Preference-Based Pattern Mining

Marc Plantevit

Université Claude Bernard Lyon 1 – LIRIS CNRS UMR5205

* Slides from different research school lectures and conference tutorials.

eBISS 2018

About me.

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

Lab: **LIRIS UMR 5205**

Team: **Data Mining & Machine Learning (head since 2019)**

Research Interest: **Foundations of constraint-based pattern mining, sequences, augmented graphs, subgroup discovery, XAI.**

Evolution of Sciences

Before 1600: Empirical Science

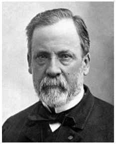
- Babylonian mathematics: 4 basis operations done with tablets and the resolution of practical problems based on words describing all the steps. $\Rightarrow$ that worked and they manage to solve 3 degree equations.
- Ancient Egypt: No theorization of algorithms. We give only examples made empirically, 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

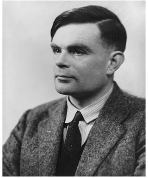
Each discipline has grown a theoretical component. Theoretical models often motivate experiments and generalize our understanding.

- 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.

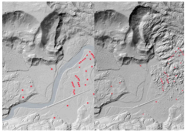
Evolution of Database Technology

- 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.

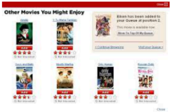
Why Data Mining?

- 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)
- ...

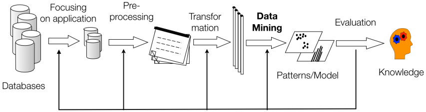
What is Data Mining

- 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

Data Mining

- Core of KDD
- Search for knowledge in data



Iterative and Interactive Process

Fayad et al., 1996

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.

ML versus DM

Predictive (global) modeling

- Turn the data into an as accurate as possible prediction machine.
- Ultimate purpose is **automatization**.
- E.g., autonomously driving a car based on sensor inputs

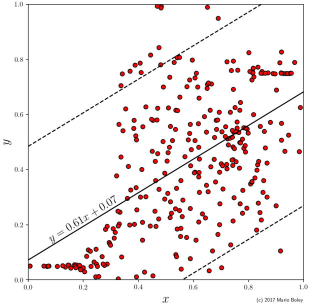
M. Boley www.realkd.org

Exploratory data analysis.

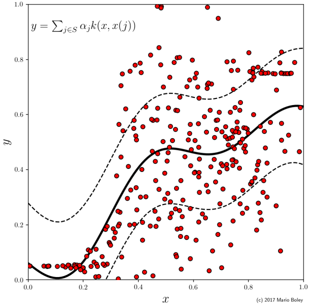
- Automatically discover novel insights about the domain in which the data was measured.
- Use machine discoveries to synergistically **boost** human expertise.
- E.g., understanding commonalities and differences among PET scans of Alzheimers patients.

ML versus DM

"A good prediction machine does not necessarily provide explicit insights into the data domains"



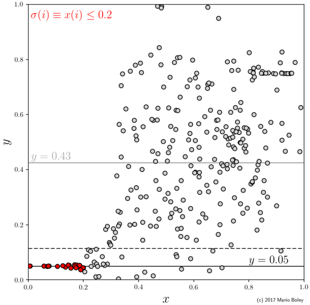
Global linear regression model

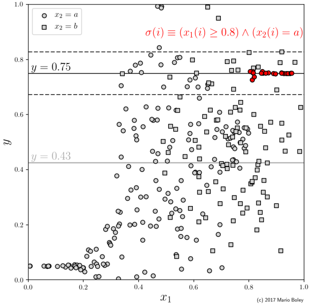


Gaussian process model.

ML versus DM

"A complex theory of everything might be of less value than a simple observation about a specific part of the data space"

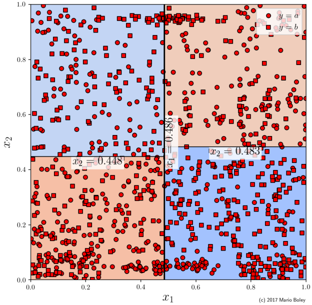


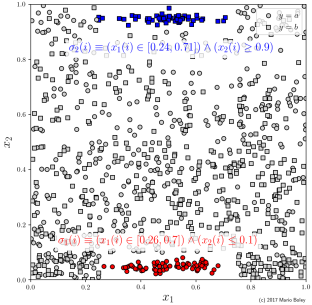


Identifying interesting subspace and the power of saying "I don't know for other points"

ML versus DM

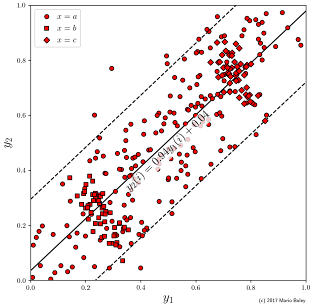
"Subgroups look similar to decision trees but good tree learners are forced to brush over some local structure in favor of the global picture"

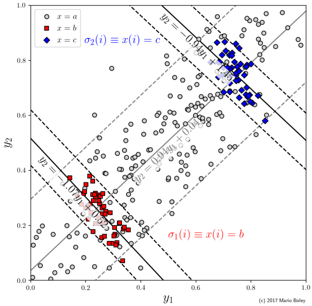




ML versus DM

"Going one step further, we can find local trends that are opposed to the global trend"





Roadmap

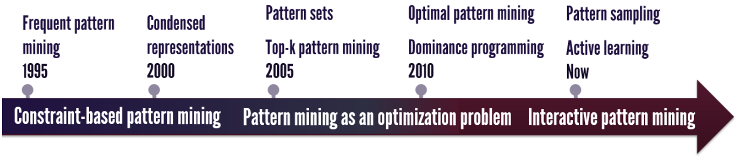
We will focus on **descriptive data mining** especially on Constraint-based Pattern Mining with an **inductive database vision**.

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

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

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

Roadmap



How have we moved from (only) frequent pattern discovery to interactive pattern mining?
How have we moved from the retrieval era to the exploratory analysis era?

Roadmap

- A short view on the constraint-based pattern mining toolbox and its limitation
  - Claim #1: this is not a tutorial on constraint-based pattern mining!

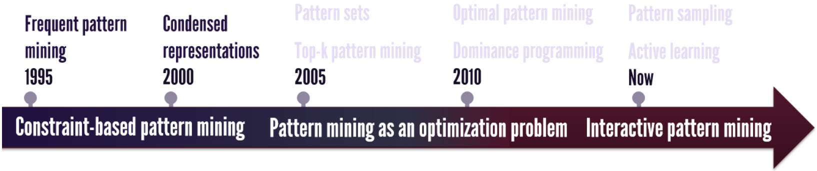
Roadmap

- A short view on the constraint-based pattern mining toolbox and its limitation
  - Claim #1: this is not a tutorial on constraint-based pattern mining!
- Pattern mining as an optimization problem based on user's preferences:
  - From all solutions to the optimal ones (top k, skyline, pattern set, etc.).
  - Claim #2: this is not a tutorial on preference learning!

Roadmap

- A short view on the constraint-based pattern mining toolbox and its limitation
  - Claim #1: this is not a tutorial on constraint-based pattern mining!
- Pattern mining as an optimization problem based on user's preferences:
  - From all solutions to the optimal ones (top k, skyline, pattern set, etc.).
  - Claim #2: this is not a tutorial on preference learning!
- Interactive pattern mining:
  - Dealing with implicit user's preferences.
  - How to ensure interactivity (instant mining, pattern space sampling)
  - Forgetting the completeness of the extraction.
  - Claim #3: this is not a tutorial on preference learning either!

- We have done some enlightenment choices.
  - Linearisation of the pattern mining research history.
- We are not exhaustive !
  - Feel free to mention us some important papers that are missing.
- Most of the examples will consider the itemsets as pattern language.
  - It is the simplest to convey the main ideas and intuitions.
- Feel free to interrupt us at any time if you have some questions.



Constraint-based pattern mining:

the toolbox and its limits

➥ the need of preferences in pattern mining

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 <u>itemset</u> 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 $d_{i,j} \in \{\text{true},\text{false}\}$

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.

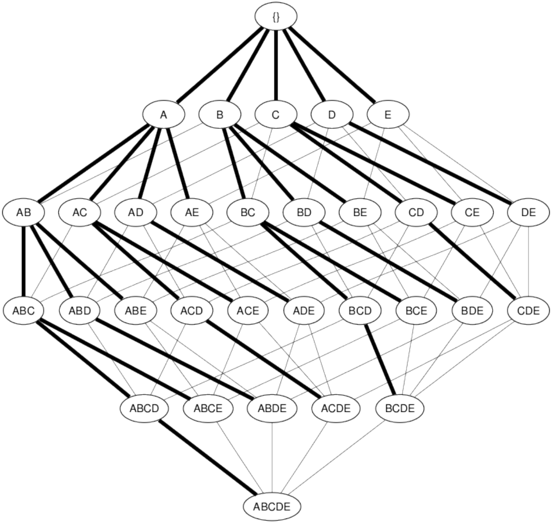
Naive algorithm

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? $2^{|\mathcal{A}|}$.

Prefix-based enumeration



Complexity of the naive approach

Question

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

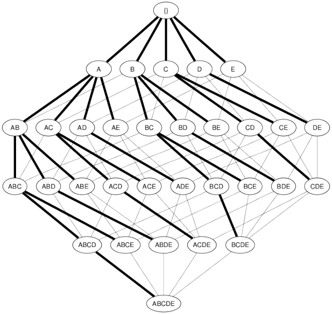
Question

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

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	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

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

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	$\times$	$\times$	$\times$
o_2	$\times$	$\times$	
o_3		$\times$	
o_4			$\times$

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

Anti-monotonicity of the frequency

Corollary

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

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

	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

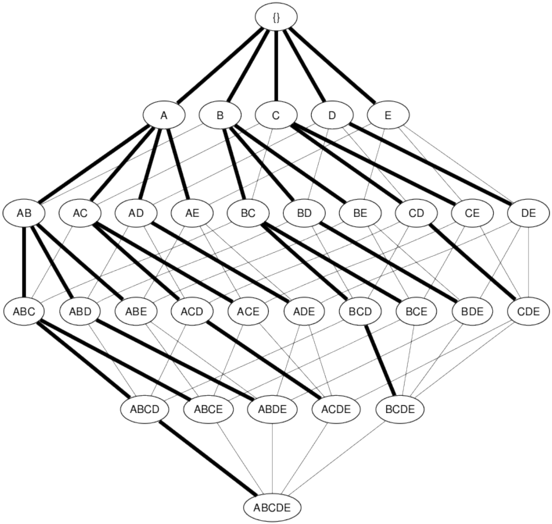
Given a dataset $\mathcal{D}$ of objects described with Boolean attributes in $\mathcal{A}$ and a minimal frequency $\mu \in \mathbb{N}$:

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

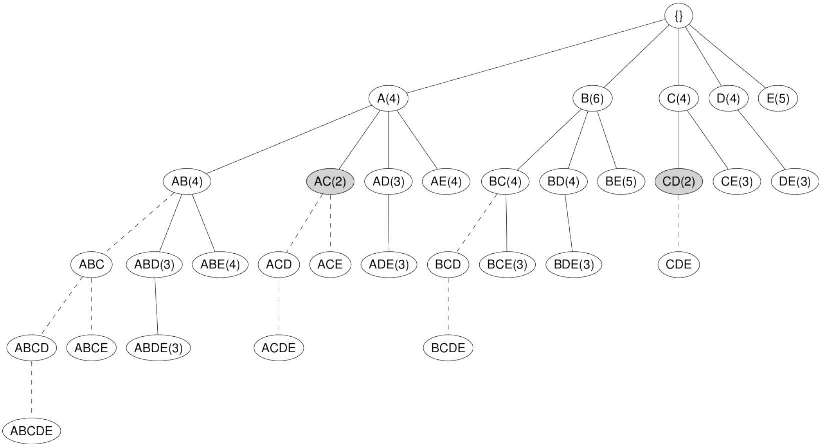
	a_1	a_2	a_3
o_1	×	×	×
o_2	×	×	
o_3		×	
o_4			×

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

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



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



APriori enumeration

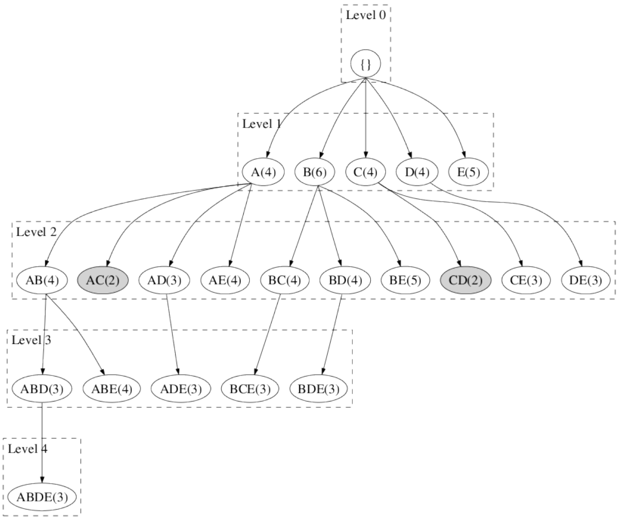
To check the frequency of every parent, the enumeration tree must be traversed breadth-first.

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



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

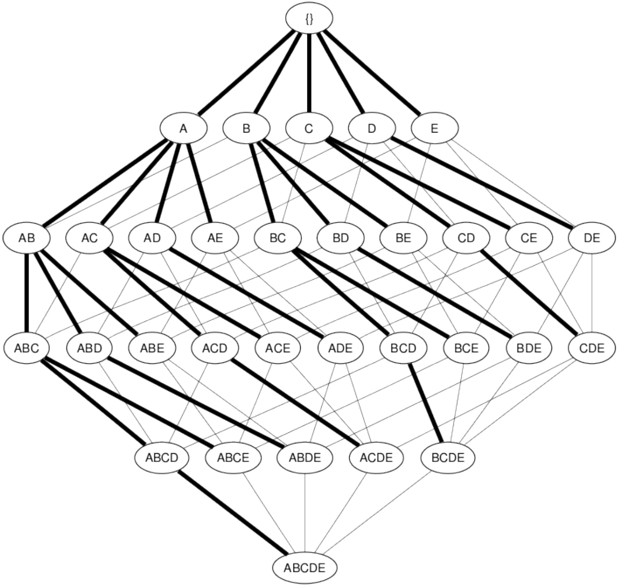
itemset
{2 3 5}

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

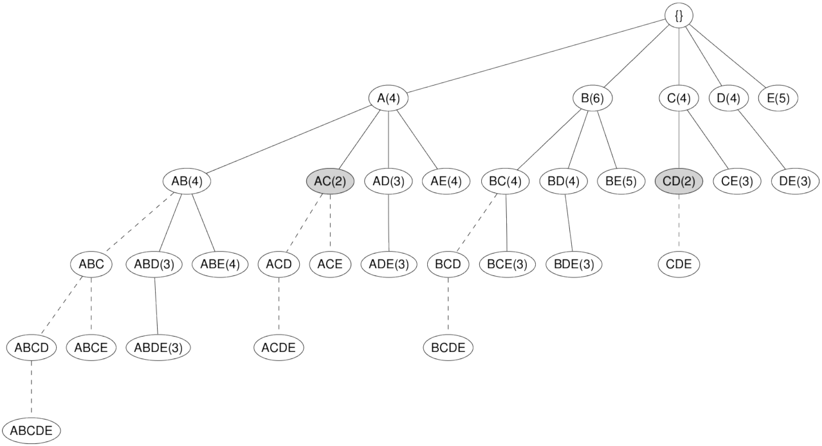
Observation

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	×	×	×
o_2	×	×	
o_3		×	
o_4			×

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

Vertical representation of the data

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

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

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

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

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

where $i_j \subseteq \mathcal{O}$

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

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

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

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

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

where $i_j \subseteq \mathcal{O}$

Unless the minimal relative frequency is very low, storing the support on <u>bitsets</u> 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

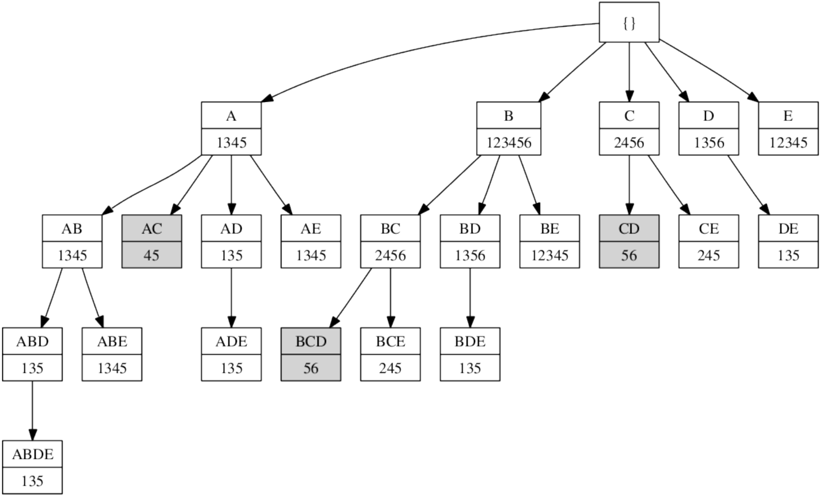
Like APriori:

- 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 ($\mu = 3$)



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: $\{\}, \{a_1, a_2, a_3, a_4, a_5\}, \{a_6, a_7, a_8, a_9, a_{10}\}, \{a_{11}, a_{12}, a_{13}, a_{14}, a_{15}\}$.

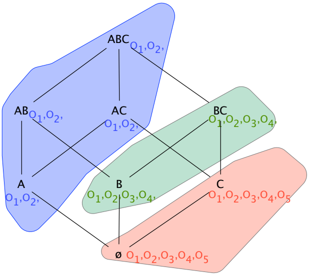
☞ the need to focus on a **condensed representation** of frequent patterns.

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.

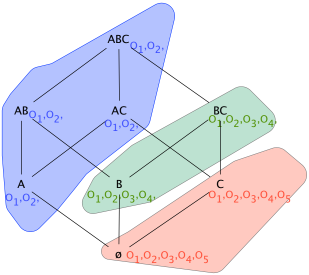
$\mathcal{O}$	A	B	C
o_1	×	×	×
o_2	×	×	×
o_3		×	×
o_4		×	×
o_5			×



Closed and Free Patterns

Equivalence classes based on support.

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



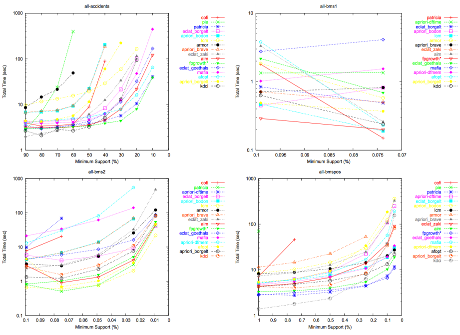
- **Closed** patterns are maximal element of each equivalence class: ABC, BC, and C.
- **Generators** or **Free** patterns are minimal elements (not necessary unique) of each equivalent class: $\{\}, A$ and B

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

Few researchers (in DM) are aware about this strong intersection.

A strong intersection with Formal Concept Analysis (Ganter and Wille, 1999).

- transactional DB $\equiv$ **formal context** is a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is a binary relation called incidence that expresses which objects have which attributes.
- closed itemset $\equiv$ **concept intent**
- FCA gives the mathematical background about closed patterns.
- Algorithms: **LCM** is an efficient implementation of **Close By One.** (Sergei O. Kuznetsov, 1993).



(FIMI Workshop@ICDM, 2003 and 2004)

The FIM Era: during more than a decade, only ms were worth it!

Even if the complete collection of frequent itemsets is known useless, the main objective of many algorithms is to earn ms according to their competitors!!

What about the end-user (and the pattern interestingness)?

➜ partially answered with constraints.

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

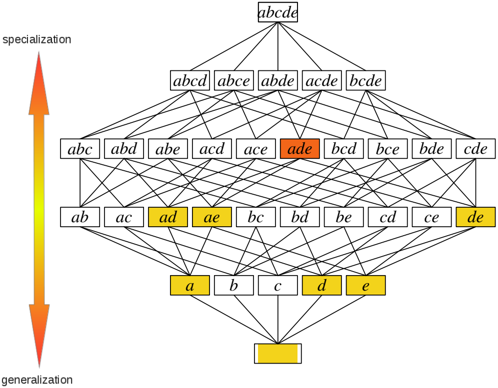
Constraint properties are used to infer constraint values on (many) patterns without having to evaluate them individually.

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

Constraint properties - 1

Monotone constraint

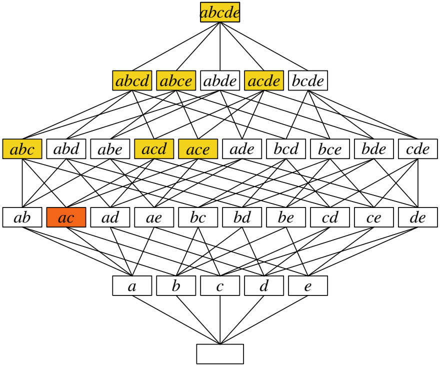
$\forall \varphi_1 \preceq \varphi_2,\ \mathcal{C}(\varphi_1, \mathcal{D}) \Rightarrow \mathcal{C}(\varphi_2, \mathcal{D})$



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

Anti-monotone constraint

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

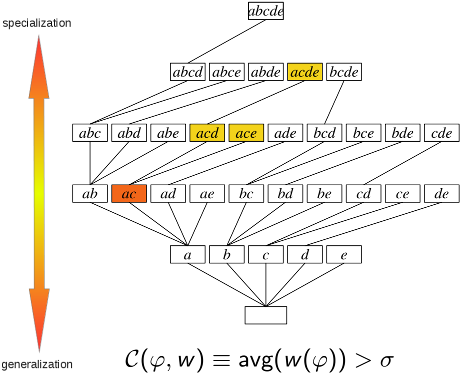


$$\mathcal{C}(\varphi, \mathcal{D}) \equiv a \notin \varphi \wedge c \notin \varphi$$

Constraint properties - 2

Convertible constraints

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



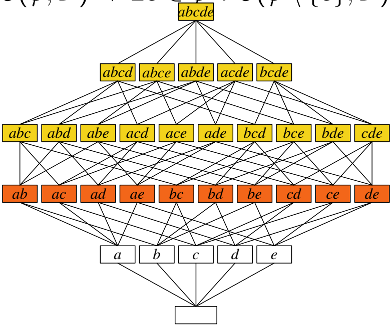
$$\mathcal{C}(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma$$

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

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

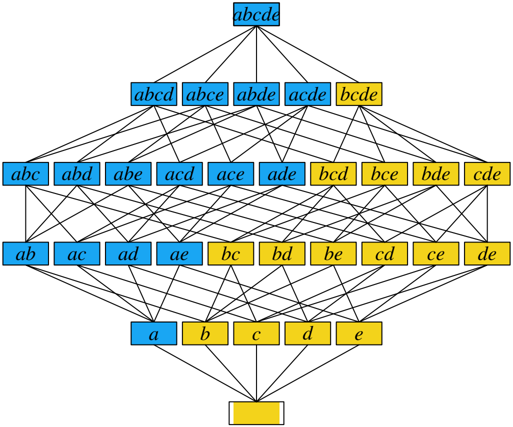
Examples

$v \in P$	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

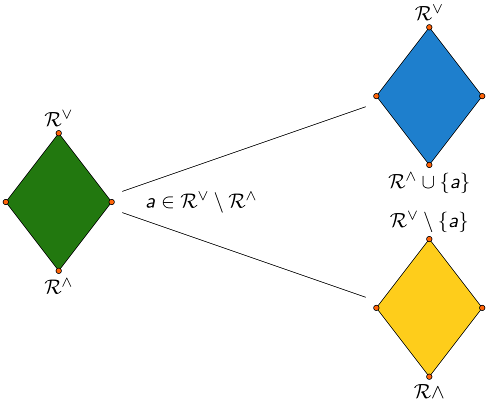
Enumeration strategy

Binary partition: the element 'a' is enumerated



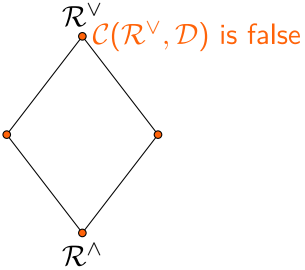
Enumeration strategy

Binary partition: the element 'a' is enumerated



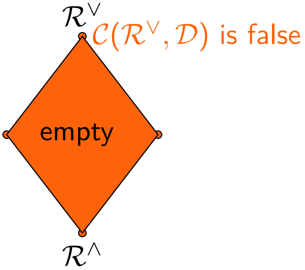
Constraint evaluation

Monotone constraint



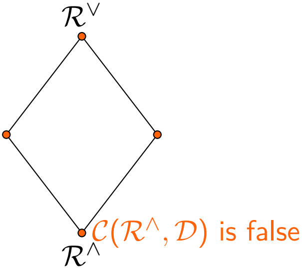
Constraint evaluation

Monotone constraint



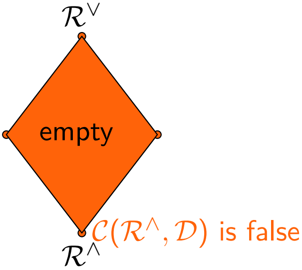
Constraint evaluation

Anti-monotone constraint



Constraint evaluation

Anti-monotone constraint



A new class of constraints

Piecewise monotone and anti-monotone constraints[1]

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:

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

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

[1] 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)

An example

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

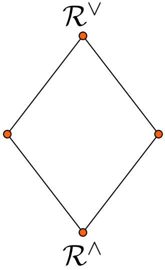
$\mathcal{C}(\varphi, \mathcal{D})$ is piecewise monotone and anti-monotone with

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

$\forall x \preceq y$,

- $f_{1,\varphi}$ 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



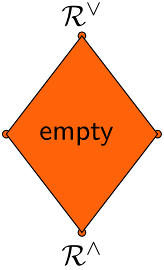
Evaluation

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

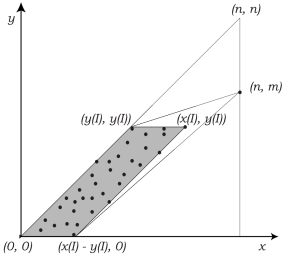
Function `Generic_CBPM_enumeration`$(\mathcal{R}^\vee, \mathcal{R}^\wedge)$

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

Tight Upper-bound computation

- 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



Studying constraints $\equiv$ looking for efficient and effective upper bound in a branch and bound algorithm !

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]
- Pairwise change behavior [ECMLPKDD'17]
- Exceptional attributed Graphs [Machine Learning 2017, ICDM'16, ComplexNetwork17]

Toward declarativity

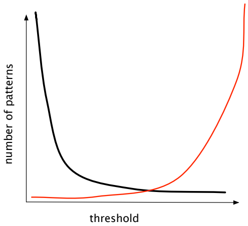
Why declarative approaches?

- for each problem, do not write a solution from scratch

Declarative approaches:

- CP approaches (Khiari et al., CP10, Guns et al., TKDE 2013)
- SAT approaches (Boudane et al., IJCAI16, Jabbour et al., CIKM13)
- ILP approaches (Mueller et al, DS10, Babaki et al., CPAIOR14, Ouali et al. IJCAI16)
- ASP approaches (Gebser et al., IJCAI16)

Thresholding problem

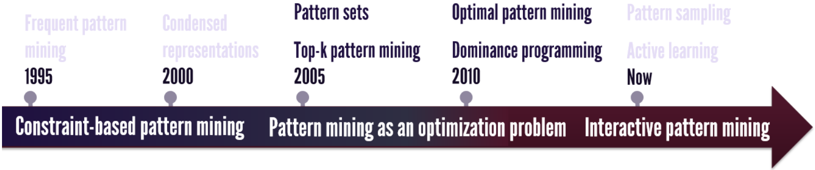


- A too stringent threshold: trivial patterns
- A too weak threshold: too many patterns, unmanageable and diversity not necessary ensured.
- Some attempts to tackle this issue:
  - Interestingness is not a dichotomy! [BB05]
  - Taking benefit from hierarchical relationships [HF99, DPRB14]
- But setting thresholds remains an issue in pattern mining.

Constraint-based pattern mining:

concluding remarks

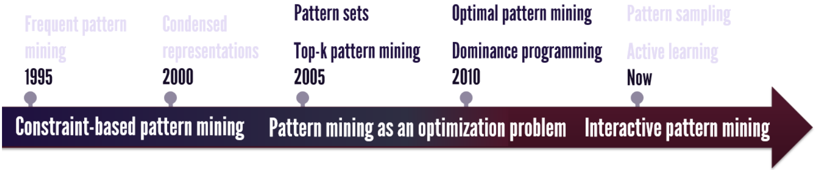
- how to fix thresholds?
- how to handle numerous patterns including non-informative patterns? how to get a global picture of the set of patterns?
- how to design the proper constraints/preferences?



Pattern mining as an optimization problem

Pattern mining

as an optimization problem



- performance issue
- the more, the better
- data-driven

- quality issue
- the less, the better
- user-driven

In this part:

- preferences to express user's interests
- focusing on the best patterns: dominance relation, optimal pattern sets, subjective interest

Addressing pattern mining tasks

with user preferences

Idea: a preference expresses a user's interest (no required threshold)

Examples based on measures/dominance relation:

- *"the higher the frequency, growth rate and aromaticity are, the better the patterns"*
- *"I prefer pattern X_1 to pattern X_2 if X_1 is not dominated by X_2 according to a set of measures"*

➥ measures/preferences: a natural criterion for ranking patterns and presenting the "best" patterns

Preference-based approaches

in this tutorial

- **in this part**: preferences are explicit (typically given by the user depending on his/her interest/subjectivity)

  in the last part: preferences are implicit

- *quantitative/qualitative preferences:*
  - **quantitative:**

    $$\textit{measures} \begin{cases} \textit{constraint-based data mining}: \textit{frequency}, \textit{size}, \ldots \\ \textit{background knowledge}: \textit{price}, \textit{weight}, \textit{aromaticity}, \ldots \\ \textit{statistics}: \textit{entropy}, \textit{pvalue}, \ldots \end{cases}$$

  - **qualitative:** "I prefer pattern X_1 to pattern X_2" (pairwise comparison between patterns).

    With qualitative preferences: two patterns can be incomparable.

Measures

Many works on:

- interestingness measures (Geng et al. ACM Computing Surveys06)
- utility functions (Yao and Hamilton DKE06)
- statistically significant rules (Hämäläinen and Nykänen ICDM08)

Examples:

- $area(X) = frequency(X) \times size(X)$ (tiling: surface)
- $lift(X_1 \rightarrow X_2) = \frac{\mathcal{D} \times frequency(X_1 X_2)}{frequency(X_2) \times frequency(X_1)}$
- *utility functions:* utility of the mined patterns (e.g. weighted items, weighted transactions).
  An example: No of Product $\times$ Product profit

Putting the pattern mining task to
an optimization problem

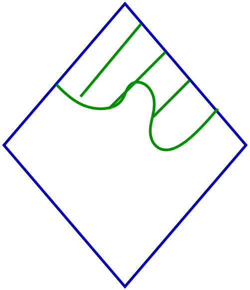
The most interesting patterns according to measures/preferences:

- **free/closed patterns** (Boulicaut et al. DAMI03, Bastide et al. SIGKDD Explorations00)
  ➥ given an equivalent class, I prefer the shortest/longest patterns
- **one measure**: top-k patterns (Fu et al. Ismis00, Jabbour et al. ECML/PKDD13)
- **several measures**: how to find a trade-off between several criteria?
  ➥ skyline patterns (Cho et al. IJDWM05, Soulet et al. ICDM'11, van Leeuwen and Ukkonen ECML/PKDD13)
- **dominance programming** (Negrevergne et al. ICDM13), **optimal patterns** (Ugarte et al. ICTAI15)
- **subjective interest/interest according to a background knowledge** (De Bie DAMI2011)

top-k pattern mining: an example

Goal: finding the k patterns maximizing an interestingness measure.

Tid	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



- the 3 most frequent patterns: B, E, BE[a]

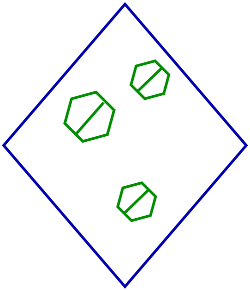
  ➥ easy due to the anti-monotone property of frequency

[a]Other patterns have a frequency of 5: C, D, BC, BD, CD, BCD

top-k pattern mining: an example

Goal: finding the k patterns maximizing an interestingness measure.

Tid	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



- the 3 most frequent patterns: *B*, *E*, *BE*[a]
  ➥ easy due to the anti-monotone property of frequency
- the 3 patterns maximizing area: *BCDE*, *BCD*, *CDE*
  ➥ branch & bound
  (Zimmermann and De Raedt MLJ09)

[a]Other patterns have a frequency of 5: *C*, *D*, *BC*, *BD*, *CD*, *BCD*

top-k pattern mining

an example of pruning condition

top-k patterns according to *area*, $k = 3$

Tid	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F

Principle:

- $\mathcal{C}and$: the current set of the k best candidate patterns
- when a candidate pattern is inserted in $\mathcal{C}and$, a more efficient pruning condition is deduced

A: lowest value of *area* for the patterns in $\mathcal{C}and$

L: size of the longest transaction in $\mathcal{D}$ (here: $L = 6$)

a pattern X must satisfy $frequency(X) \geq \frac{A}{L}$ to be inserted in $\mathcal{C}and$
➥ pruning condition according to the frequency (thus anti-monotone)

Example with a depth first search approach:

- initialization: $\mathcal{C}and = \{B, BE, BEC\}$
  ($area(BEC) = 12$, $area(BE) = 10$, $area(B) = 6$)
  ➥ $frequency(X) \geq \frac{6}{6}$
- new candidate $BECD$: $\mathcal{C}and = \{BE, BEC, BECD\}$
  ($area(BECD) = 16$, $area(BEC) = 12$, $area(BE) = 10$)
  ➥ $frequency(X) \geq \frac{10}{6}$ which is more efficient than $frequency(X) \geq \frac{6}{6}$
- new candidate $BECDF$...

top-k pattern mining in a nutshell

Advantages:

- compact
- threshold free
- best patterns

Drawbacks:

- complete resolution is costly, sometimes heuristic search (beam search)
  (van Leeuwen and Knobbe DAMI12)
- diversity issue: top-k patterns are often very similar
- several criteria must be aggregated

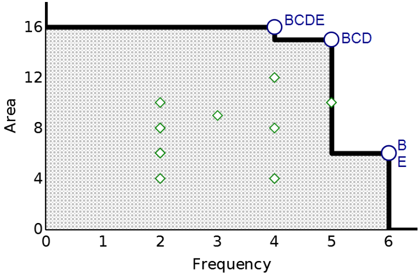
➥ skylines patterns: a trade-off between several criteria

Skypatterns (Pareto dominance)

Notion of skylines (database) in pattern mining (Cho at al. IJDWM05, Papadopoulos et al. DAMI08, Soulet et al. ICDM11, van Leeuwen and Ukkonen ECML/PKDD13)

Tid	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F

Patterns	**freq**	**area**
~~*AB*~~	2	4
~~*AEF*~~	2	6
B	6	6
BCDE	4	16
~~*CDEF*~~	2	8
E	6	6
⋮	⋮	⋮



$|\mathcal{L}_{\mathcal{I}}| = 2^6$, but only 4 skypatterns

$$Sky(\mathcal{L}_{\mathcal{I}}, \{freq, area\}) = \{BCDE, BCD, B, E\}$$

Skylines vs skypatterns

Problem	Skylines	Skypatterns
Mining task	a set of non dominated transactions	a set of non dominated patterns
Size of the space search	$\mid \mathcal{D} \mid$	$\mid \mathcal{L} \mid$
domain	a lot of works	very few works

usually: $\mid \mathcal{D} \mid << \mid \mathcal{L} \mid$

$\mathcal{D}$	set of transactions
$\mathcal{L}$	set of patterns

Skypatterns: how to process?

A naive enumeration of all candidate patterns ($\mathcal{L}_\mathcal{I}$) and then comparing them is not feasible...

Two approaches:

1. take benefit from the pattern condensed representation according to the condensable measures of the given set of measures M
   - skylineability to obtain M' ($M' \subseteq M$) giving a more concise pattern condensed representation
   - the pattern condensed representation w.r.t. M' is a superset of the representative skypatterns w.r.t. M which is (much smaller) than $\mathcal{L}_\mathcal{I}$.

2. use of the dominance programming framework (together with skylineability)

Dominance programming

Dominance: a pattern is optimal if it is not dominated by another.
Skypatterns: dominance relation = Pareto dominance

1. **Principle:**
   - starting from an initial pattern s_1
   - searching for a pattern s_2 such that s_1 is not preferred to s_2
   - searching for a pattern s_3 such that s_1 and s_2 are not preferred to s_3
     $\vdots$
   - until there is no pattern satisfying the whole set of constraints

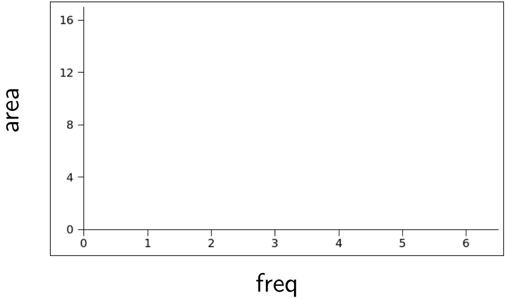
2. **Solving:**
   - constraints are dynamically posted during the mining step

Principle: increasingly reduce the dominance area by processing pairwise comparisons between patterns. Methods using Dynamic CSP (Negrevergne et al. ICDM13, Ugarte et al. CPAIOR14, AIJ 2017).

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$$M = \{freq, area\}$$

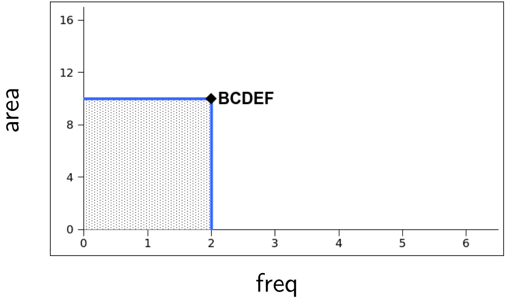
$$q(X) \equiv closed_{M'}(X)$$

$$Candidates =$$

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$$M = \{freq, area\}$$

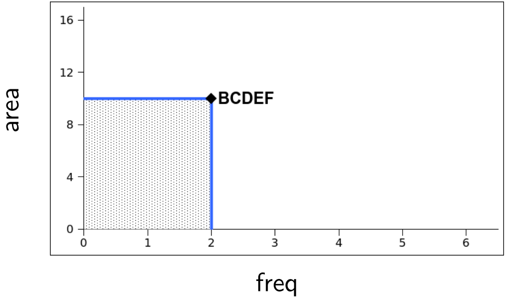
$$q(X) \equiv closed_{M'}(X)$$

$$Candidates = \{\underbrace{BCDEF}_{s_1},$$

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$$M = \{freq, area\}$$

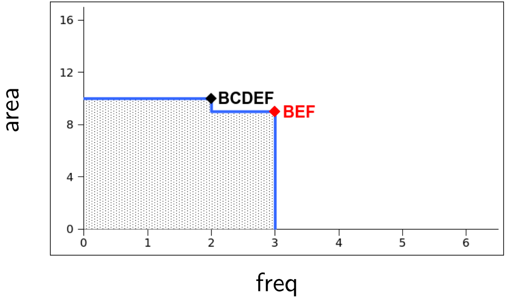
$$q(X) \equiv closed_{M'}(X) \wedge \neg(s_1 \succ_M X)$$

$$Candidates = \{\underbrace{BCDEF}_{s_1},$$

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$$M = \{freq, area\}$$

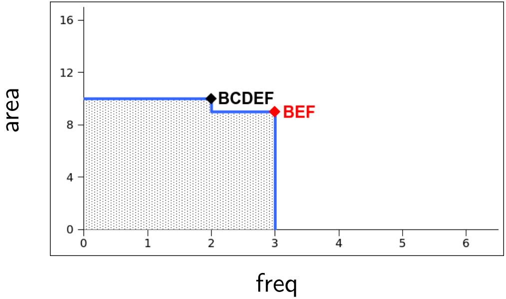
$$q(X) \equiv closed_{M'}(X) \wedge \neg(s_1 \succ_M X)$$

$$Candidates = \{\underbrace{BCDEF}_{s_1}, \underbrace{BEF}_{s_2},$$

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$$M = \{freq, area\}$$

$$q(X) \equiv closed_{M'}(X) \wedge \neg(s_1 \succ_M X) \wedge \neg(s_2 \succ_M X)$$

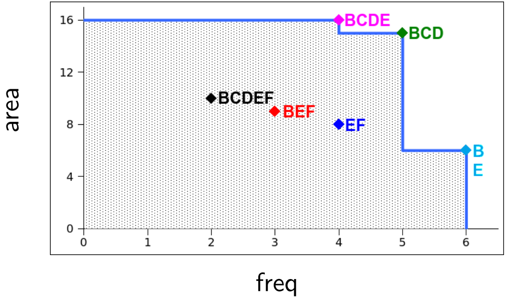
$$Candidates = \{\underbrace{\text{BCDEF}}_{s_1}, \underbrace{\text{BEF}}_{s_2},$$

Dominance programming:

example of the skypatterns

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F

$|\mathcal{L}_\mathcal{I}| = 2^6 = 64$ patterns
4 skypatterns



$$M = \{freq, area\}$$

$$q(X) \equiv closed_{M'}(X) \wedge \neg(s_1 \succ_M X) \wedge \neg(s_2 \succ_M X) \wedge \neg(s_3 \succ_M X) \wedge \neg(s_4 \succ_M X) \wedge \neg(s_5 \succ_M X) \wedge \neg(s_6 \succ_M X) \wedge \neg(s_7 \succ_M X)$$

$$Candidates = \{\underbrace{\cancel{BCDEF}}_{s_1}, \underbrace{\cancel{BEF}}_{s_2}, \underbrace{\cancel{EF}}_{s_3}, \underbrace{\underbrace{BCDE}_{s_4}, \underbrace{BCD}_{s_5}, \underbrace{B}_{s_6}, \underbrace{E}_{s_7}}_{Sky(\mathcal{L}_\mathcal{I}, M)}\}$$

Dominance programming: to sum up

The dominance programming framework encompasses many kinds of patterns:

	dominance relation
maximal patterns	inclusion
closed patterns	inclusion at same frequency
top-k patterns	order induced by the interestingness measure
skypatterns	Pareto dominance

maximal patterns $\subseteq$ closed patterns

top-k patterns $\subseteq$ skypatterns

A step further

a preference is defined by any property between two patterns (i.e., pairwise comparison) and not only the Pareto dominance relation: measures on a set of patterns, overlapping between patterns, coverage,. . .

➥ preference-based optimal patterns

In the following:
(1) define preference-based optimal patterns,
(2) show how many tasks of local patterns fall into this framework,
(3) deal with optimal pattern sets.

Preference-based optimal patterns

A preference $\rhd$ is a strict partial order relation on a set of patterns $\mathbb{S}$.

$x \rhd y$ indicates that x is preferred to y

(Ugarte et al. ICTAI15): a pattern x is optimal (OP) according to $\rhd$ iff $\nexists y_1, \ldots y_p \in \mathbb{S}, \forall 1 \leq j \leq p,\ y_j \rhd x$

(a single y is enough for many data mining tasks)

Characterisation of a set of OPs: a set of patterns:

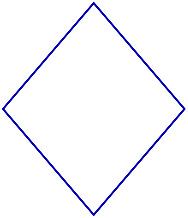
$$\left\{ x \in \mathbb{S} \mid \ \mathtt{fundamental}(x) \ \wedge \nexists y_1, \ldots y_p \in \mathbb{S}, \forall 1 \leq j \leq p,\ y_j \rhd x \ \right\}$$

$\mathtt{fundamental}(x)$: x must satisfy a property defined by the user

for example: having a minimal frequency, being closed, ...

Local patterns: examples

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$\mathbb{S} = \mathcal{L}_{\mathcal{I}}$

(Mannila et al. DAMI97)

Large tiles

$\mathrm{c}(x) \equiv \mathit{freq}(x) \times \mathtt{size}(x) \geq \psi_{area}$

Example: $\mathit{freq}(BCD) \times \mathtt{size}(BCD) = 5 \times 3 = 15$

Frequent sub-groups

$$\mathrm{c}(x) \equiv \quad \mathit{freq}(x) \geq \psi_{freq} \wedge \nexists\, y \in \mathbb{S} : T_1(y) \supseteq T_1(x) \wedge T_2(y) \subseteq T_2(x) \wedge (T(y) = T(x) \Rightarrow y \subset x)$$

Skypatterns

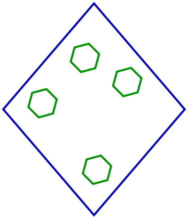
$$\mathrm{c}(x) \equiv \quad \mathtt{closed}_M(x) \wedge \nexists\, y \in \mathbb{S} : y \succ_M x$$

Frequent top-k patterns according to m

$$\mathrm{c}(x) \equiv \quad \mathit{freq}(x) \geq \psi_{freq} \wedge \nexists\, y_1, \ldots, y_k \in \mathbb{S} : \bigwedge_{1 \leq j \leq k} \mathtt{m}(y_j) > \mathtt{m}(x)$$

Local (optimal) patterns: examples

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F



$\mathbb{S} = \mathcal{L}_{\mathcal{I}}$

(Mannila et al. DAMI97)

Large tiles

$c(x) \equiv freq(x) \times \texttt{size}(x) \geq \psi_{area}$

Frequent sub-groups

$$c(x) \equiv \quad freq(x) \geq \psi_{freq} \ \wedge \nexists\, y \in \mathbb{S} : T_1(y) \supseteq T_1(x) \wedge T_2(y) \subseteq T_2(x) \wedge (T(y) = T(x) \Rightarrow y \subset x)$$

Skypatterns

$$c(x) \equiv \quad \texttt{closed}_M(x) \wedge \nexists\, y \in \mathbb{S} : y \succ_M x$$

Frequent top-k patterns according to m

$$c(x) \equiv \quad freq(x) \geq \psi_{freq} \wedge \nexists\, y_1, \ldots, y_k \in \mathbb{S} : \bigwedge_{1 \leq j \leq k} \mathtt{m}(y_j) > \mathtt{m}(x)$$

Pattern sets: sets of patterns

Patterns sets (De Raedt and Zimmermann SDM07): sets of patterns satisfying a global viewpoint (instead of evaluating and selecting patterns based on their individual merits)

Search space ($\mathbb{S}$): local patterns versus pattern sets

example: $\mathcal{I} = \{A, B\}$

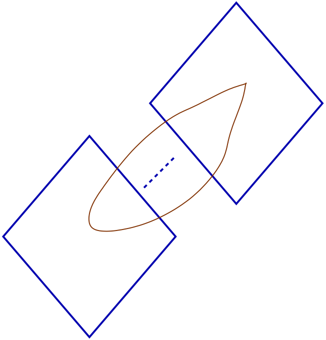
- all local patterns: $\mathbb{S} = \mathcal{L}_{\mathcal{I}} = \{\emptyset, A, B, AB\}$
- all pattern sets:
  $\mathbb{S} = 2^{\mathcal{L}_{\mathcal{I}}} = \{\emptyset, \{A\}, \{B\}, \{AB\}, \{A, B\}, \{A, AB\}, \{B, AB\}, \{A, B, AB\}\}$

Many data mining tasks: classification (Liu et al. KDD98), clustering (Ester et al. KDD96), database tiling (Geerts et al. DS04), pattern summarization (Xin et al. KDD06), pattern teams (Knobbe and Ho PKDD06),...

Many input ("preferences") can be given by the user:
coverage, overlapping between patterns, syntactical properties, measures, number of local patterns,...

Coming back on OP (Ugarte et al. ICTAI15)

Pattern sets of length k: examples



$\mathbb{S} \subset 2^{\mathcal{L}_{\mathcal{I}}}$

(sets of length k)

Conceptual clustering (without overlapping)

$$\mathtt{clus}(x) \equiv \bigwedge_{i \in [1..k]} \mathtt{closed}(x_i) \wedge \bigcup_{i \in [1..k]} \mathtt{T}(x_i) = \mathcal{T} \wedge \bigwedge_{i,j \in [1..k]} \mathtt{T}(x_i) \cap \mathtt{T}(x_j) = \emptyset$$

Conceptual clustering with optimisation

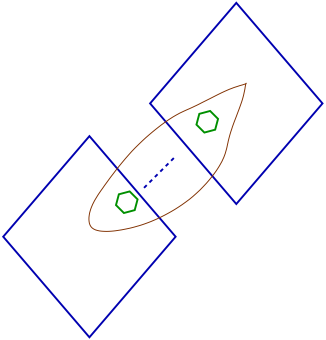
$$\mathtt{c}(x) \equiv \mathtt{clus}(x) \wedge \nexists\, y \in 2^{\mathcal{L}_{\mathcal{I}}}, \min_{j \in [1..k]} \{freq(y_j)\} > \min_{i \in [1..k]} \{freq(x_i)\}$$

Pattern teams

$$\mathtt{c}(x) \equiv \mathtt{size}(x) = k \wedge \nexists\, y \in 2^{\mathcal{L}_{\mathcal{I}}}, \Phi(y) > \Phi(x)$$

Coming back on OP (Ugarte et al. ICTAI15)

(Optimal) pattern sets of length k: examples



$\mathbb{S} \subset 2^{\mathcal{L}_\mathcal{I}}$

(sets of length k)

Conceptual clustering (without overlapping)

$$\texttt{clus}(x) \equiv \bigwedge_{i\in[1..k]} \texttt{closed}(x_i) \wedge \bigcup_{i\in[1..k]} \texttt{T}(x_i) = \mathcal{T} \wedge \bigwedge_{i,j\in[1..k]} \texttt{T}(x_i) \cap \texttt{T}(x_j) = \emptyset$$

Conceptual clustering with optimisation

$$\texttt{c}(x) \equiv \texttt{clus}(x) \wedge \nexists\, y \in 2^{\mathcal{L}_\mathcal{I}}, \min_{j\in[1..k]}\{freq(y_j)\} > \min_{i\in[1..k]}\{freq(x_i)\}$$

Pattern teams

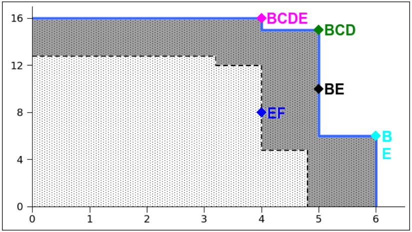
$$\texttt{c}(x) \equiv \texttt{size}(x) = k \wedge \nexists\, y \in 2^{\mathcal{L}_\mathcal{I}}, \Phi(y) > \Phi(x)$$

Relax the dogma "must be optimal":

soft patterns

Stringent aspect of the classical constraint-based pattern mining framework: *what about a pattern which slightly violates a query?*

example: introducing softness in the skypattern mining:
➥ soft-skypatterns



put the user in the loop to determine the best patterns w.r.t. his/her preferences

Introducing softness is easy with Constraint Programming:
➥ same process: it is enough to update the posted constraints

Many other works in this broad field

Example: heuristic approaches

pattern sets based on the Minimum Description Length principle: a small set of patterns that compress - KRIMP (Siebes et al. SDM06)

$L(D, CT)$: the total compressed size of the encoded database and the code table:

$$L(D, CT) = L(D|CT) + L(CT|D)$$

Many usages:

- characterizing the differences and the norm between given components in the data - DIFFNORM (Budhathoki and Vreeken ECML/PKDD15)
- causal discovery (Budhathoki and Vreeken ICDM16)
- missing values (Vreeken and Siebes ICDM08)
- handling sequences (Bertens et al. KDD16)
- ...

and many other works on data compression/summarization (e.g. Kiernan and Terzi KDD08),...

Nice results based on the frequency. How handling other measures?

Subjective interestingness

$$SI = \frac{IC}{DL} = \frac{Information\ content}{Assimilation\ cost}$$

The idea: the user as part of the process, he/she states expectations/beliefs, e.g.: number of items bought by customers, popularity of items, overall graph density (in dense subgraph mining)

➥ whatever contrasts with this = subjectively interesting

- producing a set of patterns: the background distribution is updated according to the patterns previously extracted
- iterative approach: at each step, the best pattern according the interestingness criterion is extracted (trade off between information content and descriptional complexity)

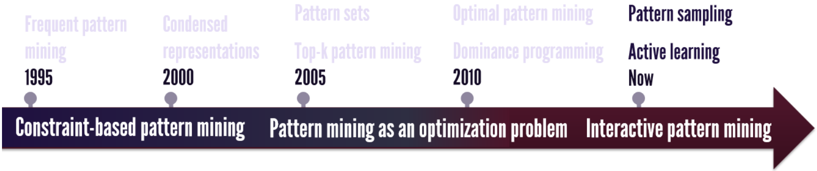
(Gallo et al. ECML/PKDD07, De Bie DAMI11, De Bie IDA13, van Leeuwen et al. MLJ16)

Pattern mining as an optimization problem: concluding remarks

In the approaches indicated in this part:

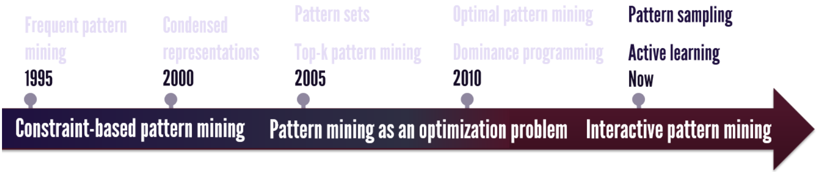
- measures/preferences are explicit and must be given by the user. . . (but there is no threshold :-)
- diversity issue: top-k patterns are often very similar
- complete approaches (optimal w.r.t the preferences):
  ➥ stop completeness "Please, please stop making new algorithms for mining *all* patterns"
  Toon Calders (ECML/PKDD 2012, most influential paper award)

A further step: interactive pattern mining (including the instant data mining challenge), implicit preferences and learning preferences



Interactive pattern mining

Interactive pattern mining



Idea: *"I don't know what I am looking for, but I would definitely know if I see it."*

➠ preference acquisition

In this part:

- Easier: no user-specified parameters (constraint, threshold or measure)!
- Better: learn user preferences from user feedback
- Faster: instant pattern discovery

Addressing pattern mining

with user interactivity

Advanced Information Retrieval-inspired techniques

- Query by Example in information retrieval (QEIR) (Chia et al. SIGIR08)
- Active feedback with Information Retrieval (Shen et al. SIGIR05)
- SVM Rank (Joachims KDD02)
- ...

Challenge: pattern space $\mathcal{L}$ is often much larger than the dataset $\mathcal{D}$

Interactive pattern mining: overview

Interactive data exploration using pattern mining. (van Leeuwen 2014)

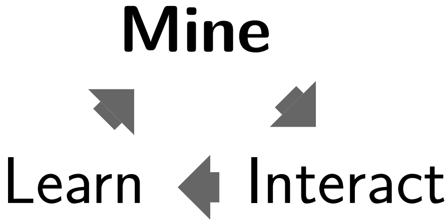
Mine

Learn

Interact

Interactive pattern mining: overview

Interactive data exploration using pattern mining. (van Leeuwen 2014)

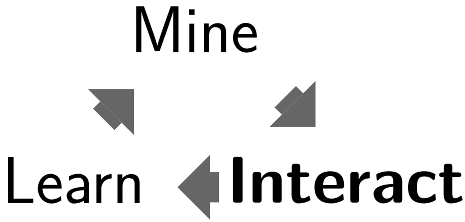


Mine

- Provide a sample of k patterns to the user (called the query $\mathcal{Q}$)

Interactive pattern mining: overview

Interactive data exploration using pattern mining. (van Leeuwen 2014)

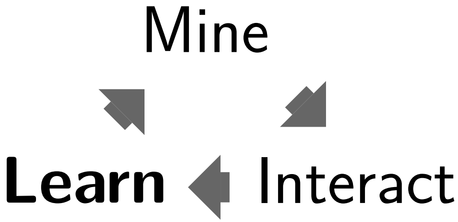


Interact

- Like/dislike or rank or rate the patterns

Interactive pattern mining: overview

Interactive data exploration using pattern mining. (van Leeuwen 2014)



Learn

- Generalize user feedback for building a preference model

Interactive pattern mining: overview

Interactive data exploration using pattern mining. (van Leeuwen 2014)

Mine

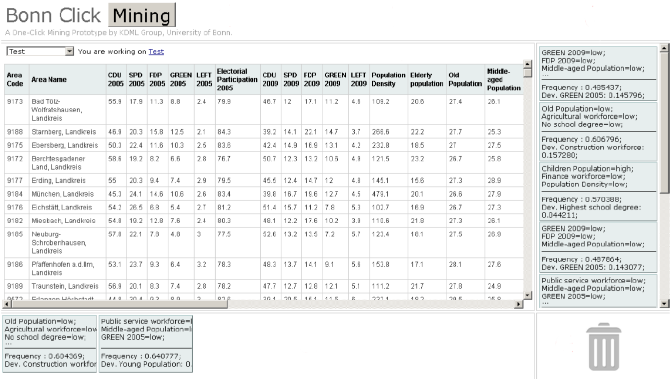
Learn Interact

Mine (again!)

- Provide a sample of k patterns **benefiting from the preference model**

Interactive pattern mining

Multiple mining algorithms

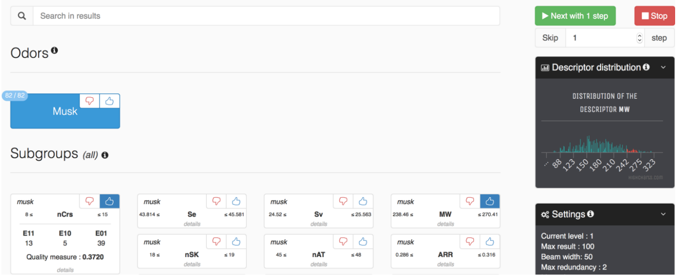


Area Code	Area Name	CDU 2005	SPD 2005	FDP 2005	GREEN 2005	LEFT 2005	Electorial Participation 2005	CDU 2009	SPD 2009	FDP 2009	GREEN 2009	LEFT 2009	Population Density	Elderly population	Old Population	Middle-aged Population
9173	Bad Tölz-Wolfratshausen, Landkreis	55.9	17.9	11.3	8.8	2.4	79.9	46.7	12	17.1	11.2	4.6	109.2	20.6	27.4	26.1
9188	Starnberg, Landkreis	46.9	20.3	15.8	12.5	2.1	84.3	39.2	14.1	22.1	14.7	3.7	266.6	22.2	27.7	25.3
9175	Ebersberg, Landkreis	50.3	22.4	11.6	10.3	2.5	83.6	42.4	14.9	16.9	13.1	4.2	232.8	18.5	27	27.5
9172	Berchtesgadener Land, Landkreis	58.6	19.2	8.2	6.6	2.8	76.7	50.7	12.3	13.2	10.6	4.9	121.5	23.2	26.7	25.8
9177	Erding, Landkreis	55	20.3	9.4	7.4	2.9	79.5	45.5	12.4	14.7	12	4.8	145.1	15.6	27.3	28.9
9184	München, Landkreis	45.3	24.1	14.6	10.6	2.6	83.4	39.8	16.7	19.6	12.7	4.5	479.1	20.1	26.6	27.9
9176	Eichstätt, Landkreis	54.2	26.5	6.8	5.4	2.7	81.2	51.4	15.7	11.2	7.8	5.3	102.7	16.9	26.7	27.3
9182	Miesbach, Landkreis	54.8	19.2	12.8	7.6	2.4	80.3	48.1	12.2	17.6	10.2	3.9	110.6	21.8	27.3	26.1
9185	Neuburg-Schrobenhausen, Landkreis	57.8	22.1	7.8	4.8	3	77.5	52.6	13.2	13.5	7.2	5.7	123.4	18.1	27.5	26.9
9186	Pfaffenhofen a.d.Ilm, Landkreis	53.1	23.7	9.3	6.4	3.2	78.3	48.3	13.7	14.1	9.1	5.6	153.8	17.1	28.1	27.6
9189	Traunstein, Landkreis	56.9	20.1	8.3	7.4	2.8	78.2	47.7	12.7	12.8	12.1	5.1	111.2	21.7	27.8	24.9

One Click Mining - Interactive Local Pattern Discovery through Implicit Preference and Performance Learning. (Boley et al. IDEA13)

Interactive pattern mining

Platform that implements descriptive rule discovery algorithms suited for neuroscientists



h(odor): Interactive Discovery of Hypotheses on the Structure-Odor Relationship in Neuroscience. (Bosc et al. ECML/PKDD16 (demo))

Interactive pattern mining: challenges

- MINE
  - Instant discovery for facilitating the iterative process
  - Preference model integration for improving the pattern quality
  - Pattern diversity for completing the preference model
- INTERACT
  - Simplicity of user feedback (binary feedback $>$ graded feedback)
  - Accuracy of user feedback (binary feedback $<$ graded feedback)
- LEARN
  - Expressivity of the preference model
  - Ease of learning of the preference model

Interactive pattern mining: challenges

- MINE
  - *Instant discovery for facilitating the iterative process*
  - *Preference model integration for improving the pattern quality*
  - Pattern diversity for completing the preference model
- INTERACT
  - Simplicity of user feedback (binary feedback $>$ graded feedback)
  - Accuracy of user feedback (binary feedback $<$ graded feedback)
- LEARN
  - *Expressivity of the preference model*
  - Ease of learning of the preference model

➠ Optimal mining problem (according to preference model)

Interactive pattern mining: challenges

- MINE
  - Instant discovery for facilitating the iterative process
  - Preference model integration for improving the pattern quality
  - *Pattern diversity for completing the preference model*
- INTERACT
  - *Simplicity of user feedback (binary feedback $>$ graded feedback)*
  - *Accuracy of user feedback (binary feedback $<$ graded feedback)*
- LEARN
  - Expressivity of the preference model
  - *Ease of learning of the preference model*

➠ Active learning problem

Learn: Preference model

How user preferences are represented?

Problem

- Expressivity of the preference model
- Ease of learning of the preference model

Learn: Preference model

How user preferences are represented?

Problem

- Expressivity of the preference model
- Ease of learning of the preference model

Weighted product model

- A weight on items $\mathcal{I}$
- Score for a pattern X = product of weights of items in X
- (Bhuiyan et al. CIKM12, Dzyuba et al. PAKDD17)

$$
\begin{array}{lccccccc}
 & \omega_A & & \omega_B & & \omega_C & & \\
AB & 4 & \times & 1 & & & = & 4 \\
BC & & & 1 & \times & 0.5 & = & 0.5
\end{array}
$$

LEARN: Preference model

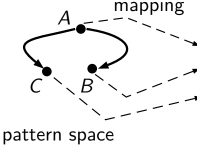
How user preferences are represented?

Problem

- Expressivity of the preference model
- Ease of learning of the preference model

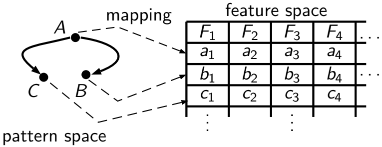
Feature space model

- Partial order over the pattern language $\mathcal{L}$
- Mapping between a pattern X and a set of features:



F_1	F_2	F_3	F_4	…
a_1	a_2	a_3	a_4	
b_1	b_2	b_3	b_4	…
c_1	c_2	c_3	c_4	
⋮		⋮		

LEARN: Feature space model



Feature space

- = assumption about the user preferences
- the more, the better

Different feature spaces:

- Attributes of the mined dataset (Rueping ICML09)
- Expected and measured frequency (Xin et al. KDD06)
- Attributes, coverage, chi-squared, length and so on (Dzyuba et al. ICTAI13)

INTERACT: User feedback

How user feedback are represented?

Problem

- Simplicity of user feedback (binary feedback $>$ graded feedback)
- Accuracy of user feedback (binary feedback $<$ graded feedback)

INTERACT: User feedback

How user feedback are represented?

Problem

- Simplicity of user feedback (binary feedback $>$ graded feedback)
- Accuracy of user feedback (binary feedback $<$ graded feedback)

Weighted product model

- Binary feedback (like/dislike) (Bhuiyan et al. CIKM12, Dzyuba et al. PAKDD17)

pattern	feedback
A	like
AB	like
BC	dislike

INTERACT: User feedback

How user feedback are represented?

Problem

- Simplicity of user feedback (binary feedback $>$ graded feedback)
- Accuracy of user feedback (binary feedback $<$ graded feedback)

Feature space model

- Ordered feedback (ranking) (Xin et al. KDD06, Dzyuba et al. ICTAI13)

$$A \succ AB \succ BC$$

- Graded feedback (rate) (Rueping ICML09)

pattern	feedback
A	0.9
AB	0.6
BC	0.2

LEARN: Preference learning method

How user feedback are generalized to a model?

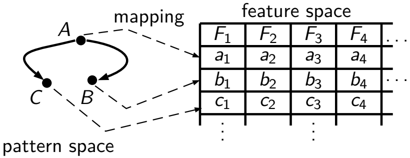
- **Weighted product model**
  - Counting likes and dislikes for each item: $\omega = \beta^{(\#\text{like - }\#\text{dislike})}$ (Bhuiyan et al. ICML12, Dzyuba et al. PAKDD17)

pattern	feedback	A	B	C
A	like	1		
AB	like	1	1	
BC	dislike		-1	-1
		$2^{2-0} = 4$	$2^{1-1} = 1$	$2^{0-1} = 0.5$

- **Feature space model**
  - $=$ learning to rank (Rueping ICML09, Xin et al. KDD06, Dzyuba et al. ICTAI13)

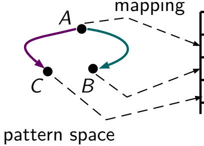
Learn: Learning to rank

How to learn a model from a ranking?



LEARN: Learning to rank

How to learn a model from a ranking?



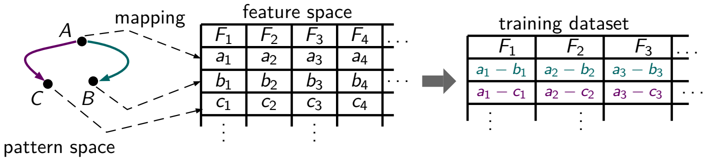
F_1	F_2	F_3	F_4	...
a_1	a_2	a_3	a_4	
b_1	b_2	b_3	b_4	...
c_1	c_2	c_3	c_4	
⋮		⋮		

F_1	F_2	F_3	...
$a_1 - b_1$	$a_2 - b_2$	$a_3 - b_3$	
$a_1 - c_1$	$a_2 - c_2$	$a_3 - c_3$	...
⋮	⋮		

1. Calculate the distances between feature vectors for each pair (training dataset)

LEARN: Learning to rank

How to learn a model from a ranking?



1. Calculate the distances between feature vectors for each pair (training dataset)
2. Minimize the loss function stemming from this training dataset

Algorithms: SVM Rank (Joachims KDD02), AdaRank (Xu et al. SIGIR07),...

LEARN: Active learning problem

How are selected the set of patterns (query $\mathcal{Q}$)?

Problem

- Mining the most relevant patterns according to *Quality*
- Querying patterns that provide more information about preferences
  (NP-hard problem for pair-wise preferences (Ailon JMLR12))

- Heuristic criteria:
  - **Local diversity:** diverse patterns among the current query $\mathcal{Q}$
  - **Global diversity:** diverse patterns among the different queries $\mathcal{Q}_i$
  - **Density:** dense regions are more important

LEARN: Active learning heuristics
(Dzyuba et al. ICTAI13)

What is the interest of the pattern X for the current pattern query $\mathcal{Q}$?

- **Maximal Marginal Relevance:** querying diverse patterns in $\mathcal{Q}$

$$\alpha Quality(X) + (1-\alpha) \min_{Y \in \mathcal{Q}} dist(X, Y)$$

- **Global MMR:** taking into account previous queries

$$\alpha Quality(X) + (1-\alpha) \min_{Y \in \bigcup_i \mathcal{Q}_i} dist(X, Y)$$

- **Relevance, Diversity, and Density:** querying patterns from dense regions provides more information about preferences

$$\alpha Quality(X) + \beta Density(X) + (1-\alpha-\beta) \min_{Y \in \mathcal{Q}} dist(X, Y)$$

MINE: Mining strategies

What method is used to mine the pattern query $\mathcal{Q}$?

Problem

- Instant discovery for facilitating the iterative process
- Preference model integration for improving the pattern quality
- Pattern diversity for completing the preference model

MINE: Mining strategies

What method is used to mine the pattern query $\mathcal{Q}$?

Problem

- Instant discovery for facilitating the iterative process
- Preference model integration for improving the pattern quality
- Pattern diversity for completing the preference model

Post-processing

- Re-rank the patterns with the updated quality (Rueping ICML09, Xin et al. KDD06)
- Clustering as heuristic for improving the local diversity (Xin et al. KDD06)

MINE: Mining strategies

What method is used to mine the pattern query $\mathcal{Q}$?

Problem

- Instant discovery for facilitating the iterative process
- Preference model integration for improving the pattern quality
- Pattern diversity for completing the preference model

Optimal pattern mining (Dzyuba et al. ICTAI13)

- Beam search based on reweighing subgroup quality measures for finding the best patterns
- Previous active learning heuristics (and more)

Mine: Mining strategies

What method is used to mine the pattern query $\mathcal{Q}$?

Problem

- Instant discovery for facilitating the iterative process
- Preference model integration for improving the pattern quality
- Pattern diversity for completing the preference model

Pattern sampling (Bhuiyan et al. CIKM12, Dzyuba et al. PAKDD17)

- Randomly draw pattern with a distribution proportional to their updated quality
- Sampling as heuristic for diversity and density

Objective evaluation protocol

Methodology = simulate a user

1. Select a subset of data or pattern as **user interest**
2. Use a metric for simulating user feedback

User interest:

- A set of items (Bhuiyan et al. CIKM12, Dzyuba et al. PAKDD17)
- A sample for modeling the user's prior knowledge (Xin et al. KDD06)
- A class (Rueping ICML09, Dzyuba et al. ICTAI13)

Results

Objective evaluation results

- Dozens of iterations for few dozens of examined patterns
- Important pattern features depends on the user interest
- Randomized selectors ensure high diversity

Results

Objective evaluation results

- Dozens of iterations for few dozens of examined patterns
- Important pattern features depends on the user interest
- Randomized selectors ensure high diversity

Questions?

- How to select the right set of (hidden) features for modeling user preferences?
- How to subjectively evaluate interactive pattern mining?
  ➠ qualitative benchmarks for pattern mining

Creedo – Scalable and Repeatable Extrinsic Evaluation for Pattern Discovery Systems by Online User Studies. (Boley et al. IDEA15)

Instant pattern discovery

The need

"the user should be allowed to pose and refine queries at any moment in time and the system should respond to these queries instantly"

Providing Concise Database Covers Instantly by Recursive Tile Sampling. (Moens et al. DS14)

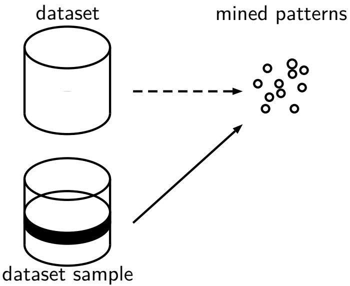
➠ few seconds between the query and the answer

Methods

- ~~Sound and complete pattern mining~~
- Beam search Subgroup Discovery methods
- Monte Carlo tree search (Bosc et al. 2016)
- **Pattern sampling**

Dataset sampling vs Pattern sampling

Dataset sampling

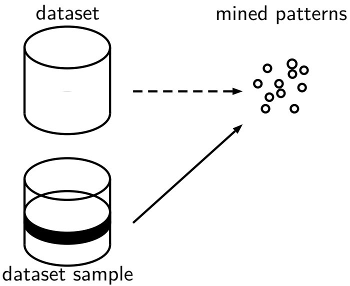


Finding all patterns from a transaction sample
➠ input space sampling

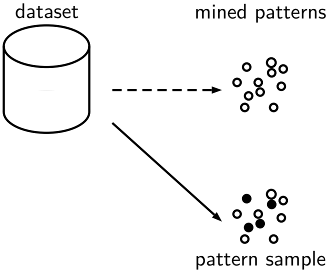
Sampling large databases for association rules. (Toivonen et al. VLDB96)

Dataset sampling vs Pattern sampling

Dataset sampling



Pattern sampling



Finding all patterns from a transaction sample
➠ input space sampling

Finding a pattern sample from all transactions
➠ output space sampling

Random sampling from databases. (Olken, PhD93)

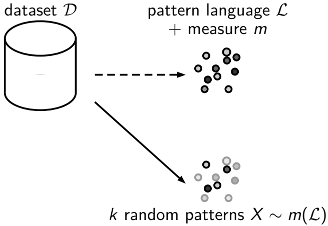
Pattern sampling: References

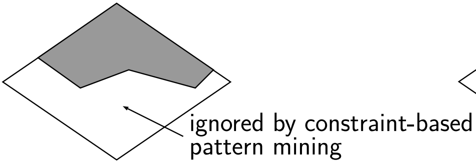
- Output Space Sampling for Graph Patterns. (Al Hasan et al. VLDB09)
- Direct local pattern sampling by efficient two-step random procedures. (Boley et al. KDD11)
- Interactive Pattern Mining on Hidden Data: A Sampling-based Solution. (Bhuiyan et al. CIKM12)
- Linear space direct pattern sampling using coupling from the past. (Boley et al. KDD12)
- Randomly sampling maximal itemsets. (Moens et Goethals IDEA13)
- Instant Exceptional Model Mining Using Weighted Controlled Pattern Sampling. (Moens et al. IDA14)
- Unsupervised Exceptional Attributed Sub-graph Mining in Urban Data (Bendimerad et al. ICDM16)
- Giacometti and Soulet: Dense Neighborhood Pattern Sampling in Numerical Data. (Giacometti and Soulet SIAMDM18)

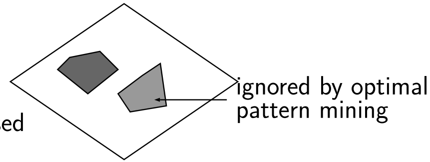
Pattern sampling: Problem

Problem

- **Inputs:** a pattern language $\mathcal{L}$ + a measure $m : \mathcal{L} \rightarrow \Re$
- **Output:** a family of k realizations of the random set $R \sim m(\mathcal{L})$





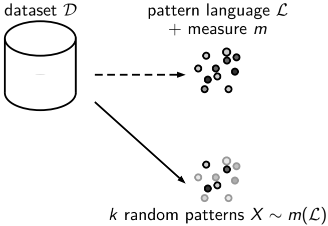


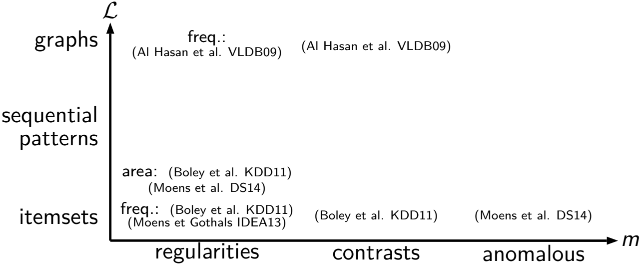
Pattern sampling addresses the full pattern language $\mathcal{L}$ ➠ **diversity!**

Pattern sampling: Problem

Problem

- **Inputs:** a pattern language $\mathcal{L}$ + a measure $m : \mathcal{L} \rightarrow \Re$
- **Output:** a family of k realizations of the random set $R \sim m(\mathcal{L})$



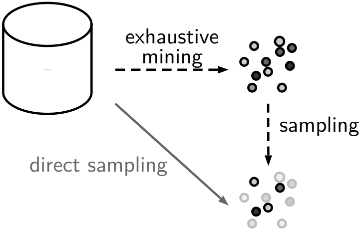


Pattern sampling: Challenges

Naive method

1. Mine all the patterns with their interestingness m
2. Sample this set of patterns according to m

➠ Time consuming / infeasible

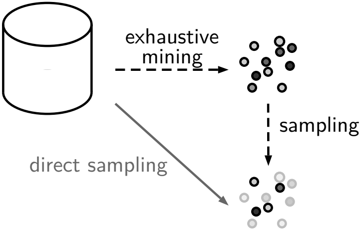


Pattern sampling: Challenges

Naive method

1. Mine all the patterns with their interestingness *m*
2. Sample this set of patterns according to *m*

➠ Time consuming / infeasible

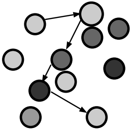


Challenges

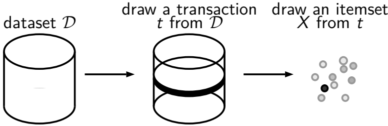
- Trade-off between pre-processing computation and processing time per pattern
- Quality of sampling

Two main families

1. Stochastic techniques
   - Metropolis-Hastings algorithm
   - Coupling From The Past



2. Direct techniques
   - Item/transaction sampling with rejection
   - **Two-step random procedure**



Two-step procedure: Toy example

Direct local pattern sampling by efficient two-step random procedures. (Boley et al. KDD11)

TId	Items		
t_1	A	B	C
t_2	A	B	
t_3		B	C
t_4			C

Mine all frequent patterns

Itemset	freq.
A	2
B	3
C	3
AB	2
AC	1
BC	2
ABC	1

Pick 14 itemsets

Itemsets
A, A
B, B, B
C, C, C
AB, AB
AC
BC, BC
ABC

Two-step procedure: Toy example

Direct local pattern sampling by efficient two-step random procedures. (Boley et al. KDD11)

TId	Items		
t_1	A	B	C
t_2	A	B	
t_3		B	C
t_4			C

infeasible

~~Mine all frequent patterns~~

Itemset	freq.
A	2
B	3
C	3
AB	2
AC	1
BC	2
ABC	1

Pick 14 itemsets

Direct sampling

Itemsets
A, A
B, B, B
C, C, C
AB, AB
AC
BC, BC
ABC

Two-step procedure: Toy example

Direct local pattern sampling by efficient two-step random procedures. (Boley et al. KDD11)

infeasible

~~Mine all frequent patterns~~

TId	Items		
t_1	A	B	C
t_2	A	B	
t_3		B	C
t_4			C

Itemset	freq.
A	2
B	3
C	3
AB	2
AC	1
BC	2
ABC	1

Pick 14 itemsets

Itemsets
A, A
B, B, B
C, C, C
AB, AB
AC
BC, BC
ABC

Rearrange itemsets

TId	Itemsets
t_1	A, B, C, AB, AC, BC, ABC
t_2	A, B, AB
t_3	B, C, BC
t_4	C

Two-step procedure: Toy example

Direct local pattern sampling by efficient two-step random procedures. (Boley et al. KDD11)

infeasible

~~Mine all frequent patterns~~

TId	Items			weight ω
t_1	A	B	C	$2^3 - 1 = 7$
t_2	A	B		$2^2 - 1 = 3$
t_3		B	C	$2^2 - 1 = 3$
t_4			C	$2^1 - 1 = 1$

Itemset	freq.
A	2
B	3
C	3
AB	2
AC	1
BC	2
ABC	1

Pick 14 itemsets

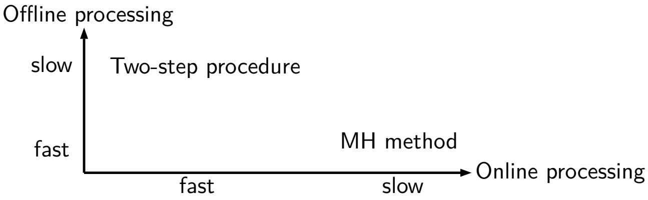
Itemsets
A, A
B, B, B
C, C, C
AB, AB
AC
BC, BC
ABC

1. Pick a transaction proportionally to ω

TId	Itemsets
t_1	A, B, C, AB, AC, BC, ABC
t_2	A, B, AB
t_3	B, C, BC
t_4	C

2. Pick an itemset uniformly

Two-step procedure: Comparison

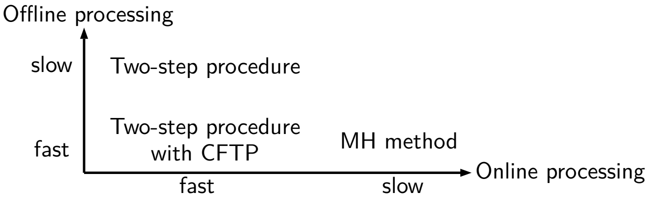


Complexity depends on the measure m:

Measure $m(X)$	Preprocessing	k realizations
$supp(X,\mathcal{D})$	$O(\lvert\mathcal{I}\rvert \times \lvert\mathcal{D}\rvert)$	$O(k(\lvert\mathcal{I}\rvert + \ln\lvert\mathcal{D}\rvert))$
$supp(X,\mathcal{D}) \times \lvert X\rvert$	$O(\lvert\mathcal{I}\rvert \times \lvert\mathcal{D}\rvert)$	$O(k(\lvert\mathcal{I}\rvert + \ln\lvert\mathcal{D}\rvert))$
$supp_+(X,\mathcal{D}) \times (\lvert\mathcal{D}_-\rvert - supp_-(X,\mathcal{D}))$	$O(\lvert\mathcal{I}\rvert^2 \times \lvert\mathcal{D}\rvert^2)$	$O(k(\lvert\mathcal{I}\rvert + \ln^2\lvert\mathcal{D}\rvert))$
$supp(X,\mathcal{D})^2$	$O(\lvert\mathcal{I}\rvert^2 \times \lvert\mathcal{D}\rvert^2)$	$O(k(\lvert\mathcal{I}\rvert + \ln^2\lvert\mathcal{D}\rvert))$

Preprocessing time may be prohibitive

Two-step procedure: Comparison



Complexity depends on the measure m:

Measure $m(X)$	Preprocessing	k realizations
$supp(X, \mathcal{D})$	$O(\lvert\mathcal{I}\rvert \times \lvert\mathcal{D}\rvert)$	$O(k(\lvert\mathcal{I}\rvert + \ln \lvert\mathcal{D}\rvert))$
$supp(X, \mathcal{D}) \times \lvert X\rvert$	$O(\lvert\mathcal{I}\rvert \times \lvert\mathcal{D}\rvert)$	$O(k(\lvert\mathcal{I}\rvert + \ln \lvert\mathcal{D}\rvert))$
$supp_+(X, \mathcal{D}) \times (\lvert\mathcal{D}_-\rvert - supp_-(X, \mathcal{D}))$	$O(\lvert\mathcal{I}\rvert^2 \times \lvert\mathcal{D}\rvert^2)$	$O(k(\lvert\mathcal{I}\rvert + \ln^2 \lvert\mathcal{D}\rvert))$
$supp(X, \mathcal{D})^2$	$O(\lvert\mathcal{I}\rvert^2 \times \lvert\mathcal{D}\rvert^2)$	$O(k(\lvert\mathcal{I}\rvert + \ln^2 \lvert\mathcal{D}\rvert))$

Preprocessing time may be prohibitive ➠ hybrid strategy with stochastic process for the first step:

Linear space direct pattern sampling using coupling from the past. (Boley et al. KDD12)

Pattern sampling

Summary

Pros

- Compact collection of patterns
- Threshold free
- Diversity
- Very fast

Cons

- Patterns far from optimality
- Not suitable for all interestingness measures

Pattern sampling

Summary

Pros

- Compact collection of patterns
- Threshold free
- Diversity
- Very fast

Cons

- Patterns far from optimality
- Not suitable for all interestingness measures

Interactive pattern sampling

Interactive Pattern Mining on Hidden Data: A Sampling-based Solution. (Bhuiyan et al. CIKM12)

➠ how to integrate more sophisticated user preference models?

Pattern set and sampling

Pattern-based models with iterative pattern sampling

- ORIGAMI: Mining Representative Orthogonal Graph Patterns. (Al Hasan et al. ICDM07)
- Randomly sampling maximal itemsets. (Moens et Goethals IDEA13)
- Providing Concise Database Covers Instantly by Recursive Tile Sampling. (Moens et al. DS14)

➠ how to sample a set of patterns instead of indivual patterns?

- Flexible constrained sampling with guarantees for pattern mining. (Dzyuba et al. 2016)

Interactive pattern mining:

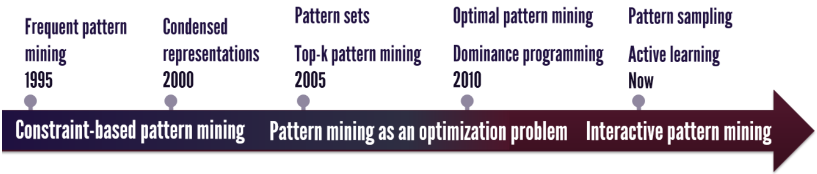
concluding remarks

- Preferences are not explicitly given by the user...
  ...but, representation of user preferences should be anticipated in upstream.

- Instant discovery enables a tight coupling between user and system...
  ...but, most advanced models are not suitable.

Concluding remarks

Preference-based pattern mining

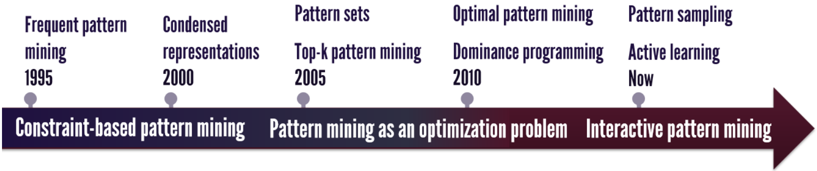


User preferences are more and more prominent...

from simple preference models to complex ones

- from frequency to anti-monotone constraints and more complex ones
- from 1 criterion (top-k) to multi-criteria (skyline)
- from weighted product model to feature space model

Preference-based pattern mining

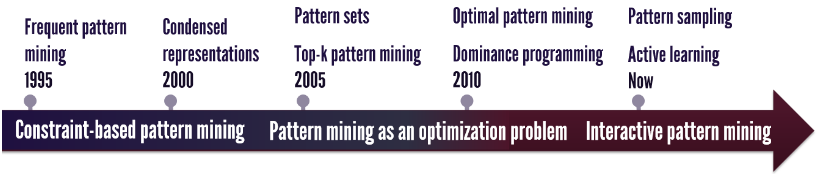


User preferences are more and more prominent. . .

from preference elicitation to preference acquisition

- user-defined constraint
- no threshold with optimal pattern mining
- no user-specified interestingness

Preference-based pattern mining



User preferences are more and more prominent in the community. . .

from data-centric methods:

- 2003-2004: Frequent Itemset Mining Implementations
- 2002-2007: Knowledge Discovery in Inductive Databases

to user-centric methods:

- 2010-2014: Useful Patterns
- 2015-2017: Interactive Data Exploration and Analytics

Multi-pattern domain exploration

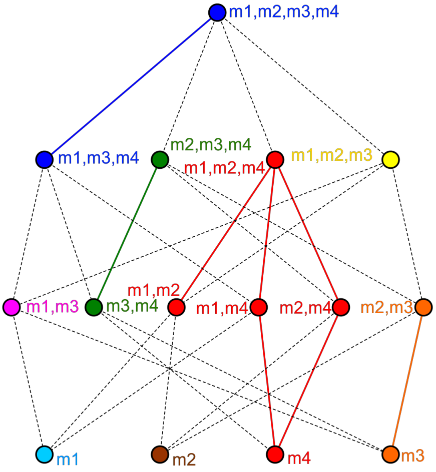
- The user has to choose its pattern domain of interest.
- What about (interactive) multi-pattern domain exploration?
  - Some knowledge nuggets can be depicted with simple pattern domain (e.g., itemset) while others require more sophisticated pattern domain (e.g., sequence, graph, dynamic graphs, etc.).
  - Examples in Olfaction:
    - Odorant molecules.
    - unpleasant odors in presence of Sulfur atom in chemicals $\Rightarrow$ itemset is enough.
    - Some chemicals have the same 2-d graph representation and totally different odor qualities (e.g., isomers) $\Rightarrow$ need to consider 3-d graph pattern domain.
  - How to fix the good level of description?
- Toward pattern sets involving several pattern domains.

Role/acquisition of preferences

through the skypattern cube

- equivalence classes on measures

  ➠ highlight the role of measures



Role/acquisition of preferences

through the skypattern cube

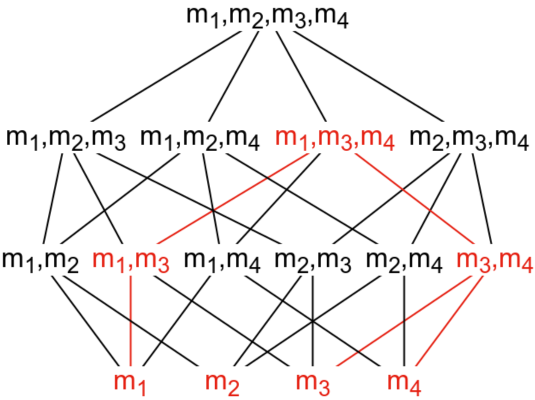
- equivalence classes on measures

  ➠ highlight the role of measures

- skypattern cube compression:

  user navigation and recommendation

- preference acquisition



Pattern mining in the AI field

- **cross-fertilization between data mining and constraint programming/SAT/ILP** (De Raedt et al. KDD08): designing generic and declarative approaches

  ➥ make easier the exploratory data mining process
  - avoiding writing solutions from scratch
  - easier to model new problems
- **open issues:**
  - how go further to integrate preferences?
  - how to define/learn constraints/preference?
  - how to visualize results and interact with the end user?
  - ...

  Many other directions associated to the AI field:
  integrating background knowledge, *knowledge representation*,...

Special thanks to:

Bruno Crémilleux (Université Caen, France)

Arnaud Soulet (Université de Tours, France)

Tijl de Bie (Ghent University, Belgium)

Albert Bifet (Télécom ParisTech, Paris)

Mario Boley (Max Planck Institute for Informatics, Saarbrücken, Germany)

Wouter Duivesteijn (TU Eindhoven, The Netherlands)

Matthijs van Leeuwen (Leiden University, The Netherlands)

Chedy Raïssi (INRIA-NGE, France)

Jilles Vreeken (Saarland University, Saarbrücken, Germany)

Albrecht Zimmermann (Université de Caen Normandie, France)

John O. R. Aoga, Tias Guns, and Pierre Schaus.
An efficient algorithm for mining frequent sequence with constraint programming.
In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II, pages 315–330, 2016.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, Jeremy Besson, and Mohammed J Zaki.
Origami: Mining representative orthogonal graph patterns.
In Seventh IEEE international conference on data mining (ICDM 2007), pages 153–162. IEEE, 2007.

Nir Ailon.
An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity.
Journal of Machine Learning Research, 13(Jan):137–164, 2012.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami.
Mining association rules between sets of items in large databases.
In Acm sigmod record, volume 22, pages 207–216. ACM, 1993.

Stefano Bistarelli and Francesco Bonchi.
Interestingness is not a dichotomy: Introducing softness in constrained pattern mining.
In Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings, pages 22–33, 2005.

Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti.
Free-sets: A condensed representation of boolean data for the approximation of frequency queries.
Data Min. Knowl. Discov., 7(1):5–22, 2003.

Francesco Bonchi, Josep Domingo-Ferrer, Ricardo A. Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, editors.
IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. IEEE, 2016.

Behrouz Babaki, Tias Guns, and Siegfried Nijssen.
Constrained clustering using column generation.
In International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems, pages 438–454. Springer, 2014.

Roberto J. Bayardo, Bart Goethals, and Mohammed Javeed Zaki, editors.
FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004, volume 126 of CEUR Workshop Proceedings. CEUR-WS.org, 2005.

Tijl De Bie.
Maximum entropy models and subjective interestingness: an application to tiles in binary databases.
Data Min. Knowl. Discov., 23(3):407–446, 2011.

Tijl De Bie.
Subjective interestingness in exploratory data mining.
In Advances in Intelligent Data Analysis XII - 12th International Symposium, IDA 2013, London, UK, October 17-19, 2013. Proceedings, pages 19–31, 2013.

Abdelhamid Boudane, Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi.
A sat-based approach for mining association rules.
In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 2472–2478, 2016.

Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli.
Fast generation of best interval patterns for nonmonotonic constraints.
In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II, pages 157–172, 2015.

Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli.
Revisiting pattern structure projections.
In 13th Int. Conf. ICFCA 2015, pages 200–215, 2015.

Mario Boley, Maike Krause-Traudes, Bo Kang, and Björn Jacobs.
Creedoscalable and repeatable extrinsic evaluation for pattern discovery systems by online user studies.
In ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, page 20. Citeseer, 2015.

Francesco Bonchi and Claudio Lucchese.
Extending the state-of-the-art of constraint-based pattern discovery.
Data Knowl. Eng., 60(2):377–399, 2007.

Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner.
Direct local pattern sampling by efficient two-step random procedures.

In Proceedings of the 17th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, pages 582–590. ACM, 2011.

Mario Boley, Sandy Moens, and Thomas Gärtner.
Linear space direct pattern sampling using coupling from the past.
In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 69–77. ACM, 2012.

Mansurul Bhuiyan, Snehasis Mukhopadhyay, and Mohammad Al Hasan.
Interactive pattern mining on hidden data: a sampling-based solution.
In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 95–104. ACM, 2012.

Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel.
One click mining: Interactive local pattern discovery through implicit preference and performance learning.
In Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, pages 27–35. ACM, 2013.

Guillaume Bosc, Marc Plantevit, Jean-François Boulicaut, Moustafa Bensafi, and Mehdi Kaytoue.
h (odor): Interactive discovery of hypotheses on the structure-odor relationship in neuroscience.
In ECML/PKDD 2016 (Demo), 2016.

Guillaume Bosc, Chedy Raïssy, Jean-François Boulicaut, and Mehdi Kaytoue.
Any-time diverse subgroup discovery with monte carlo tree search.
arXiv preprint arXiv:1609.08827, 2016.

Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal.
Mining frequent patterns with counting inference.
SIGKDD Explorations, 2(2):66–75, 2000.

Kailash Budhathoki and Jilles Vreeken.
The difference and the norm - characterising similarities and differences between databases.
In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II, pages 206–223, 2015.

Kailash Budhathoki and Jilles Vreeken.
Causal inference by compression.
In Bonchi et al. [BDB+16], pages 41–50.

Roel Bertens, Jilles Vreeken, and Arno Siebes.
Keeping it short and simple: Summarising complex event sequences with multivariate patterns.
In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pages 735–744, 2016.

Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut.
Closed patterns meet n-ary relations.
TKDD, 3(1), 2009.

Vineet Chaoji, Mohammad Al Hasan, Saeed Salem, Jérémy Besson, and Mohammed J. Zaki.

ORIGAMI: A novel and effective approach for mining representative orthogonal graph patterns.
Statistical Analysis and Data Mining, 1(2):67–84, 2008.

Moonjung Cho, Jian Pei, Haixun Wang, and Wei Wang.
Preference-based frequent pattern mining.
Int. Journal of Data Warehousing and Mining (IJDWM), 1(4):56–77, 2005.

Toon Calders, Christophe Rigotti, and Jean-François Boulicaut.
A survey on condensed representations for frequent sets.
In Constraint-Based Mining and Inductive Databases, European Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany, March 11-13, 2004, Revised Selected Papers, pages 64–80, 2004.

Ming-Wei Chang, Lev-Arie Ratinov, Nicholas Rizzolo, and Dan Roth.
Learning and inference with constraints.
In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008, pages 1513–1518, 2008.

Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tou Ng.
A lattice-based approach to query-by-example spoken document retrieval.
In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 363–370. ACM, 2008.

James Cussens.
Bayesian network learning by compiling to weighted MAX-SAT.
In UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008, pages 105–112, 2008.

Duen Horng Chau, Jilles Vreeken, Matthijs van Leeuwen, and Christos Faloutsos, editors.
Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, IDEA@KDD 2013, Chicago, Illinois, USA, August 11, 2013. ACM, 2013.

Tijl De Bie.
Subjective interestingness in exploratory data mining.
In Advances in Intelligent Data Analysis XII, pages 19–31. Springer, 2013.

Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, and Luc De Raedt.
Interactive learning of pattern rankings.
International Journal on Artificial Intelligence Tools, 23(06):1460026, 2014.

Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut.
Granularity of co-evolution patterns in dynamic attributed graphs.
In Advances in Intelligent Data Analysis XIII - 13th International Symposium, IDA 2014, Leuven, Belgium, October 30 - November 1, 2014. Proceedings, pages 84–95, 2014.

Vladimir Dzyuba and Matthijs van Leeuwen.
Learning what matters–sampling interesting patterns.
In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 534–546. Springer, 2017.

Vladimir Dzyuba, Matthijs van Leeuwen, and Luc De Raedt.
Flexible constrained sampling with guarantees for pattern mining.
arXiv preprint arXiv:1610.09263, 2016.

Vladimir Dzyuba, Matthijs Van Leeuwen, Siegfried Nijssen, and Luc De Raedt.

Active preference learning for ranking patterns.
In IEEE 25th Int. Conf. on Tools with Artificial Intelligence (ICTAI 2013), pages 532–539. IEEE, 2013.

Vladimir Dzyuba.
Mine, Interact, Learn, Repeat: Interactive Pattern-based Data Exploration.
PhD thesis, KU Leuven, 2017.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu.
A density-based algorithm for discovering clusters in large spatial databases with noise.
In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA, pages 226–231, 1996.

Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrac.
Foundations of Rule Learning.
Cognitive Technologies. Springer, 2012.

Johannes Fürnkranz and Eyke Hüllermeier.
Preference Learning.
Springer, 2011.

Ada Wai-Chee Fu, Renfrew W.-w. Kwong, and Jian Tang.
Mining N-most interesting itemsets.
In Foundations of Intelligent Systems, 12th International Symposium, ISMIS, pages 59–67, 2000.

Frédéric Flouvat, Jérémy Sanhes, Claude Pasquier, Nazha Selmaoui-Folcher, and Jean-François Boulicaut.

Improving pattern discovery relevancy by deriving constraints from expert models.
In ECAI, pages 327–332, 2014.

Arianna Gallo, Tijl De Bie, and Nello Cristianini.

In Knowledge Discovery in Databases (PKDD 2007), pages 438–445. Springer, 2007.

Floris Geerts, Bart Goethals, and Taneli Mielikäinen.

In Discovery Science, 7th International Conference, DS 2004, Padova, Italy, October 2-5, 2004, Proceedings, pages 278–289, 2004.

Martin Gebser, Thomas Guyet, René Quiniou, Javier Romero, and Torsten Schaub.

In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 1497–1504, 2016.

Liqiang Geng and Howard J Hamilton.

ACM Computing Surveys (CSUR), 38(3):9, 2006.

Bart Goethals, Sandy Moens, and Jilles Vreeken.

In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 757–760. ACM, 2011.

Tias Guns, Siegfried Nijssen, and Luc De Raedt.

IEEE Trans. Knowl. Data Eng., 25(2):402–418, 2013.

Arnaud Giacometti and Arnaud Soulet.

International Journal of Data Science and Analytics, pages 1–12, 2016.

Arnaud Giacometti and Arnaud Soulet.

In Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part II, pages 196–207, 2016.

Arnaud Giacometti and Arnaud Soulet.

In SIAM DM, pages 756–764, 2018.

Bernhard Ganter and Rudolf Wille.

Springer, 1999.

Bart Goethals and Mohammed Javeed Zaki, editors.

Jiawei Han and Yongjian Fu.

Mining multiple-level association rules in large databases.
IEEE Transactions on knowledge and data engineering, 11(5):798–805, 1999.

Wilhelmiina Hämäläinen and Matti Nykänen.

In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, pages 203–212, 2008.

Tony Hey, Stewart Tansley, Kristin M Tolle, et al.

Microsoft research Redmond, WA, 2009.

Mohammad Al Hasan and Mohammed J. Zaki.

PVLDB, 2(1):730–741, 2009.

Tomasz Imielinski and Heikki Mannila.

Commun. ACM, 39(11):58–64, 1996.

Thorsten Joachims.

In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 133–142. ACM, 2002.

Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi.

In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III, pages 403–418, 2013.

Saïd Jabbour, Lakhdar Sais, Yakoub Salhi, and Takeaki Uno.

In 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013, pages 289–298, 2013.

Mehdi Khiari, Patrice Boizumault, and Bruno Crémilleux.

In Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK, September 6-10, 2010. Proceedings, pages 552–567, 2010.

In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006, pages 237–244, 2006.

In Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings, pages 577–584, 2006.

In Integration of AI and OR Techniques in Constraint Programming - 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings, pages 198–215, 2016.

Jerry Kiernan and Evimaria Terzi.

In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, pages 417–425, 2008.

Sergei O. Kuznetsov.

Nauchno-Tekhnicheskaya Informatsiya, ser. 2(1):17–20, 1993.

In proceedings of Fourth International Conference on Knowledge Discovery & Data Mining (KDD'98), pages 80–86, New York, August 1998. AAAI Press.

The Journal of Machine Learning Research, 5:153–188, 2004.

In International Symposium on Intelligent Data Analysis, pages 203–214. Springer, 2014.

Providing concise database covers instantly by recursive tile sampling.
In International Conference on Discovery Science, pages 216–227. Springer, 2014.

Sandy Moens and Bart Goethals.

In Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, pages 79–86. ACM, 2013.

Marianne Mueller and Stefan Kramer.

In International Conference on Discovery Science, pages 159–173. Springer, 2010.

Shinichi Morishita and Jun Sese.

In Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA, pages 226–236, 2000.

Benjamin Negrevergne, Anton Dries, Tias Guns, and Siegfried Nijssen.

In IEEE 13th Int. Conf. on Data Mining (ICDM 2013), pages 557–566. IEEE, 2013.

Raymond T Ng, Laks VS Lakshmanan, Jiawei Han, and Alex Pang.

In ACM Sigmod Record, volume 27, pages 13–24. ACM, 1998.

Siegfried Nijssen and Albrecht Zimmermann.

In Frequent Pattern Mining, pages 147–163. Springer, 2014.

Frank Olken.

PhD thesis, University of California, Berkeley, 1993.

Abdelkader Ouali, Samir Loudni, Yahia Lebbah, Patrice Boizumault, Albrecht Zimmermann, and Lakhdar Loukil.

In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 647–654, 2016.

Jian Pei, Jiawei Han, and Laks V. S. Lakshmanan.

Data Min. Knowl. Discov., 8(3):227–252, 2004.

Kai Puolamäki, Bo Kang, Jefrey Lijffijt, and Tijl De Bie.

In Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II, pages 214–229, 2016.

Apostolos N. Papadopoulos, Apostolos Lyritsis, and Yannis Manolopoulos.

Data Min. Knowl. Discov., 17(1):57–76, 2008.

Luc De Raedt, Tias Guns, and Siegfried Nijssen.

In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, pages 204–212, 2008.

Stefan Rueping.

In Proceedings of the 26th Annual International Conference on Machine Learning, pages 913–920. ACM, 2009.

Luc De Raedt and Albrecht Zimmermann.

In Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA, pages 237–248, 2007.

Intell. Data Anal., 13(1):109–133, 2009.

In IEEE 11th Int. Conf on Data Mining (ICDM 2011), pages 655–664. IEEE, 2011.

In Proceedings of the Sixth SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA, pages 395–406, 2006.

In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 59–66. ACM, 2005.

Hannu Toivonen et al.

In VLDB, volume 96, pages 134–145, 1996.

Charalampos E. Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria A. Tsiarli.

In The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013, pages 104–112, 2013.

Artif. Intell., 244:48–69, 2017.

In Integration of AI and OR Techniques in Constraint Programming - 11th International Conference, CPAIOR 2014, Cork, Ireland, May 19-23, 2014. Proceedings, pages 71–87, 2014.

Modeling and mining optimal patterns using dynamic csp.
In IEEE 27th Int. Conf. on Tools with Artificial Intelligence (ICTAI 2015), pages 33–40. IEEE, 2015.

Takeaki Uno.

In ISAAC 2007, pages 402–414, 2007.

Matthijs van Leeuwen.

In Interactive Knowledge Discovery and Data Mining in Biomedical Informatics, pages 169–182. Springer, 2014.

Matthijs van Leeuwen, Tijl De Bie, Eirini Spyropoulou, and Ceédric Mesnage.

Machine Learning, In press.

Matthijs van Leeuwen and Arno J. Knobbe.

Data Min. Knowl. Discov., 25(2):208–242, 2012.

Matthijs van Leeuwen and Antti Ukkonen.

In Machine Learning and Knowledge Discovery in Databases, pages 272–287. Springer, 2013.

Jilles Vreeken and Arno Siebes.

In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, pages 1067–1072. IEEE Computer Society, 2008.

Dong Xin, Hong Cheng, Xifeng Yan, and Jiawei Han.

In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006, pages 444–453, 2006.

Dong Xin, Xuehua Shen, Qiaozhu Mei, and Jiawei Han.

In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 773–778. ACM, 2006.

Data Knowl. Eng., 59(3):603–626, 2006.

Machine Learning, 77(1):125–159, 2009.