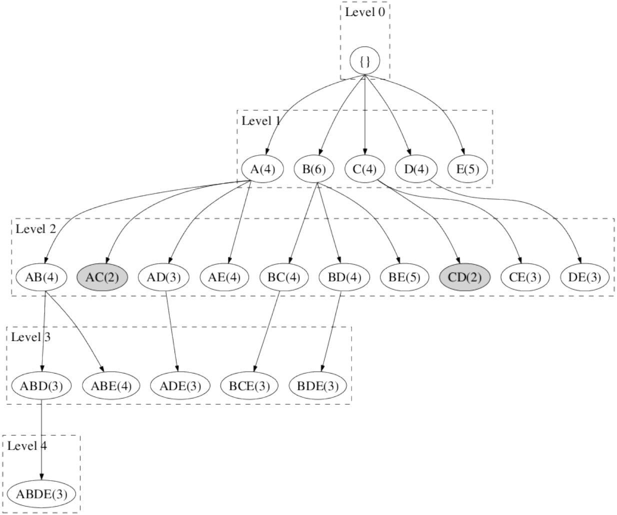
To check the frequency of every parent, the enumeration tree must be traversed breadth-first.

APriori enumeration

To check the frequency of every parent, the enumeration tree must be traversed breadth-first.

The two first parents (in the lexicographic order $\preceq$) are close to each other in the prefix-based tree. Indeed, they only differ by the last attribute. Instead of considering all possible children of a parent, APriori searches this second parent and, if found, enumerate, by union, their child.

Level-wise enumeration of the itemsets



APriori algorithm

Input: $\mathcal{A}, \mathcal{D}$ as an array of subsets of $\mathcal{A}, \mu \in \mathbb{N}$
Output: $\{X \subseteq \mathcal{A} \mid f(X, \mathcal{D}) \geq \mu\}$
$\mathcal{P} \leftarrow \{\{a\} \mid a \in \mathcal{A}\}$
while $\mathcal{P} \neq \emptyset$ **do**
  $\mathcal{P} \leftarrow$ **output_frequent**$(\mathcal{P}, \mathcal{D}, \mu)$
  $\mathcal{P} \leftarrow$ **children**$(\mathcal{P})$
end while

children

Input: A lexicographically ordered collection $\mathcal{P} \subseteq 2^{\mathcal{A}}$
Output: $\{X \subseteq 2^{\mathcal{A}} \mid \forall a \in X, X \setminus \{a\} \in \mathcal{P}\}$ lexico. ordered
$\mathcal{P}' \leftarrow \emptyset$
for all $P_1 \in \mathcal{P}$ **do**
  for all $P_2 \in \{P_2 \in \mathcal{P} \mid P_1 \prec P_2 \wedge P_2 \setminus \{\textbf{last}(P_2)\} = P_1 \setminus \{\textbf{last}(P_1)\}$
  do
    $X \leftarrow P_1 \cup P_2$
    if $\forall P \in \{X \setminus \{\textbf{member}(X)\} \mid P_2 \prec P\}, P \in \mathcal{P}$ **then**
      $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{X\}$
    end if
  end for
end for
return $\mathcal{P}'$

children

Input: A lexicographically ordered collection $\mathcal{P} \subseteq 2^{\mathcal{A}}$
Output: $\{X \subseteq 2^{\mathcal{A}} \mid \forall a \in X, X \setminus \{a\} \in \mathcal{P}\}$ lexico. ordered
$\mathcal{P}' \leftarrow \emptyset$
for all $P_1 \in \mathcal{P}$ **do**
  for all $P_2 \in \{P_2 \in \mathcal{P} \mid P_1 \prec P_2 \wedge P_2 \setminus \{\mathbf{last}(P_2)\} = P_1 \setminus \{\mathbf{last}(P_1)\}$
  do
    $X \leftarrow P_1 \cup P_2$
    if $\forall P \in \{X \setminus \{\mathbf{member}(X)\} \mid P_2 \prec P\}, P \in \mathcal{P}$ **then**
      $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{X\}$
    end if
  end for
end for
return $\mathcal{P}'$

Example

minsup = 2

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

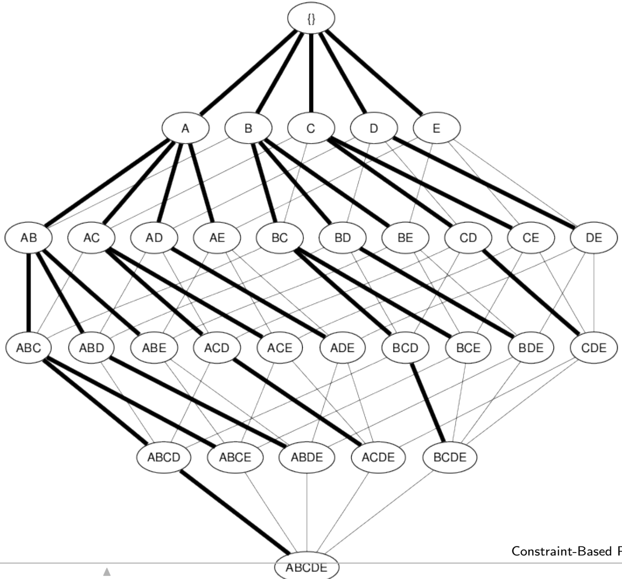
itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Depth-first enumeration of the itemsets



Fail-first principle

Observation

An itemset has a greater probability to be infrequent if the frequencies of its attributes, taken individually, are low.

Fail-first principle

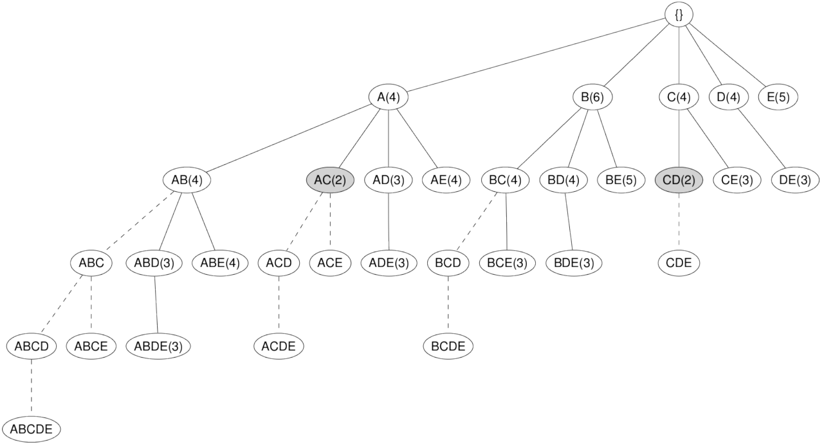
Observation

An itemset has a greater probability to be infrequent if the frequencies of its attributes, taken individually, are low.

Fail-first principle

Taking advantage of the anti-monotonicity of the frequency, it is better to enumerate the infrequent itemsets first.

The unbalanced enumeration tree



Heuristic choice of a lexicographic order

Input: $\mathcal{A}, \mathcal{D}$ as an array of subsets of $\mathcal{A}, \mu \in \mathbb{N}$
Output: $\{X \subseteq \mathcal{A} \mid f(X, \mathcal{D}) \geq \mu\}$
$\mathcal{P} \leftarrow \{\{a\} \mid a \in \mathcal{A}\}$
while $\mathcal{P} \neq \emptyset$ **do**
  $\mathcal{P} \leftarrow$ **output_frequent**$(\mathcal{P}, \mathcal{D}, \mu)$
  $\mathcal{P} \leftarrow$ **children**$(\mathcal{P})$
end while

Whatever the order on $\mathcal{A}$, the frequent itemsets are correctly and completely listed...

Heuristic choice of a lexicographic order

Input: $\mathcal{A}, \mathcal{D}$ as an array of subsets of $\mathcal{A}, \mu \in \mathbb{N}$
Output: $\{X \subseteq \mathcal{A} \mid f(X, \mathcal{D}) \geq \mu\}$
$\mathcal{P} \leftarrow \{\{a\} \mid a \in \mathcal{A}\}$ ordered by increasing $f(\{a\}, \mathcal{D})$
while $\mathcal{P} \neq \emptyset$ **do**
  $\mathcal{P} \leftarrow$ **output_frequent**$(\mathcal{P}, \mathcal{D}, \mu)$
  $\mathcal{P} \leftarrow$ **children**$(\mathcal{P})$
end while

Whatever the order on $\mathcal{A}$, the frequent itemsets are correctly and completely listed... but this heuristic choice usually leads to the enumeration of much less infrequent itemsets.

Iterative computation of the supports

Theorem

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, i. e., the dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ and $k \in \mathbb{N}$ itemsets $(P_i)_{i=1..k} \in (2^{\mathcal{A}})^k$:

$$\{o \in \mathcal{O} \mid \{o\} \times \cup_{i=1}^{k} P_i \subseteq \mathcal{D}\} = \cap_{i=1}^{k} \{o \in \mathcal{O} \mid \{o\} \times P_i \subseteq \mathcal{D}\} \ .$$

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

$$\begin{array}{rcl}
\{o \in \mathcal{O} \mid \{o\} \times \{a_1\} \subseteq \mathcal{D}\} & = & \{o_1, o_2\} \\
\{o \in \mathcal{O} \mid \{o\} \times \{a_2\} \subseteq \mathcal{D}\} & = & \{o_1, o_2, o_3\} \\
\{o \in \mathcal{O} \mid \{o\} \times \{a_3\} \subseteq \mathcal{D}\} & = & \{o_1, o_4\} \\
\hline
\{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2, a_3\} \subseteq \mathcal{D}\} & = & \{o_1\}
\end{array}$$

Iterative computation of the supports

Theorem

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, i. e., the dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ and $k \in \mathbb{N}$ itemsets $(P_i)_{i=1..k} \in (2^{\mathcal{A}})^k$:

$$\{o \in \mathcal{O} \mid \{o\} \times \cup_{i=1}^{k} P_i \subseteq \mathcal{D}\} = \cap_{i=1}^{k} \{o \in \mathcal{O} \mid \{o\} \times P_i \subseteq \mathcal{D}\} \ .$$

	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

$$\begin{array}{rcl} \{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2\} \subseteq \mathcal{D}\} & = & \{o_1, o_2\} \\ \{o \in \mathcal{O} \mid \{o\} \times \{a_3\} \subseteq \mathcal{D}\} & = & \{o_1, o_4\} \\ \hline \{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2, a_3\} \subseteq \mathcal{D}\} & = & \{o_1\} \end{array}$$

Iterative computation of the supports

Theorem

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, i. e., the dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ and $k \in \mathbb{N}$ itemsets $(P_i)_{i=1..k} \in (2^{\mathcal{A}})^k$:

$$\{o \in \mathcal{O} \mid \{o\} \times \cup_{i=1}^{k} P_i \subseteq \mathcal{D}\} = \cap_{i=1}^{k} \{o \in \mathcal{O} \mid \{o\} \times P_i \subseteq \mathcal{D}\} \ .$$

	a_1	a_2	a_3
o_1	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

$$\begin{array}{rcl} \{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2\} \subseteq \mathcal{D}\} & = & \{o_1, o_2\} \\ \{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_3\} \subseteq \mathcal{D}\} & = & \{o_1\} \\ \hline \{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2, a_3\} \subseteq \mathcal{D}\} & = & \{o_1\} \end{array}$$

Vertical representation of the data

Relational representation:
$\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

$$\begin{array}{c|cccc} & a_1 & a_2 & \ldots & a_n \\ \hline o_1 & d_{1,1} & d_{1,2} & \ldots & d_{1,n} \\ o_2 & d_{2,1} & d_{2,2} & \ldots & d_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_m & d_{m,1} & d_{m,2} & \ldots & d_{m,n} \end{array}$$

where $d_{i,j} \in \{\text{true,false}\}$

Vertical representation: $\mathcal{D}$ is an array of subsets of $\mathcal{O}$

$$i_1 \quad i_2 \quad \ldots \quad i_n$$

where $i_j \subseteq \mathcal{O}$

Vertical representation of the data

Relational representation:
$\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

	a_1	a_2	$\dots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\dots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\dots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\dots$	$d_{m,n}$

where $d_{i,j} \in \{\text{true,false}\}$

Vertical representation: $\mathcal{D}$ is an array of subsets of $\mathcal{O}$

$$i_1 \quad i_2 \quad \dots \quad i_n$$

where $i_j \subseteq \mathcal{O}$

For a linear time intersection of the i_j, they are sorted (arbitrary order on $\mathcal{O}$) in a pre-processing step and the support of any enumerated itemset X will respect this order.

Vertical representation of the data

Relational representation:
$\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$

	a_1	a_2	$\ldots$	a_n
o_1	$d_{1,1}$	$d_{1,2}$	$\ldots$	$d_{1,n}$
o_2	$d_{2,1}$	$d_{2,2}$	$\ldots$	$d_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_m	$d_{m,1}$	$d_{m,2}$	$\ldots$	$d_{m,n}$

where $d_{i,j} \in \{\text{true,false}\}$

Vertical representation: $\mathcal{D}$ is an array of subsets of $\mathcal{O}$

$$i_1 \quad i_2 \quad \ldots \quad i_n$$

where $i_j \subseteq \mathcal{O}$

Unless the minimal relative frequency is very low, storing the support on *bitsets* provide the best space and time performances.

Eclat enumeration

Like APriori:

- The anti-monotonicity of the frequency prunes the enumeration tree;

Eclat enumeration

Like APriori:

- The anti-monotonicity of the frequency prunes the enumeration tree;
- the two first parents (in the lexicographic order $\preceq$) are searched to generate by union their child;

Eclat enumeration

Like APriori:

- The anti-monotonicity of the frequency prunes the enumeration tree;
- the two first parents (in the lexicographic order $\preceq$) are searched to generate by union their child;
- Ordering the attributes by increasing frequency heuristically leads to the enumeration of much less infrequent itemsets.

Eclat enumeration

Like APriori:

- The anti-monotonicity of the frequency prunes the enumeration tree;
- the two first parents (in the lexicographic order $\preceq$) are searched to generate by union their child;
- Ordering the attributes by increasing frequency heuristically leads to the enumeration of much less infrequent itemsets.

However:

- the frequency of the other parents is not checked;

Eclat enumeration

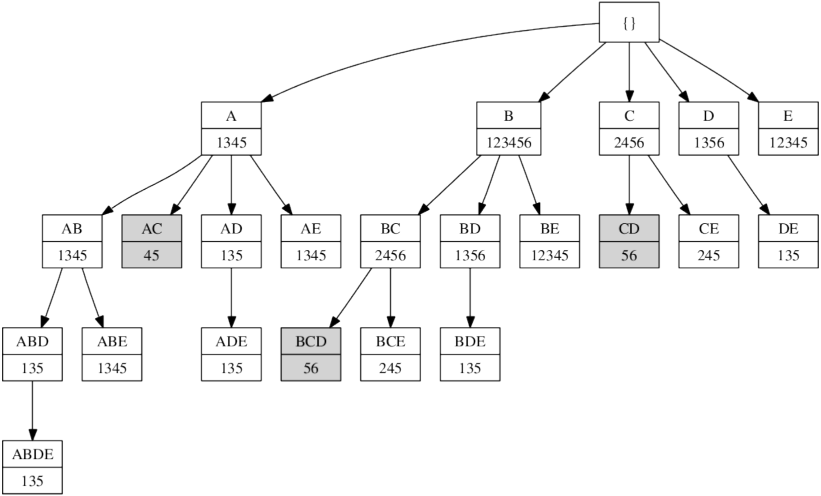
Like APriori:

- The anti-monotonicity of the frequency prunes the enumeration tree;
- the two first parents (in the lexicographic order $\preceq$) are searched to generate by union their child;
- Ordering the attributes by increasing frequency heuristically leads to the enumeration of much less infrequent itemsets.

However:

- the frequency of the other parents is not checked;
- thanks to that, the enumeration tree is traversed in a less memory-hungry way (but, contrary to APriori, the supports of the frequent itemsets are stored too).

Pruning the enumeration tree ($\mu = 3$)



Eclat algorithm

Input: $\mathcal{A}, \mathcal{D}$ as an array of subsets of $\mathcal{O}, \mu \in \mathbb{N}$
Output: $\{X \subseteq \mathcal{A} \mid f(X, \mathcal{D}) \geq \mu\}$
Eclat$(\mathcal{P}, \mu)$ {Initial call: $\mathcal{P} = \{(\{a_j\}, i_j) \mid j = 1..m \wedge |i_j| \geq \mu\}$}
for all $(P_1, i_{P_1}) \in \mathcal{P}$ **do**
  output(P_1)
  $\mathcal{P}' \leftarrow \emptyset$
  for all $(P_2, i_{P_2}) \in \{(P_2, i_{P_2}) \in \mathcal{P} \mid P_1 \prec P_2\}$ **do**
    $i \leftarrow i_{P_1} \cap i_{P_2}$
    if $|i| \geq \mu$ **then**
      $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{(P_1 \cup P_2, i)\}$
    end if
  end for
  Eclat$(\mathcal{P}', \mu)$
end for

Pattern flooding

$\mu = 2$

$\mathcal{O}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
o_1	×	×	×	×	×										
o_2	×	×	×	×	×										
o_3	×	×	×	×	×										
o_4						×	×	×	×	×					
o_5						×	×	×	×	×					
o_6						×	×	×	×	×					
o_7											×	×	×	×	×
o_8											×	×	×	×	×

- How many frequent patterns?

Pattern flooding

$\mu = 2$

$\mathcal{O}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
o_1	×	×	×	×	×										
o_2	×	×	×	×	×										
o_3	×	×	×	×	×										
o_4						×	×	×	×	×					
o_5						×	×	×	×	×					
o_6						×	×	×	×	×					
o_7											×	×	×	×	×
o_8											×	×	×	×	×

- How many frequent patterns? $1 + (2^5 - 1) \times 3 = 94$ patterns

Pattern flooding

$\mu = 2$

$\mathcal{O}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
o_1	×	×	×	×	×										
o_2	×	×	×	×	×										
o_3	×	×	×	×	×										
o_4						×	×	×	×	×					
o_5						×	×	×	×	×					
o_6						×	×	×	×	×					
o_7											×	×	×	×	×
o_8											×	×	×	×	×

- How many frequent patterns? $1 + (2^5 - 1) \times 3 = 94$ patterns but actually 4 interesting ones: $\{\}, \{a_1, a_2, a_3, a_4, a_5\}, \{a_6, a_7, a_8, a_9, a_{10}\}, \{a_{11}, a_{12}, a_{13}, a_{14}, a_{15}\}$.

Pattern flooding

$\mu = 2$

$\mathcal{O}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
o_1	×	×	×	×	×										
o_2	×	×	×	×	×										
o_3	×	×	×	×	×										
o_4						×	×	×	×	×					
o_5						×	×	×	×	×					
o_6						×	×	×	×	×					
o_7											×	×	×	×	×
o_8											×	×	×	×	×

- How many frequent patterns? $1 + (2^5 - 1) \times 3 = 94$ patterns but actually 4 interesting ones: $\{\}, \{a_1, a_2, a_3, a_4, a_5\}, \{a_6, a_7, a_8, a_9, a_{10}\}, \{a_{11}, a_{12}, a_{13}, a_{14}, a_{15}\}$.

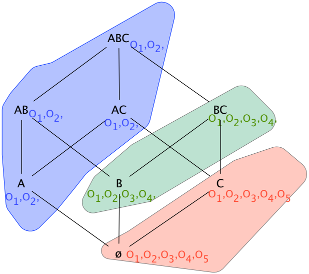
☞ the need to focus on a **condensed representation** of frequent patterns.

Toon Calders, Christophe Rigotti, Jean-François Boulicaut: A Survey on Condensed Representations for Frequent Sets. Constraint-Based Mining and Inductive Databases 2004: 64-80.

Closed and Free Patterns

Equivalence classes based on support.

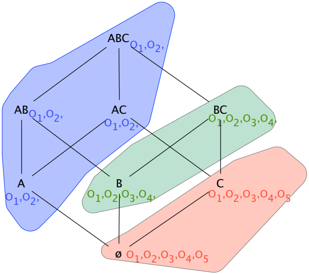
$\mathcal{O}$	A	B	C
o_1	×	×	×
o_2	×	×	×
o_3		×	×
o_4		×	×
o_5			×



Closed and Free Patterns

Equivalence classes based on support.

$\mathcal{O}$	A	B	C
o_1	×	×	×
o_2	×	×	×
o_3		×	×
o_4		×	×
o_5			×



- **Closed** patterns are maximal element of each equivalence class: ABC, BC, and C.
- **Generators** or **Free** patterns are minimal elements (not necessary unique) of each equivalent class: $\{\}$, A and B

Y. Bastide, et al. Mining frequent patterns with counting inference. SIGKDD Expl., 2000.

Outline

Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

Pattern constraints

Constraints are needed for:

- only retrieving patterns that describe an interesting subgroup of the data
- making the extraction feasible

Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

➜ They are defined up to the partial order $\preceq$ used for listing the patterns

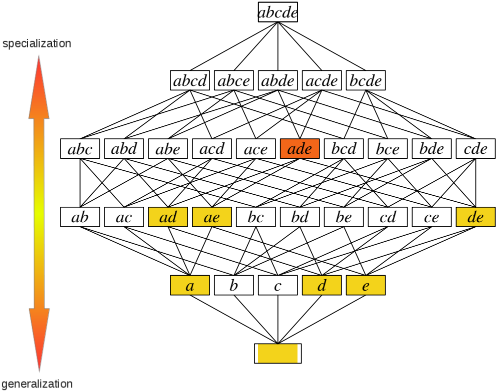
Constraint properties - 1

Monotone constraint

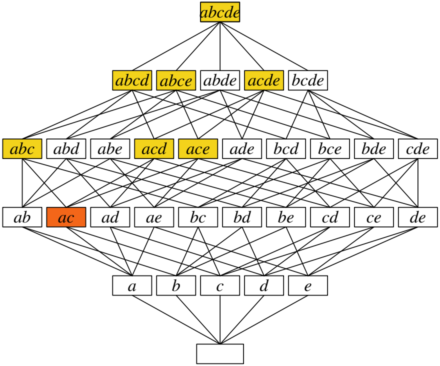
$\forall \varphi_1 \preceq \varphi_2,\ \mathcal{C}(\varphi_1, \mathcal{D}) \Rightarrow \mathcal{C}(\varphi_2, \mathcal{D})$

Anti-monotone constraint

$\forall \varphi_1 \preceq \varphi_2,\ \mathcal{C}(\varphi_2, \mathcal{D}) \Rightarrow \mathcal{C}(\varphi_1, \mathcal{D})$



$\mathcal{C}(\varphi, \mathcal{D}) \equiv b \in \varphi \vee c \in \varphi$

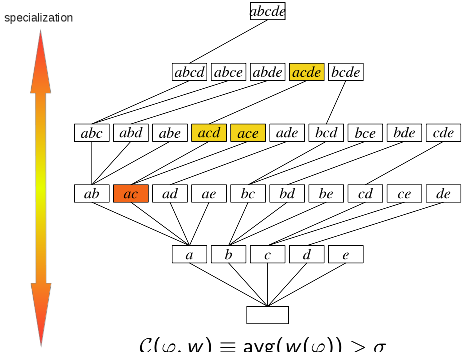


$\mathcal{C}(\varphi, \mathcal{D}) \equiv a \notin \varphi \wedge c \notin \varphi$

Constraint properties - 2

Convertible constraints

$\preceq$ is extended to the prefix order $\leq$ so that $\forall \varphi_1 \leq \varphi_2$, $\mathcal{C}(\varphi_2, \mathcal{D}) \Rightarrow \mathcal{C}(\varphi_1, \mathcal{D})$



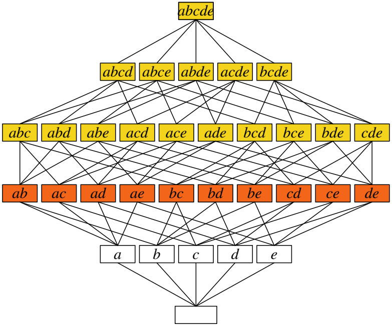
$\mathcal{C}(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma$

$w(a) \geq w(b) \geq w(c) \geq w(d) \geq w(e)$

Pei and Han – 2000

Loose AM constraints

$\mathcal{C}(\varphi, \mathcal{D}) \Rightarrow \exists e \in \varphi \; : \; \mathcal{C}(\varphi \setminus \{e\}, \mathcal{D})$



$\mathcal{C}(\varphi, w) \equiv \text{var}(w(\varphi)) \leq \sigma$

Bonchi and Lucchese – 2007

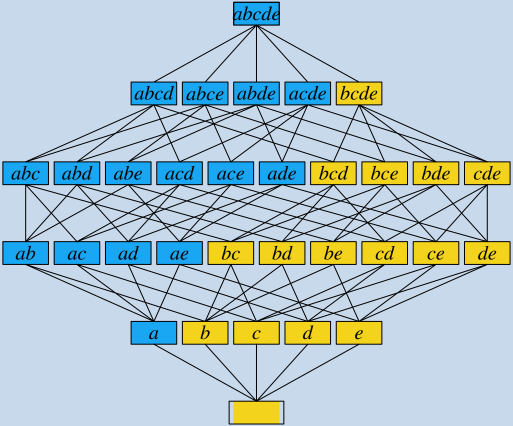
Examples

$v \in P$	M
$P \supseteq S$	M
$P \subseteq S$	AM
$min(P) \leq \sigma$	AM
$min(P) \geq \sigma$	M
$max(P) \leq \sigma$	M
$max(P) \leq \sigma$	AM
$range(P) \leq \sigma$	AM
$range(P) \geq \sigma$	M
$avg(P)\theta\sigma, \theta \in \{\leq, =, \geq\}$	Convertible
$\mathrm{var}(w(\varphi)) \leq \sigma$	LAM

Outline

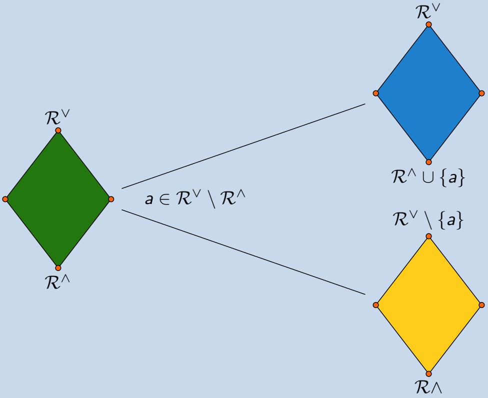
Enumeration strategy

Binary partition: the element 'a' is enumerated



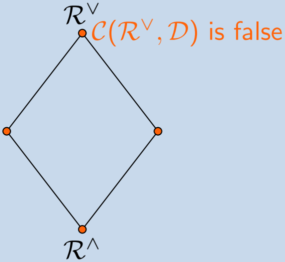
Enumeration strategy

Binary partition: the element 'a' is enumerated



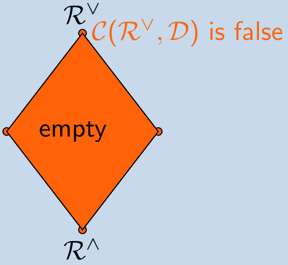
Constraint evaluation

Monotone constraint



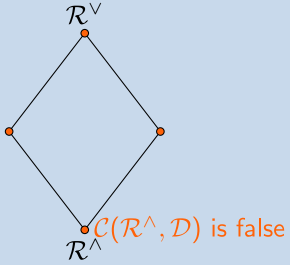
Constraint evaluation

Monotone constraint



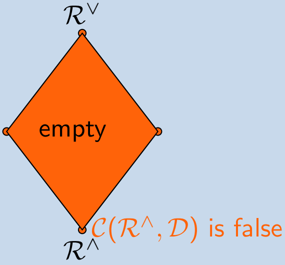
Constraint evaluation

Anti-monotone constraint



Constraint evaluation

Anti-monotone constraint



A new class of constraints

Piecewise monotone and anti-monotone constraints[a]

1. $\mathcal{C}$ involves p times the pattern φ: $\mathcal{C}(\varphi, \mathcal{D}) = f(\varphi_1, \cdots \varphi_p, \mathcal{D})$
2. $f_{i,\varphi}(x) = (\varphi_1, \cdots \varphi_{i-1}, x, \varphi_{i+1}, \cdots, \varphi_p, \mathcal{D})$
3. $\forall i = 1 \ldots p$, $f_{i,\varphi}$ is either monotone or anti-monotone:

$$\forall\, x \preceq y, \begin{cases} f_{i,\varphi}(x) \Rightarrow f_{i,\varphi}(y) \text{ iff } f_{i,\varphi} \text{ is monotone} \\ f_{i,\varphi}(y) \Rightarrow f_{i,\varphi}(x) \text{ iff } f_{i,\varphi} \text{ is anti-monotone} \end{cases}$$

[a] A.k.a. primitive-based constraints

A.Soulet, B. Crémilleux: Mining constraint-based patterns using automatic relaxation. Intell. Data Anal. 13(1): 109-133 (2009)

L. Cerf, J. Besson, C. Robardet, J-F. Boulicaut: Closed patterns meet n-ary relations. TKDD 3(1) (2009)

A. Buzmakov, S. O. Kuznetsov, A.Napoli: Fast Generation of Best Interval Patterns for Nonmonotonic Constraints. ECML/PKDD (2) 2015: 157-172

An example

- $\forall e,\ w(e) \geq 0$
- $\mathcal{C}(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma \equiv \frac{\sum_{e \in \varphi} w(e)}{|\varphi|} > \sigma$.

$\mathcal{C}(\varphi, \mathcal{D})$ is piecewise monotone and anti-monotone with

$$f(\varphi_1, \varphi_2, \mathcal{D}) = \frac{\sum_{e \in \varphi_1} w(e)}{|\varphi_2|}$$

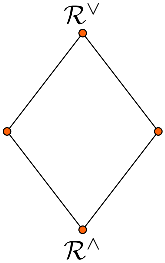
$\forall x \preceq y$,

- $f_{1,\varphi}$ is monotone: $f(x, \varphi_2, \mathcal{D}) = \frac{\sum_{e \in x} w(e)}{|\varphi_2|} > \sigma \Rightarrow \frac{\sum_{e \in y} w(e)}{|\varphi_2|} > \sigma$
- $f_{2,\varphi}$ is anti-monotone: $f(\varphi_1, y, \mathcal{D}) = \frac{\sum_{e \in \varphi_1} w(e)}{|y|} > \sigma \Rightarrow \frac{\sum_{e \in \varphi_1} w(e)}{|x|} > \sigma$

Piecewise constraint exploitation

Evaluation

If $f(\mathcal{R}^{\vee}, \mathcal{R}^{\wedge}, \mathcal{D}) = \frac{\sum_{e \in \mathcal{R}^{\vee}} w(e)}{|\mathcal{R}^{\wedge}|}$



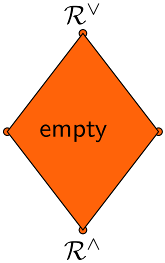
Propagation

- $\exists e \in \mathcal{R}^{\vee} \setminus \mathcal{R}^{\wedge}$ such that $f(\mathcal{R}^{\vee} \setminus \{e\}, \mathcal{R}^{\wedge}, \mathcal{D}) \leq \sigma$, then e is moved in $\mathcal{R}^{\wedge}$
- $\exists e \in \mathcal{R}^{\vee} \setminus \mathcal{R}^{\wedge}$ such that $f(\mathcal{R}^{\vee}, \mathcal{R}^{\wedge} \cup \{e\}, \mathcal{D}) \leq \sigma$, then e is removed from $\mathcal{R}^{\vee}$

Piecewise constraint exploitation

Evaluation

If $f(\mathcal{R}^\vee, \mathcal{R}^\wedge, \mathcal{D}) = \frac{\sum_{e \in \mathcal{R}^\vee} w(e)}{|\mathcal{R}^\wedge|} \leq \sigma$ then $\mathcal{R}$ is empty.



Propagation

- $\exists e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge$ such that $f(\mathcal{R}^\vee \setminus \{e\}, \mathcal{R}^\wedge, \mathcal{D}) \leq \sigma$, then e is moved in $\mathcal{R}^\wedge$
- $\exists e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge$ such that $f(\mathcal{R}^\vee, \mathcal{R}^\wedge \cup \{e\}, \mathcal{D}) \leq \sigma$, then e is removed from $\mathcal{R}^\vee$

Algorithmic principles

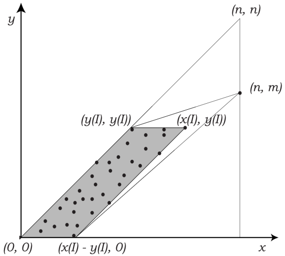
Function Generic_CBPM_enumeration($\mathcal{R}^\vee, \mathcal{R}^\wedge$)

if Check_constraints($\mathcal{R}^\wedge, \mathcal{R}^\vee$) **then**
    $(\mathcal{R}^\wedge, \mathcal{R}^\vee) \leftarrow$ Constraint_Propagation($\mathcal{R}^\wedge, \mathcal{R}^\vee$)
    if $\mathcal{R}^\wedge = \mathcal{R}^\vee$ **then**
        output $\mathcal{R}^\wedge$
    else
        for all $e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge$ **do**
            Generic_CBPM_Enumeration($\mathcal{R}^\wedge \cup \{e\}, \mathcal{R}^\vee$)
            Generic_CBPM_Enumeration($\mathcal{R}^\wedge, \mathcal{R}^\vee \setminus \{e\}$)
        end for
    end if
end if

Open Question

Is convexity $\equiv$ piece-wise constraints ?

- Convex measures can be taken into account by computing some upper bounds with $\mathcal{R}^\wedge$ and $\mathcal{R}^\vee$.
- Branch and bound enumeration



Shinichi Morishita, Jun Sese: Traversing Itemset Lattice with Statistical Metric Pruning. PODS 2000: 226-236

Case Studies

Mining of

- Multidimensional and multi-level sequences [ACM TKDD 2010]
- Maximal homogeneous clique set [KAIS 2014]
- Rules in Boolean tensors/dynamic graphs [SDM 11, IDA J. 2013]
- Topological patterns in static attributed graphs [TKDE 2013]
- Temporal dependencies in streams [KDD'13, IDA J. 2016]
- Trend dynamic sub-graphs [DS 12, PKDD 13, IDA 14]
- δ-free sequential patterns [ICDM'14]
- Triggering patterns [ASONAM 14, Social Network Analysis J. 2015]
- Events in geo-localized social medias [ECMLPKDD'15]

Outline

The Thresholding Issue

- *Avoid the threshold issue*
    - What is the "best" value of my minimal frequency?
    - Which k in top-k?
    - Combining several measures?
- *Give the end-user a new and easy way to express his preferences*
    - In a multidimensional space: each dimension is a measure
- *Discovering patterns satisfying a global property*
    - Dominance relation
    - *The skyline operator* over the pattern domains

What if it also gives a way to discover less (and useful) patterns?

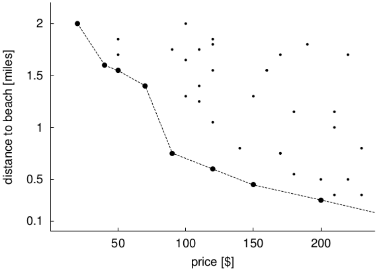
Motivations: Why Skylines?

Introduced by [Börzsönyi et al. @ICDE 2001].

Hotel Example

Hotel on the beach

F_ID	Price	Distance to the sea (min)
f_1	2	11
f_2	5	7
f_3	3	13
f_4	**2**	**10**
f_5	3	10
f_6	**4**	**7**



- Data point X *dominates* Y if all attributes of X are **better than or equal to** the corresponding attributes from Y
- A skyline query returns all data points that are not *dominated by others*

Notion of Skyline Patterns

The basic idea: if a pattern is dominated by another according to all measures in a set M then it is discarded in the output.
($X \succ_M Y$: X dominates Y)

Let P be a pattern set. A **skypattern** of P with respect to M is a pattern not dominated in P with respect to M.

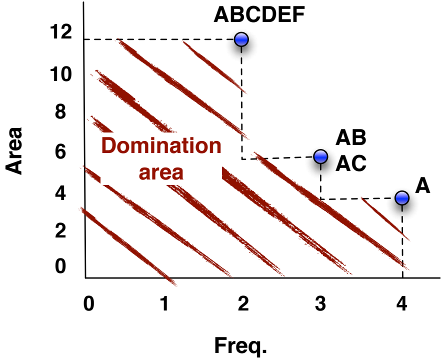
The **skypattern operator** $\mathcal{S}ky(P, M)$: returns all the skypatterns of P with respect to M:

$$\mathcal{S}ky(P, M) = \{X \in P | \nexists Y \in P : Y \succ_M X\}$$

Example

Tid	Items
t_1	$A\ B\ C\ D\ E\ F$
t_2	$A\ B\ C\ D\ E\ F$
t_3	$A\ B$
t_4	D
t_5	$A\ C$
t_6	E

Patterns	freq	length	area
$ABCDEF$	2	6	12
AB	3	2	6
AC	3	2	6
A	4	1	4



$$Sky(\mathcal{L}, \{freq, area\}) = \{ABCDEF, AB, AC, A\}$$

$ABCD$, C, E are in the domination area.

Many other measures can be addressed : min(x.price), sum(x.val), etc.

Algorithmic Issues and Objective

Mining Task: $\mathcal{S}ky$ **($\mathcal{L}$, M)**

Given a set of measures M, we aim at returning all the skypatterns w.r.t M.

- A naive enumeration of all candidate patterns ($\mathcal{L}$) and then comparisons between them is not possible.
- **Key idea:** Take benefit from the pattern condensed representation according to the condensable measures of M.

Basic Definitions

Preserving function

Let E be a set. A function $p : \mathcal{L} \rightarrow E$ is preserving iff for each $i \in \mathcal{I}$ and for each $X \subseteq Y$ if $p(X \cup \{i\}) = p(X)$ then $p(Y \cup i)$ equals to p(Y).

The addition of an item i does not modify $p(X)$, then the addition of i does not modify the value of p for any specialization of X.

Ex.: freq, $\text{freq}_{\vee}$, count, min, max, sum, etc.

A. Soulet and B. Crémilleux, ECML/PKDD 2008.

Basic Definitions

Preserving function

Let E be a set. A function $p : \mathcal{L} \to E$ is preserving iff for each $i \in \mathcal{I}$ and for each $X \subseteq Y$ if $p(X \cup \{i\}) = p(X)$ then $p(Y \cup i)$ equals to p(Y).

The addition of an item i does not modify $p(X)$, then the addition of i does not modify the value of p for any specialization of X.

Ex.: freq, $\text{freq}_\vee$, count, min, max, sum, etc.

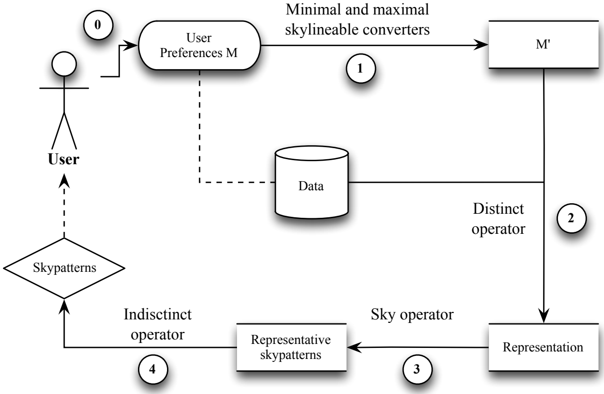
Condensable function

Let E be a set. A function $f : \mathcal{L} \to E$ is condensable iff there exist a function F and k preserving functions $p1, ..., pk$ such that $f = F(p1, ..., pk)$.

Condensable function is a compound of preserving functions $\equiv$ Piece-wise (anti-)monotone constraint.

A. Soulet and B. Crémilleux, ECML/PKDD 2008.

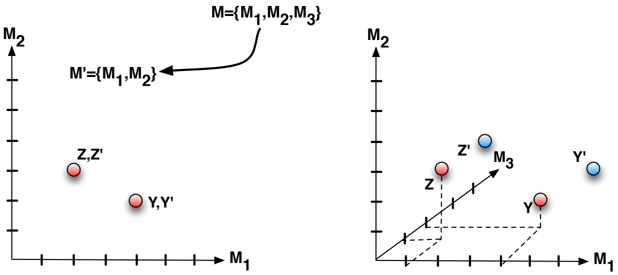
The Aetheris Approach



A. Soulet, C. Raïssi, M. Plantevit, B. Crémilleux, ICDM 2011

Skylineability

1. Looking for a smaller set of measures M' from M enabling us to focus on a condensed representation.
M is M'-**skylineable** with respect to $\subset$ (resp. $\supset$) iff for any patterns $X =_{M'} X'$ such that $X \subset X'$ (resp. $X \supset X'$), one has $X \succeq_M X'$.



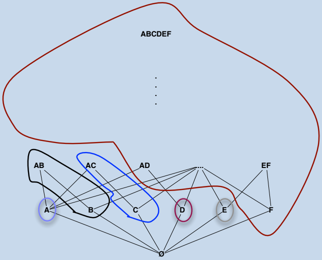
M is M'-skylineable w.r.t. $\subset$:

$$\begin{cases} Z =_{M'} Z' \text{ and } Z \subset Z' \longrightarrow Z \succeq_M Z' \\ Y =_{M'} Y' \text{ and } Y \subset Y' \longrightarrow Y \succeq_M Y' \end{cases}$$

Skylineability: Example

$M = \{freq, area\}$ is strictly $\{freq\}$-skylineable w.r.t. $\supset$.

Tid	Items
t_1	A B C D E F
t_2	A B C D E F
t_3	A B
t_4	D
t_5	A C
t_6	E



Patterns	freq	length
$ABCDEF$	2	6
AB	3	2
AC	3	2
A	4	1

$B =_{freq} AB$: we can directly deduce that $AB \succ_M B$.

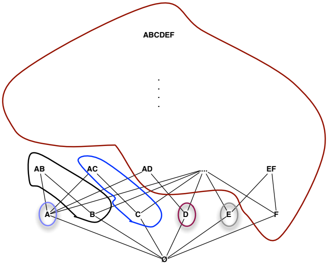
Minimal and maximal skylineable converters to compute M'.

Computing Concise Representations According to M'

$$\mathcal{D}is_\theta(P, M') = \{X \in P | \forall Y \theta X : X \neq_{M'} Y\} \text{ where } \theta \in \{\subset, \supset\}$$

Distinct operator: returns all the patterns X of P such that their generalizations (or specializations) are distinct from X w.r.t. M.

Tid	Items					
t_1	A	B	C	D	E	F
t_2	A	B	C	D	E	F
t_3	A	B				
t_4				D		
t_5	A		C			
t_6					E	



Example

$\mathcal{D}is_\subset(\mathcal{L}, \{freq\}) = \{A, B, C, D, E, F, AD, AE, BC, BD, BE, CD, CE, DE\}$ and $\mathcal{D}is_\supset(\mathcal{L}, \{freq\}) = \{A, D, E, AB, AC, ABCDEF\}$

Aetheris Approach

1. Compute the best M'
2. Process distinct patterns given M'
3. Compute the skyline patterns from the condensed representation
4. Finalize by generating all the skypatterns: retrieving of all the indistinct patterns from their representatives

$$\mathcal{I}nd(\mathcal{L}, M', P) = \{X \in \mathcal{L} | \exists Y \in P : X =_{M'} Y\}$$

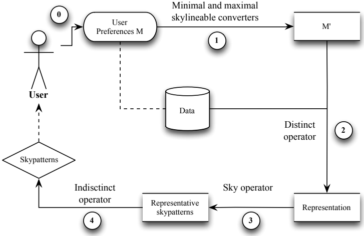
Example: $\mathcal{I}nd(\mathcal{L}, \{freq\}, \{AB, AC\}) = \{B, C, AB, AC\}$

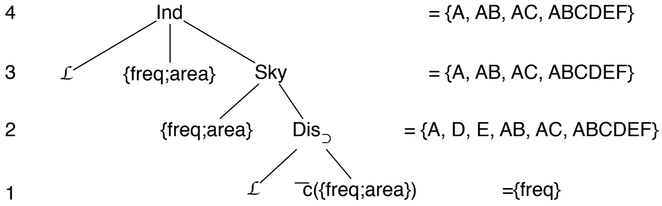
Finally:

$$\mathcal{S}ky(\mathcal{L}, M) = \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M))$$

Aetheris Approach: Example

Tid	Items					
t_1	A	B	C	D	E	F
t_2	A	B	C	D	E	F
t_3	A	B				
t_4				D		
t_5	A		C			
t_6					E	





Experiments on Itemset Data ($\mathcal{L} = 2^{\mathcal{I}}$)

Experiments on UCI data

- 16 benchmarks.
- Synthesis of 128 experiments.
- Runtimes only consider the application of skyline operator.

Comparisons of 3 approaches

1. **Baseline approach:** $\mathcal{S}ky\ (\{X \subseteq \mathcal{I} \mid freq(X, \mathcal{D}) \geq 1\}, M)$.
2. **Optimal Constraint-Based approach:** Assume that user set the *optimal* thresholds, i.e. for each measure $M_i \in M$,

$$\sigma_{M_i} := \min_{s \in \mathcal{S}ky\ (\mathcal{L},\ \mathrm{M})} (M_i(s))$$

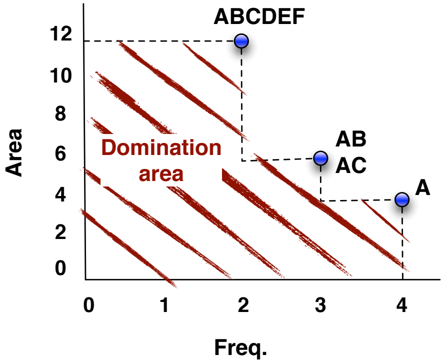
.

3. **Aetheris approach:** $\mathcal{S}ky(\mathcal{L}, M) = \mathcal{I}nd(\mathcal{L}, M, \mathcal{S}ky(\mathcal{D}is_\theta(\mathcal{L}, M'), M))$

Optimal Constraint-Based Approach Settings in a Nutshell

Tid	Items
t_1	A B C D E F
t_2	A B C D E F
t_3	A B
t_4	D
t_5	A C
t_6	E

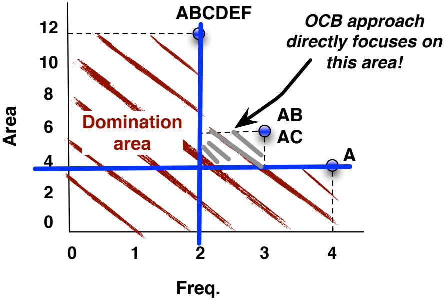
Patterns	freq	length	area
$ABCDEF$	2	6	12
AB	3	2	6
AC	3	2	6
A	4	1	4



$$Sky(\mathcal{L}, \{freq, area\}) = \{ABCDEF, AB, AC, A\}$$

Optimal Constraint-Based Approach Settings in a Nutshell

Tid	Items
t_1	A B C D E F
t_2	A B C D E F
t_3	A B
t_4	D
t_5	A C
t_6	E



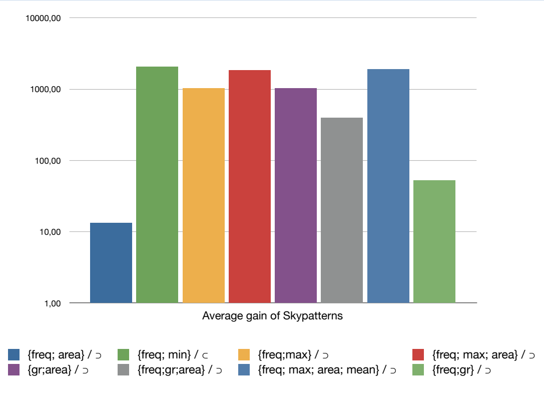
Patterns	freq	length	area
$ABCDEF$	2	6	12
AB	3	2	6
AC	3	2	6
A	4	1	4

$$Sky(\mathcal{L}, \{freq, area\}) = \{ABCDEF, AB, AC, A\}$$

☛ $\sigma_{sup} = 2$ and $\sigma_{area} = 4$

Results: Conciseness Gain

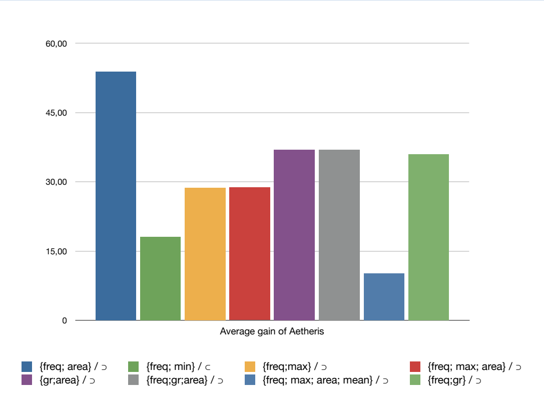
Average gain of skypatterns according to OCB patterns



The gain of a skyline approach is always important (greater than 10 and much greater in almost all the cases).

Results: Performance Gain

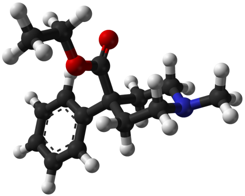
Runtime gain of Aetheris according to Baseline

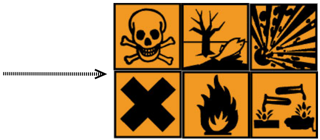


Aetheris always outperforms the baseline approach with at least a factor of 10.

Case Study: Discovering Toxicophores

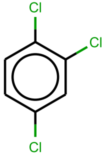
- Collaboration with the CERM Laboratory.
- Establishing relationships between chemicals and (eco)toxicity





Our aim: investigating the use of skypatterns to discover toxicophores

ECB[a] dataset: 567 chemicals (372 very toxic/195 harmful)



trichlorobenzene

[a]European Chemicals Bureau – http://echa.europa.eu/

Case study: Results

Experiment 1: contrast measures (e.g., growth rate) are useful to discover toxicophores

- only 8 skypatterns!
- the method is able to automatically discover already known environmental toxicophores:
  ➥ it suggests good insights for the others

Experiment 2: background knowledge can easily be integrating adding aromaticity and density measures

- the whole set of skypatterns remains small (38 skypatterns)
- discovering of skypatterns including an amine function not detected in Experiment 1

- Useful results from a *user-preference* point of view.
- No thresholds → Threshold-free constraint based pattern mining is possible!
- Enable to mine **pattern sets** (global constraint).
- The skyline operator can be pushed within the extraction:

W. Ugarte, P. Boizumault, B. Crémilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raïssi, A. Soulet, Artificial Intelligence 2015.

B. Négrevergne, A. Dries, T. Guns, S. Nijssen, ICDM 2013.

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!

Sequence mining

Key notions

- $\mathcal{I} = \{i_1, i_2 \ldots i_m\}$ 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.

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.

⇒ 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

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

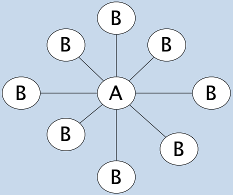
In a graph collection

- 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

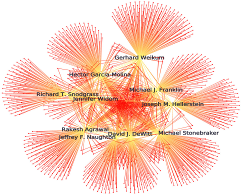
Let $\gamma \in]0,1]$, $C \subseteq V$ is a pseudo-clique if $\frac{2 \times |E[C]|}{|C| \times (|C|-1)} \geq \gamma$.

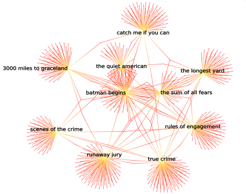
Takeaki Uno: An Efficient Algorithm for Solving Pseudo Clique Enumeration Problem. Algorithmica 2010

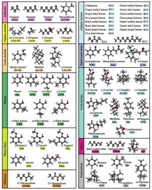
Question: Which Pb is the most difficult? Why?

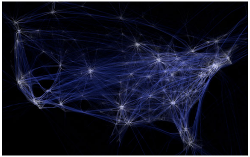
Outline

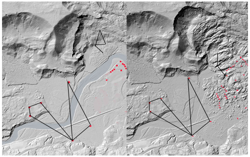
From Data to Augmented Graphs

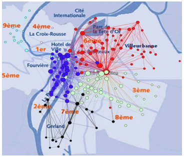












- Graphs are often **dynamic** with **attributes** related to **vertices** and/or **edges**.

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.

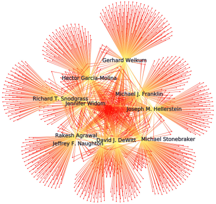
  A. Silva, W. Meira Jr., M. J. Zaki: Mining Attribute-structure Correlated Patterns in Large Attributed Graphs. PVLDB (2012)

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?

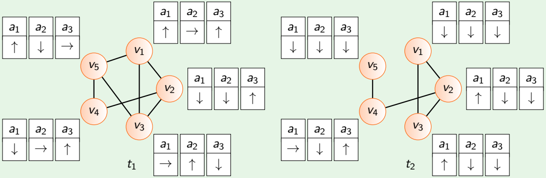
PRK$^+$ DEG$^+$ECML/PKDD$^+$	PRK$^+$ BETWEEN$^+$PVLDB$^+$
Christos Faloutsos	Gerhard Weikum
Jiawei Han	Jiawei Han
Philip S. Yu	David Maier
Bing Liu	Philip S. Yu
C. Lee Giles	Hector Garcia-Molina

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.

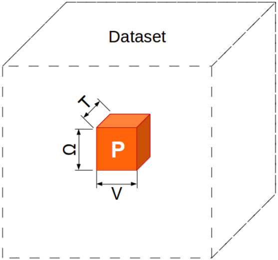
Example



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.

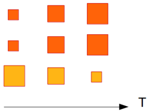


Predicates

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

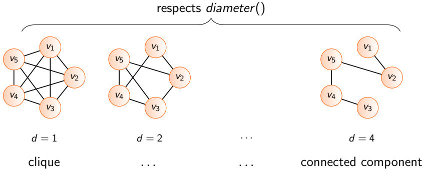
Constraint on the evolution:

- Makes sure attribute values co-evolve
- δ-**strictEvol**.
- $\forall v \in V$, $\forall t \in T$ and $\forall a^s \in \Omega$ then $\delta\text{-}trend(v, t, a) = s$



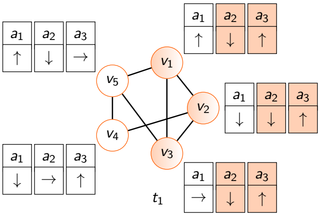
Constraint on the graph structure:

- Makes sure vertices are related through the graph structure.
- **diameter**.
- $\Delta\text{-}diameter(V, T, \Omega) = true \Leftrightarrow \forall t \in T \ diam_{G_t(V)} \leq \Delta$

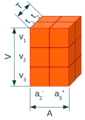


Example

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



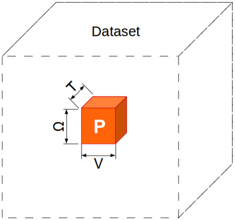
- 1-Diameter(P) is true,
- 0-strictEvol(P) is true.

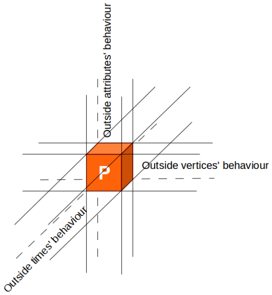


Density Measures

Intuition

Discard patterns that depict a behaviour supported by many other elements of the graph.
☛ **vertex specificity**, **temporal dynamic** and **trend relevancy**.

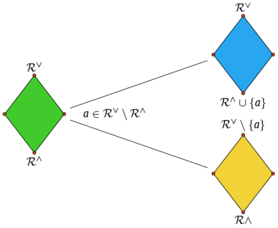




Algorithm

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)

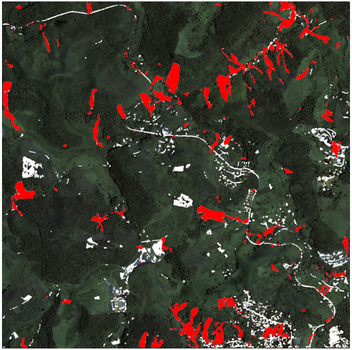
- 71 airports whose arrival delays increase over 3 weeks.
- *temporal_dynamic* $= 0$, which means that arrival delays never increased in these airports during another week.
- The hurricane strongly influenced the domestic flight organization.

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

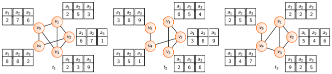


	$\lvert V\rvert$	$\lvert T\rvert$	$\lvert A\rvert$	density
Brazil landslide	10521	2	9	0.00057

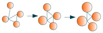
Discovering lanslides

- 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).

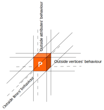
Overview



Co-evolution patterns



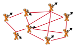
Interestingness Measures

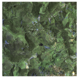


(Desmier et al., ECML/PKDD 2013)

Experimental results

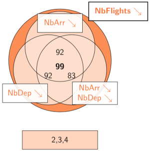
DBLP US flights Brazil landslides



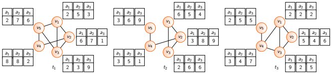


- Some obvious patterns are discarded ...
- ... but some patterns need to be generalized

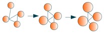
  Desmier et al, IDA 2014



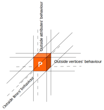
Overview



Co-evolution patterns



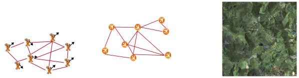
Interestingness Measures



(Desmier et al., ECML/PKDD 2013)

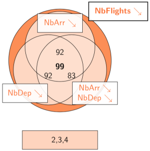
Experimental results

DBLP US flights Brazil landslides



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

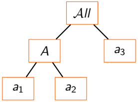
  Desmier et al, IDA 2014



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

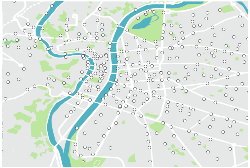
- How to have a good view of the demographic flows with only 2-point-trajectories?

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*

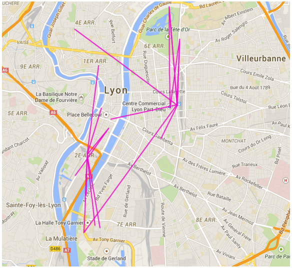
Example: The Velo'v network[1]



- 348 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.

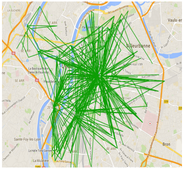
[1] http://www.velov.grandlyon.com/

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.

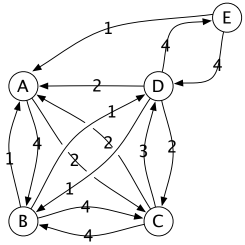
Operations on G_C

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

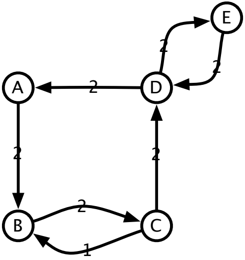
	Contexts			Trajectories
User	Gender	Age	Time	Travels
u_1	F	20	Day	(A,C), (B,A), (C,B)
u_1	F	20	Night	(D,C),(D,E),(E,A), (E,D)
u_2	M	23	Day	(A,B),(B,C),(C,A), (C,B)
u_2	M	23	Night	(A,B),(B,C),(C,B) (C,D),(D,C),(D,E), (E,D)
u_3	F	45	Day	(A,B),(B,C),(C,D), (D,A),(D,E),(E,D)
u_3	F	45	Night	(B,D),(D,B)
u_4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

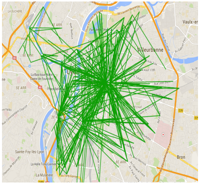
Mining Task:

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.

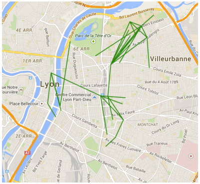
Algorithm in a nutshell

- 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;

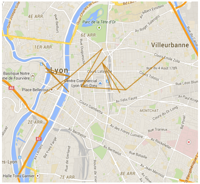
XP



(i) YoB ≥ 1962, CAT $=$ OURA



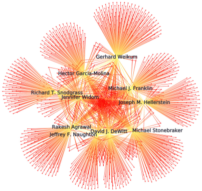
(ii) YoB ≥ 1980, TYP $=$ standard



(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 3^{rd} 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

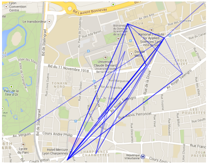


- What are the node attributes that strongly co-vary with the graph structure?
  - Co-authors that published at ICDE with a high degree and a low clustering coefficient.

  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.
- Integration of domain knowledge.
- Interactivity: replace the user in the center of the KDD process.
    - User preference learning
    - Inductive query recommendation

Thanks


YOANN PITARCH
BRUNO CRÉMILLEUX
PIERRE-NICOLAS MOUGEL
CÉLINE ROBARDET
ALBRECHT ZIMMERMANN
ADRIANA PRADO
CHEDY RAÏSSI
ARNAUD SOULET
JEAN-FRANÇOIS BOULICAUT
MEHDI KAYTOUE
LOÏC CERF
ELISE DESMIER
KIM-NGAN NGUYEN
CHRISTOPHE RIGOTTI