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
Computer Science Dept.
University Claude Bernard Lyon 1.

Lab: LIRIS UMR 5205
Team: Data Mining & Machine Learning

Research Interest: Foundations of constraint-based pattern mining, sequences, augmented graphs.
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. ⇒ 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.

...
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!

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 (GRAISeach - 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

Functionalities

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

Fayad et al., 1996
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

- **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


P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Wiley, 2005

Charu C. Aggarwal, Data Mining, Springer, 2015.

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

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 Alzheimer's patients.

M. Boley www.realkd.org
ML versus DM

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

Global linear regression model

\[ y = 0.63x + 0.07 \]

Gaussian process model

\[ y = \sum_{j \in S} \alpha_j k(x, x(j)) \]
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"
We will focus on **descriptive data mining** especially on Constraint-based Pattern Mining with an **inductive database vision**.

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

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

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:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$\ldots$</th>
<th>$a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$d_{1,1}$</td>
<td>$d_{1,2}$</td>
<td>$\ldots$</td>
<td>$d_{1,n}$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$d_{2,1}$</td>
<td>$d_{2,2}$</td>
<td>$\ldots$</td>
<td>$d_{2,n}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$o_m$</td>
<td>$d_{m,1}$</td>
<td>$d_{m,2}$</td>
<td>$\ldots$</td>
<td>$d_{m,n}$</td>
</tr>
</tbody>
</table>

Question
How many itemsets are there?

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$\ldots$</th>
<th>$a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$d_{1,1}$</td>
<td>$d_{1,2}$</td>
<td>$\ldots$</td>
<td>$d_{1,n}$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$d_{2,1}$</td>
<td>$d_{2,2}$</td>
<td>$\ldots$</td>
<td>$d_{2,n}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$o_m$</td>
<td>$d_{m,1}$</td>
<td>$d_{m,2}$</td>
<td>$\ldots$</td>
<td>$d_{m,n}$</td>
</tr>
</tbody>
</table>

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

**Question**

How many itemsets are there? $2^{\left| \mathcal{A} \right|}$.
Transactional representation of the data

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

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>\ldots</th>
<th>( a_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( d_{1,1} )</td>
<td>( d_{1,2} )</td>
<td>\ldots</td>
<td>( d_{1,n} )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( d_{2,1} )</td>
<td>( d_{2,2} )</td>
<td>\ldots</td>
<td>( d_{2,n} )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\ddots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( o_m )</td>
<td>( d_{m,1} )</td>
<td>( d_{m,2} )</td>
<td>\ldots</td>
<td>( d_{m,n} )</td>
</tr>
</tbody>
</table>

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

Example

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_3 )</td>
<td></td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_4 )</td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
</tbody>
</table>

Transaction representation: \( D \) is an array of subsets of \( \mathcal{A} \)

\[ t_1 \]
\[ t_2 \]
\[ \vdots \]
\[ t_m \]

where \( t_i \subseteq \mathcal{A} \)
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 $|\{o \in \mathcal{O} | \{o\} \times X \subseteq \mathcal{D}\}|$.

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 $\frac{|\{o \in \mathcal{O} | \{o\} \times X \subseteq \mathcal{D}\}|}{|\mathcal{O}|}$. The relative frequency is a joint probability.
Frequent itemset mining

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

**Input:**

<table>
<thead>
<tr>
<th>$O_1$</th>
<th>$d_{1,1}$</th>
<th>$d_{1,2}$</th>
<th>$\ldots$</th>
<th>$d_{1,n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_2$</td>
<td>$d_{2,1}$</td>
<td>$d_{2,2}$</td>
<td>$\ldots$</td>
<td>$d_{2,n}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$O_m$</td>
<td>$d_{m,1}$</td>
<td>$d_{m,2}$</td>
<td>$\ldots$</td>
<td>$d_{m,n}$</td>
</tr>
</tbody>
</table>

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

---

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 $\mu$ 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%).

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$o_2$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>
Frequent itemset mining: illustration

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
</tr>
</tbody>
</table>

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:

\[ \{ d \in D \mid q(d, D) \} \]

where:

- $D$ is a dataset (tuples),
- $q$ is a query.
Inductive database vision

Querying patterns:

\[ \{ X \in P \mid Q(X, D) \} \]

where:

- \( D \) is the dataset,
- \( P \) is the pattern space,
- \( Q \) is an inductive query.
Inductive database vision

Querying the frequent itemsets:

\[ \{ X \in P \mid Q(X, D) \} \]

where:
- \( D \) is the dataset,
- \( P \) is the pattern space,
- \( Q \) is an inductive query.
Querying the frequent itemsets:

\[ \{X \in P \mid Q(X, D)\} \]

where:

- \(D\) is a subset of \(O \times A\), i.e., objects described with Boolean attributes,
- \(P\) is the pattern space,
- \(Q\) is an inductive query.
Inductive database vision

Querying the frequent itemsets:

\[ \{ X \in P \mid Q(X, D) \} \]

where:

- \( D \) is a subset of \( O \times A \), i.e., objects described with Boolean attributes,
- \( P \) is \( 2^A \),
- \( Q \) is an inductive query.
Inductive database vision

Querying the frequent itemsets:

\[ \{ X \in P \mid Q(X, D) \} \]

where:

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

\[ \{ X \in P \mid Q(X, D) \} \]

where:

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

Querying the frequent itemsets:

\[ \{ X \in P \mid Q(X, D) \} \]

where:

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

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

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

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_3 )</td>
<td></td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_4 )</td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
 f(\emptyset, \mathcal{D}) &= 4 \\
 f(\{a_1\}, \mathcal{D}) &= 2 \\
 f(\{a_1, a_2\}, \mathcal{D}) &= 2 \\
 f(\{a_1, a_2, a_3\}, \mathcal{D}) &= 1
\end{align*}
\]
Anti-monotonicity of the frequency

**Theorem**

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

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$o_2$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

- $f(\emptyset, \mathcal{D}) = 4$
- $f(\{a_3\}, \mathcal{D}) = 2$
- $f(\{a_1, a_3\}, \mathcal{D}) = 1$
- $f(\{a_1, a_2, a_3\}, \mathcal{D}) = 1$
Anti-monotonicity of the frequency

Corollary

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

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$f(\emptyset, \mathcal{D}) = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$f({a_3}, \mathcal{D}) = 2$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
<td>$f({a_1, a_3}, \mathcal{D}) = 1$</td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
<td>$f({a_1, a_2, a_3}, \mathcal{D}) = 1$</td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
</tbody>
</table>
Anti-monotonicity of the frequency

Corollary

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

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
</tr>
</tbody>
</table>

$f(\emptyset, \mathcal{D}) = 4$

$f(\{a_3\}, \mathcal{D}) = 2$

$f(\{a_1, a_3\}, \mathcal{D}) = 1$

$f(\{a_1, a_2, a_3\}, \mathcal{D}) = 1$
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

Level 0

{1}

Level 1

A(4)  B(6)  C(4)  D(4)  E(5)

Level 2

AB(4)  AC(2)  AD(3)  AE(4)  BC(4)  BD(4)  BE(5)  CD(2)  CE(3)  DE(3)

Level 3

ABD(3)  ABE(4)  ADE(3)  BCE(3)  BDE(3)

Level 4

ABDE(3)
Example

\[ \text{minsups} = 2 \]

\[ \begin{array}{|c|c|}
\hline
\text{TID} & \text{Items} \\
\hline
100 & 1 3 4 \\
200 & 2 3 5 \\
300 & 1 2 3 5 \\
400 & 2 5 \\
\hline
\end{array} \]

\[ \begin{array}{|c|c|}
\hline
\text{itemset} & \text{sup} \\
\hline
\{1\} & 2 \\
\{2\} & 3 \\
\{3\} & 3 \\
\{4\} & 1 \\
\{5\} & 3 \\
\hline
\end{array} \]

\[ \text{Scan D} \]

\[ \begin{array}{|c|c|}
\hline
\text{itemset} & \text{sup} \\
\hline
\{1\} & 2 \\
\{2\} & 3 \\
\{3\} & 3 \\
\{5\} & 3 \\
\hline
\end{array} \]

\[ \text{L}_1 \]

\[ \begin{array}{|c|c|}
\hline
\text{itemset} & \text{sup} \\
\hline
\{1 3\} & 2 \\
\{2 3\} & 2 \\
\{2 5\} & 3 \\
\{3 5\} & 2 \\
\hline
\end{array} \]

\[ \rightarrow \]

\[ \text{Scan D} \]

\[ \begin{array}{|c|c|}
\hline
\text{itemset} & \text{sup} \\
\hline
\{1\} & 1 \\
\{1 3\} & 2 \\
\{1 5\} & 1 \\
\{2 3\} & 2 \\
\{2 5\} & 3 \\
\{3 5\} & 2 \\
\hline
\end{array} \]

\[ \text{L}_2 \]

\[ \rightarrow \]

\[ \text{Scan D} \]

\[ \begin{array}{|c|c|}
\hline
\text{itemset} & \text{sup} \\
\hline
\{2 3 5\} & 2 \\
\hline
\end{array} \]

\[ \text{L}_3 \]
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: \( A, D \) as an array of subsets of \( A, \mu \in \mathbb{N} \)
Output: \( \{ X \subseteq A \mid f(X, D) \geq \mu \} \)
\[
P \leftarrow \{ \{ a \} \mid a \in A \}
\]
while \( P \neq \emptyset \) do
  \[
P \leftarrow \text{output\_frequent}(P, D, \mu)
  \]
  \[
P \leftarrow \text{children}(P)
  \]
end while

Whatever the order on \( A \), 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
  $\mathcal{P} \leftarrow \text{output\_frequent}(\mathcal{P}, \mathcal{D}, \mu)$
  $\mathcal{P} \leftarrow \text{children}(\mathcal{P})$
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 $(P_i)_{i=1..k} \in (2^{\mathcal{A}})^k$:

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

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$o_2$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

$\{o \in \mathcal{O} \mid \{o\} \times \{a_1\} \subseteq \mathcal{D}\} = \{o_1, o_2\}$

$\{o \in \mathcal{O} \mid \{o\} \times \{a_2\} \subseteq \mathcal{D}\} = \{o_1, o_2, o_3\}$

$\{o \in \mathcal{O} \mid \{o\} \times \{a_3\} \subseteq \mathcal{D}\} = \{o_1, o_4\}$

$\{o \in \mathcal{O} \mid \{o\} \times \{a_1, a_2, a_3\} \subseteq \mathcal{D}\} = \{o_1\}$
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 $(P_i)_{i=1..k} \in (2^\mathcal{A})^k$:

$$\{o \in \mathcal{O} | \{o\} \times \bigcup_{i=1}^k P_i \subseteq \mathcal{D}\} = \bigcap_{i=1}^k \{o \in \mathcal{O} | \{o\} \times P_i \subseteq \mathcal{D}\}.$$

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
</tr>
</tbody>
</table>

$$\{o \in \mathcal{O} | \{o\} \times \{a_1, a_2\} \subseteq \mathcal{D}\} = \{o_1, o_2\}$$

$$\{o \in \mathcal{O} | \{o\} \times \{a_3\} \subseteq \mathcal{D}\} = \{o_1, o_4\}$$

$$\{o \in \mathcal{O} | \{o\} \times \{a_1, a_2, a_3\} \subseteq \mathcal{D}\} = \{o_1\}$$
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 $(P_i)_{i=1..k} \in (2^\mathcal{A})^k$:

$$\{ o \in \mathcal{O} | \{ o \} \times \bigcup_{i=1}^k P_i \subseteq \mathcal{D} \} = \bigcap_{i=1}^k \{ o \in \mathcal{O} | \{ o \} \times P_i \subseteq \mathcal{D} \} .$$

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td>$\times$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td>$\times$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$\{ o \in \mathcal{O} | \{ o \} \times \{ a_1, a_2 \} \subseteq \mathcal{D} \} = \{ o_1, o_2 \}$$

$$\{ o \in \mathcal{O} | \{ o \} \times \{ a_1, a_3 \} \subseteq \mathcal{D} \} = \{ o_1 \}$$

$$\{ o \in \mathcal{O} | \{ o \} \times \{ a_1, a_2, a_3 \} \subseteq \mathcal{D} \} = \{ o_1 \}$$
Vertical representation of the data

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

<table>
<thead>
<tr>
<th></th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(\ldots)</th>
<th>(a_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(o_1)</td>
<td>(d_{1,1})</td>
<td>(d_{1,2})</td>
<td>(\ldots)</td>
<td>(d_{1,n})</td>
</tr>
<tr>
<td>(o_2)</td>
<td>(d_{2,1})</td>
<td>(d_{2,2})</td>
<td>(\ldots)</td>
<td>(d_{2,n})</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\ddots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(o_m)</td>
<td>(d_{m,1})</td>
<td>(d_{m,2})</td>
<td>(\ldots)</td>
<td>(d_{m,n})</td>
</tr>
</tbody>
</table>

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

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

\[ i_1 \quad i_2 \quad \ldots \quad i_n \]

where \(i_j \subseteq \mathcal{O}\)
Vertical representation of the data

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

<table>
<thead>
<tr>
<th></th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(\ldots)</th>
<th>(a_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(o_1)</td>
<td>(d_{1,1})</td>
<td>(d_{1,2})</td>
<td>(\ldots)</td>
<td>(d_{1,n})</td>
</tr>
<tr>
<td>(o_2)</td>
<td>(d_{2,1})</td>
<td>(d_{2,2})</td>
<td>(\ldots)</td>
<td>(d_{2,n})</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(o_m)</td>
<td>(d_{m,1})</td>
<td>(d_{m,2})</td>
<td>(\ldots)</td>
<td>(d_{m,n})</td>
</tr>
</tbody>
</table>

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

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

\[ i_1 \ i_2 \ \ldots \ i_n \]

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: \( \mathcal{D} \subseteq \mathcal{O} \times \mathcal{A} \)  

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>\ldots</th>
<th>( a_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( d_{1,1} )</td>
<td>( d_{1,2} )</td>
<td>\ldots</td>
<td>( d_{1,n} )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( d_{2,1} )</td>
<td>( d_{2,2} )</td>
<td>\ldots</td>
<td>( d_{2,n} )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\ldots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( o_m )</td>
<td>( d_{m,1} )</td>
<td>( d_{m,2} )</td>
<td>\ldots</td>
<td>( d_{m,n} )</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th></th>
<th>( i_1 )</th>
<th>( i_2 )</th>
<th>\ldots</th>
<th>( i_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \mathcal{O}<em>{d</em>{1,1}} )</td>
<td>( \mathcal{O}<em>{d</em>{1,2}} )</td>
<td>\ldots</td>
<td>( \mathcal{O}<em>{d</em>{1,n}} )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \mathcal{O}<em>{d</em>{2,1}} )</td>
<td>( \mathcal{O}<em>{d</em>{2,2}} )</td>
<td>\ldots</td>
<td>( \mathcal{O}<em>{d</em>{2,n}} )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\ldots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( o_m )</td>
<td>( \mathcal{O}<em>{d</em>{m,1}} )</td>
<td>( \mathcal{O}<em>{d</em>{m,2}} )</td>
<td>\ldots</td>
<td>( \mathcal{O}<em>{d</em>{m,n}} )</td>
</tr>
</tbody>
</table>

where \( i_j \subseteq \mathcal{O} \)

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

Like APriori:

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

Like A Priori:

- 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 A Priori:

- 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 A Priori, the supports of the frequent itemsets are stored too).
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

$\mu = 2$

<table>
<thead>
<tr>
<th>$\emptyset$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
<th>$a_8$</th>
<th>$a_9$</th>
<th>$a_{10}$</th>
<th>$a_{11}$</th>
<th>$a_{12}$</th>
<th>$a_{13}$</th>
<th>$a_{14}$</th>
<th>$a_{15}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_2$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$o_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

How many frequent patterns?
Pattern flooding

\[ \mu = 2 \]

<table>
<thead>
<tr>
<th>( \emptyset )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( a_4 )</th>
<th>( a_5 )</th>
<th>( a_6 )</th>
<th>( a_7 )</th>
<th>( a_8 )</th>
<th>( a_9 )</th>
<th>( a_{10} )</th>
<th>( a_{11} )</th>
<th>( a_{12} )</th>
<th>( a_{13} )</th>
<th>( a_{14} )</th>
<th>( a_{15} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_3 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_4 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_5 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_6 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_7 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_8 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
</tbody>
</table>

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

$\mu = 2$

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

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

$\mu = 2$

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

- 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.

## Closed and Free Patterns

Equivalence classes based on support.

<table>
<thead>
<tr>
<th>$\emptyset$</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$\emptyset_2$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$\emptyset_3$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$\emptyset_4$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$\emptyset_5$</td>
<td>×</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Closed patterns are maximal elements of each equivalence class: ABC, BC, and C.

Generators or free patterns are minimal elements (not necessarily unique) of each equivalence class: $\emptyset$, A, and B. Y. Bastide, et al. Mining frequent patterns with counting inference. SIGKDD Expl., 2000.
Closed and Free Patterns

Equivalence classes based on support.

<table>
<thead>
<tr>
<th>$\emptyset$</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$O_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$O_3$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$O_4$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$O_5$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>

- **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 $B$

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).
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**
\[ \forall \varphi_1 \leq \varphi_2, \ C(\varphi_1, D) \Rightarrow C(\varphi_2, D) \]

\[ C(\varphi, D) \equiv b \in \varphi \lor c \in \varphi \]

**Anti-monotone constraint**
\[ \forall \varphi_1 \leq \varphi_2, \ C(\varphi_2, D) \Rightarrow C(\varphi_1, D) \]

\[ C(\varphi, D) \equiv a \not\in \varphi \land c \not\in \varphi \]
Constraint properties - 2

Convertible constraints
\( \preceq \) is extended to the prefix order \( \leq \) so that \( \forall \varphi_1 \leq \varphi_2, C(\varphi_2, D) \Rightarrow C(\varphi_1, D) \)

Loose AM constraints
\( C(\varphi, D) \Rightarrow \exists e \in \varphi : C(\varphi \setminus \{e\}, D) \)

\( C(\varphi, w) \equiv \text{avg}(w(\varphi)) > \sigma \)
\( w(a) \geq w(b) \geq w(c) \geq w(d) \geq w(e) \)

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

Pei and Han – 2000

Bonchi and Lucchese – 2007
Examples

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v \in P$</td>
<td>$M$</td>
</tr>
<tr>
<td>$P \supseteq S$</td>
<td>$M$</td>
</tr>
<tr>
<td>$P \subseteq S$</td>
<td>$AM$</td>
</tr>
<tr>
<td>$\min(P) \leq \sigma$</td>
<td>$AM$</td>
</tr>
<tr>
<td>$\min(P) \geq \sigma$</td>
<td>$M$</td>
</tr>
<tr>
<td>$\max(P) \leq \sigma$</td>
<td>$M$</td>
</tr>
<tr>
<td>$\max(P) \leq \sigma$</td>
<td>$AM$</td>
</tr>
<tr>
<td>$\text{range}(P) \leq \sigma$</td>
<td>$AM$</td>
</tr>
<tr>
<td>$\text{range}(P) \geq \sigma$</td>
<td>$M$</td>
</tr>
<tr>
<td>$\text{avg}(P) \theta \sigma, \theta \in {\leq,=,\geq}$</td>
<td>Convertible</td>
</tr>
<tr>
<td>$\text{var}(w(\varphi)) \leq \sigma$</td>
<td>$LAM$</td>
</tr>
</tbody>
</table>
Outline

Introduction

Frequent Itemset Mining

Constraint-based Pattern Mining
  Constraint properties
  Algorithmic principles
Enumeration strategy

Binary partition: the element 'a' is enumerated
Enumeration strategy

Binary partition: the element 'a' is enumerated

\[ \mathcal{R}^\vee \]
\[ \mathcal{R}^\wedge \]
\[ \mathcal{R}^\vee \setminus \{a\} \]
\[ \mathcal{R}^\wedge \cup \{a\} \]

\[ a \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge \]
Constraint evaluation

Monotone constraint

\[ C(\mathcal{R}^\vee, D) \text{ is false} \]
Constraint evaluation

Monotone constraint

\[ R^\lor \land (R^\lor, D) \text{ is false} \]
Constraint evaluation

Anti-monotone constraint

\[ \mathcal{R}^\vee \]

\[ \mathcal{R}^\wedge \]

\[ \mathcal{C}(\mathcal{R}^\wedge, \mathcal{D}) \text{ is false} \]
Constraint evaluation

Anti-monotone constraint

\( \mathcal{C}(\mathcal{R}^\wedge, \mathcal{D}) \) is false
A new class of constraints

Piecewise monotone and anti-monotone constraints\(^1\)

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

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

---

\(^1\)A.k.a. primitive-based constraints


An example

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

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

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

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

Evaluation
If \( f(\mathcal{R}^\lor, \mathcal{R}^\land, \mathcal{D}) = \frac{\sum_{e \in \mathcal{R}^\lor} w(e)}{\lvert \mathcal{R}^\land \rvert} \leq \sigma \) then \( \mathcal{R} \) is empty.

Propagation
- \( \exists e \in \mathcal{R}^\lor \setminus \mathcal{R}^\land \) such that \( f(\mathcal{R}^\lor \setminus \{e\}, \mathcal{R}^\land, \mathcal{D}) \leq \sigma \), then \( e \) is moved in \( \mathcal{R}^\land \)
- \( \exists e \in \mathcal{R}^\lor \setminus \mathcal{R}^\land \) such that \( f(\mathcal{R}^\lor, \mathcal{R}^\land \cup \{e\}, \mathcal{D}) \leq \sigma \), then \( e \) is removed from \( \mathcal{R}^\lor \)
Piecewise constraint exploitation

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

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

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

1: if Check_constraints($\mathcal{R}^\wedge, \mathcal{R}^\vee$) then
2:  $(\mathcal{R}^\wedge, \mathcal{R}^\vee) \leftarrow$ Constraint_Propagation($\mathcal{R}^\wedge, \mathcal{R}^\vee$)
3:  if $\mathcal{R}^\wedge = \mathcal{R}^\vee$ then
4:     output $\mathcal{R}^\wedge$
5:  else
6:     for all $e \in \mathcal{R}^\vee \setminus \mathcal{R}^\wedge$ do
7:         Generic_CBPM_Enumeration($\mathcal{R}^\wedge \cup \{e\}, \mathcal{R}^\vee$)
8:         Generic_CBPM_Enumeration($\mathcal{R}^\wedge, \mathcal{R}^\vee \setminus \{e\}$)
9:     end for
10:    end if
11: end if
Tight Upper-bound computation

- Convex measures can be taken into account by computing some upper bounds with $R^\wedge$ and $R^\vee$.
- 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]
- $\delta$-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**:

  ‣ *constraint-based data mining*: frequency, size, …

  ‣ *background knowledge*: price, weight, aromaticity, …

  ‣ *statistics*: entropy, p-value, …

▶ **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:

- $\text{area}(X) = \text{frequency}(X) \times \text{size}(X)$ (tiling: surface)
- $\text{lift}(X_1 \rightarrow X_2) = \frac{D \times \text{frequency}(X_1X_2)}{\text{frequency}(X_2) \times \text{frequency}(X_1)}$

**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.

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A</td>
</tr>
</tbody>
</table>

- the 3 most frequent patterns: $B, E, BE^a$
- easy due to the anti-monotone property of frequency

^aOther 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.

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

- the 3 most frequent patterns: $B, E, BE$
  - 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 $area$, $k = 3$

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

Principle:

- **Cand**: the current set of the $k$ best candidate patterns
- when a candidate pattern is inserted in *Cand*, a more efficient pruning condition is deduced

$A$: lowest value of $area$ for the patterns in *Cand*

$L$: size of the longest transaction in $D$ (here: $L = 6$)

a pattern $X$ must satisfy $\text{frequency}(X) \geq \frac{A}{L}$ to be inserted in *Cand*

pruning condition according to the frequency (thus anti-monotone)

Example with a depth first search approach:

- initialization: $\text{Cand} = \{B, BE, BEC\}$
  
  $(\text{area}(BEC) = 12, \text{area}(BE) = 10, \text{area}(B) = 6)$

  $\Rightarrow \text{frequency}(X) \geq \frac{6}{6}$

- new candidate *BECD*: $\text{Cand} = \{BE, BEC, BECD\}$
  
  $(\text{area}(BECD) = 16, \text{area}(BEC) = 12, \text{area}(BE) = 10)$

  $\Rightarrow \text{frequency}(X) \geq \frac{10}{6}$ which is more efficient than $\text{frequency}(X) \geq \frac{6}{6}$

- 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 et al. IJDWM05, Papadopoulos et al. DAMI08, Soulet et al. ICDM11, van Leeuwen and Ukkonen ECML/PKDD13)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_1)</td>
<td>B E F</td>
</tr>
<tr>
<td>(t_2)</td>
<td>B C D</td>
</tr>
<tr>
<td>(t_3)</td>
<td>A E F</td>
</tr>
<tr>
<td>(t_4)</td>
<td>A B C D E</td>
</tr>
<tr>
<td>(t_5)</td>
<td>B C D E</td>
</tr>
<tr>
<td>(t_6)</td>
<td>B C D E F</td>
</tr>
<tr>
<td>(t_7)</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patterns</th>
<th>freq</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AB)</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>(AEF)</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>(B)</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>(BCDE)</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>(CDEF)</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>(E)</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

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

\(\text{Sky}(\mathcal{L}_\mathcal{I}, \{\text{freq, area}\}) = \{BCDE, BCD, B, E\}\)
## Skylines vs skypatterns

<table>
<thead>
<tr>
<th>Problem</th>
<th>Skylines</th>
<th>Skypatterns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mining task</strong></td>
<td>a set of non dominated transactions</td>
<td>a set of non dominated patterns</td>
</tr>
<tr>
<td><strong>Size of the space search</strong></td>
<td>$</td>
<td>D</td>
</tr>
<tr>
<td><strong>domain</strong></td>
<td>a lot of works</td>
<td>very few works</td>
</tr>
</tbody>
</table>

usually: $|D|<<|L|$

$D$ set of transactions
$L$ set of patterns
Skypatterns: how to process?

A naive enumeration of all candidate patterns \( \mathcal{L}_I \) 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 \( \mathcal{L}_I \).

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 $s_3$
   - 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

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

$M = \{freq, area\}$

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

Candidates =
Dominance programming:
example of the skypatterns

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

\[
M = \{ \text{freq, area} \}
\]

\[
q(X) \equiv \text{closed}_{M'}(X)
\]

Candidates = \{BCDEF, \_{s_1}\}
Dominance programming:  
example of the skypatterns

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

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

$$q(X) \equiv \text{closed}_{M'}(X) \land \neg(s_1 \succ_M X)$$

Candidates = \{BCDEF, $s_1$\}
Dominance programming:
example of the skypatterns

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

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

$q(X) \equiv \text{closed}_{M'}(X) \land \neg (s_1 \succ_{M} X)$

$Candidates = \{ \underbrace{BCDEF}_{s_1}, \underbrace{BEF}_{s_2} \}$
Dominance programming:
example of the skypatterns

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B     E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B     C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A     E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A     B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B     C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B     C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A     B C D E F</td>
</tr>
</tbody>
</table>

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

$$q(X) \equiv \text{closed}_{M'}(X) \land \neg (s_1 \succ_M X) \land \neg (s_2 \succ_M X)$$

Candidates = $\{ \text{BCDEF}, \text{BEF} \}$
Dominance programming:
example of the skypatterns

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B     E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B     C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A     E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A     B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B     C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B     C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A     B C D E F</td>
</tr>
</tbody>
</table>

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

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

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

Candidates = \{ \underbrace{BCDEF, BEF, EF, BCDE, BCD, B, E} \}

Sky($\mathcal{L}_I, M$)
Dominance programming: to sum up

The dominance programming framework encompasses many kinds of patterns:

<table>
<thead>
<tr>
<th></th>
<th>dominance relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximal patterns</td>
<td>inclusion</td>
</tr>
<tr>
<td>closed patterns</td>
<td>inclusion at same frequency</td>
</tr>
<tr>
<td>top-(k) patterns</td>
<td>order induced by the interestingness measure</td>
</tr>
<tr>
<td>skypatterns</td>
<td>Pareto dominance</td>
</tr>
</tbody>
</table>

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

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

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

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

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

$$\{ x \in \mathbb{S} \mid \text{fundamental}(x) \land \exists y_1, \ldots, y_p \in \mathbb{S}, \forall 1 \leq j \leq p, y_j \succ x \}$$

**fundamental($x$):** $x$ must satisfy a property defined by the user for example: having a minimal frequency, being closed, ...
Local patterns: examples

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>B E F</td>
</tr>
<tr>
<td>$t_2$</td>
<td>B C D</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A E F</td>
</tr>
<tr>
<td>$t_4$</td>
<td>A B C D E</td>
</tr>
<tr>
<td>$t_5$</td>
<td>B C D E</td>
</tr>
<tr>
<td>$t_6$</td>
<td>B C D E F</td>
</tr>
<tr>
<td>$t_7$</td>
<td>A B C D E F</td>
</tr>
</tbody>
</table>

**Large tiles**

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

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

**Frequent sub-groups**

$c(x) \equiv \text{freq}(x) \geq \psi_{\text{freq}} \land \not\exists \ y \in S: T_1(y) \supseteq T_1(x) \land T_2(y) \subseteq T_2(x) \land (T(y) = T(x) \Rightarrow y \subset x)$

**Skypatterns**

$c(x) \equiv \text{closed}_M(x) \land \not\exists \ y \in S: y \succ_M x$

**Frequent top-\(k\) patterns according to \(m\)**

$c(x) \equiv \text{freq}(x) \geq \psi_{\text{freq}} \land \not\exists \ y_1, \ldots, y_k \in S: \land \ m(y_j) > m(x)_{1 \leq j \leq k}$

$S = L_I$

(Mannila et al. DAMI97)
Local (optimal) patterns: examples

Large tiles
\[ c(x) \equiv \text{freq}(x) \times \text{size}(x) \geq \psi_{\text{area}} \]

Frequent sub-groups
\[ c(x) \equiv \text{freq}(x) \geq \psi_{\text{freq}} \land \nexists y \in S : \]
\[ T_1(y) \supseteq T_1(x) \land T_2(y) \subseteq T_2(x) \land (T(y) = T(x) \Rightarrow y \subset x) \]

Skypatterns
\[ c(x) \equiv \text{closed}_M(x) \land \nexists y \in S : y \succ_M x \]

Frequent top-k patterns according to \( m \)
\[ c(x) \equiv \text{freq}(x) \geq \psi_{\text{freq}} \land \nexists y_1, \ldots, y_k \in S : \]
\[ \land \ m(y_j) > m(x) \quad 1 \leq j \leq k \]

(Mannila et al. DAMI97)
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 (\(S\))**: local patterns versus pattern sets

example: \(\mathcal{I} = \{A, B\}\)

- all local patterns: \(S = \mathcal{L}_\mathcal{I} = \{\emptyset, A, B, AB\}\)
- all pattern sets:
  
  \[S = 2^{\mathcal{L}_\mathcal{I}} = \{\emptyset, \{A\}, \{B\}, \{AB\}, \{A, B\}, \{A, AB\}, \{B, AB\}, \{A, B, AB\}\}\]

**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

**Conceptual clustering (without overlapping)**

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

**Conceptual clustering with optimisation**

$$
c(x) \equiv \text{clus}(x) \land \not\exists y \in 2^{\mathcal{L}_I}, \min_{j \in [1..k]} \{\text{freq}(y_j)\} > \min_{i \in [1..k]} \{\text{freq}(x_i)\}
$$

**Pattern teams**

$$
c(x) \equiv \text{size}(x) = k \land \not\exists y \in 2^{\mathcal{L}_I}, \Phi(y) > \Phi(x)
$$
Coming back on OP (Ugarte et al. ICTAI15)

(Optimal) pattern sets of length $k$: examples

**Conceptual clustering (without overlapping)**

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

**Conceptual clustering with optimisation**

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

**Pattern teams**

$$
c(x) \equiv \text{size}(x) = k \land \nexists y \in 2^{\mathcal{I}}, \Phi(y) > \Phi(x)
$$
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)

\[ L(D, CT) : \text{the total compressed size of the encoded database and the code table:} \]

\[ L(D, CT) = L(D|CT) + L(CT|D) \]

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

\[ SI = \frac{IC}{DL} = \frac{\text{Information content}}{\text{Assimilation cost}} \]

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)

\[ \Rightarrow \text{whatever contrasts with this} = \text{subjectively interesting} \]

- 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 descripational 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)
Interactive pattern mining: overview

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

Mine

Learn ← Interact

- Provide a sample of $k$ patterns to the user (called the query $Q$)
Interactive pattern mining: overview

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

Mine

Learn ↔ Interact

Interact

▶ Like/dislike or rank or rate the patterns
Interactive pattern mining: overview

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

Generalize user feedback for building a preference model
Interactive pattern mining: overview

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

Mine

Learn ← Interact

Provide a sample of $k$ patterns benefiting from the preference model
Interactive pattern mining

Multiple mining algorithms

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

$h(\text{odor})$: 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 $I$
- Score for a pattern $X = \text{product of weights of items in } X$

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

\[
\begin{align*}
\omega_A & \times \omega_B & & \omega_C \\
AB & 4 & \times & 1 & = & 4 \\
BC & 1 & \times & 0.5 & = & 0.5
\end{align*}
\]
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:

```
<table>
<thead>
<tr>
<th>feature space</th>
<th>F_1</th>
<th>F_2</th>
<th>F_3</th>
<th>F_4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_1</td>
<td>a_2</td>
<td>a_3</td>
<td>a_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b_1</td>
<td>b_2</td>
<td>b_3</td>
<td>b_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c_1</td>
<td>c_2</td>
<td>c_3</td>
<td>c_4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>pattern space</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
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)

<table>
<thead>
<tr>
<th>pattern</th>
<th>feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>like</td>
</tr>
<tr>
<td>$AB$</td>
<td>like</td>
</tr>
<tr>
<td>$BC$</td>
<td>dislike</td>
</tr>
</tbody>
</table>
INTERACT: User feedback

How user feedback are represented?

Problem

▶ Simplicity of user feedback (binary feedback $\geq$ 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)

<table>
<thead>
<tr>
<th>pattern</th>
<th>feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>0.9</td>
</tr>
<tr>
<td>$AB$</td>
<td>0.6</td>
</tr>
<tr>
<td>$BC$</td>
<td>0.2</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>pattern</th>
<th>feedback</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>like</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td>like</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>dislike</td>
<td></td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

  \[ 2^{2-0} = 4 \quad 2^{1-1} = 1 \quad 2^{0-1} = 0.5 \]

- **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?*

<table>
<thead>
<tr>
<th>feature space</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>$F_2$</td>
<td>$F_3$</td>
<td>$F_4$</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$a_1$</td>
<td>$a_2$</td>
<td>$a_3$</td>
<td>$a_4$</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$b_3$</td>
<td>$b_4$</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$c_2$</td>
<td>$c_3$</td>
<td>$c_4$</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

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

**Diagram:**
- **A** and **B** are connected by dashed lines indicating a mapping from pattern space to feature space.
- **C** is connected to **B** with dashed lines.
- The diagram shows a mapping from pattern space to feature space with nodes labeled **A**, **B**, and **C**.
**Learn: Learning to rank**

*How to learn a model from a ranking?*

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 $Q$)?*

**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 $Q$
  - **Global diversity:** diverse patterns among the different queries $Q_i$
  - **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 $Q$?

- **Maximal Marginal Relevance:** querying diverse patterns in $Q$
  
  $$\alpha \text{Quality}(X) + (1 - \alpha) \min_{Y \in Q} \text{dist}(X, Y)$$

- **Global MMR:** taking into account previous queries
  
  $$\alpha \text{Quality}(X) + (1 - \alpha) \min_{Y \in \bigcup_{i} Q_i} \text{dist}(X, Y)$$

- **Relevance, Diversity, and Density:** querying patterns from dense regions provides more information about preferences
  
  $$\alpha \text{Quality}(X) + \beta \text{Density}(X) + (1 - \alpha - \beta) \min_{Y \in Q} \text{dist}(X, Y)$$
**MINE: Mining strategies**

What method is used to mine the pattern query \( Q \)?

**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 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 $Q$?*

**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 $Q$?

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 depend 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 depend 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

Finding all patterns from a transaction sample
⇒ input space sampling

Pattern 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 $m : \mathcal{L} \rightarrow \mathbb{R}$

- **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 $m : \mathcal{L} \rightarrow \mathbb{R}$

- **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)

Mine all frequent patterns

<table>
<thead>
<tr>
<th>TId</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_1</td>
<td>A B C</td>
</tr>
<tr>
<td>t_2</td>
<td>A B</td>
</tr>
<tr>
<td>t_3</td>
<td>B C</td>
</tr>
<tr>
<td>t_4</td>
<td>C</td>
</tr>
</tbody>
</table>

Itemset frequency:

<table>
<thead>
<tr>
<th>Itemset</th>
<th>freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>AB</td>
<td>2</td>
</tr>
<tr>
<td>AC</td>
<td>1</td>
</tr>
<tr>
<td>BC</td>
<td>2</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
</tr>
</tbody>
</table>

Pick 14 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)

Mine all frequent patterns

infeasible

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
---
\(A, A\)
\(B, B, B\)
\(C, C, C\)
\(AB, AB\)
\(AC\)
\(BC, BC\)
\(ABC\)
Two-step procedure: Toy example

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

infeasible
Mine all frequent patterns

Pick 14 itemsets

Rearrange itemsets

<table>
<thead>
<tr>
<th>TId</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A, B, C</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A, B</td>
</tr>
<tr>
<td>$t_3$</td>
<td>B, C</td>
</tr>
<tr>
<td>$t_4$</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Itemset</th>
<th>freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>AB</td>
<td>2</td>
</tr>
<tr>
<td>AC</td>
<td>1</td>
</tr>
<tr>
<td>BC</td>
<td>2</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Items</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, A</td>
<td>1</td>
</tr>
<tr>
<td>B, B, B</td>
<td>2</td>
</tr>
<tr>
<td>C, C, C</td>
<td>3</td>
</tr>
<tr>
<td>AB, AB</td>
<td>2</td>
</tr>
<tr>
<td>AC</td>
<td>1</td>
</tr>
<tr>
<td>BC, BC</td>
<td>2</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TId</th>
<th>Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A, B, C, AB, AC, BC, ABC</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A, B, AB</td>
</tr>
<tr>
<td>$t_3$</td>
<td>A, B, AB</td>
</tr>
<tr>
<td>$t_4$</td>
<td>B, C, BC</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>
Two-step procedure: Toy example

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

infeasible
Mine all frequent patterns

<table>
<thead>
<tr>
<th>TId</th>
<th>Items</th>
<th>weight $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>A B C</td>
<td>$2^3 - 1 = 7$</td>
</tr>
<tr>
<td>$t_2$</td>
<td>A B</td>
<td>$2^2 - 1 = 3$</td>
</tr>
<tr>
<td>$t_3$</td>
<td>B C</td>
<td>$2^2 - 1 = 3$</td>
</tr>
<tr>
<td>$t_4$</td>
<td>C</td>
<td>$2^1 - 1 = 1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Itemset</th>
<th>freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>AB</td>
<td>2</td>
</tr>
<tr>
<td>AC</td>
<td>1</td>
</tr>
<tr>
<td>BC</td>
<td>2</td>
</tr>
<tr>
<td>ABC</td>
<td>1</td>
</tr>
</tbody>
</table>

1. Pick a transaction proportionally to $\omega$

2. Pick an itemset uniformly

Pick 14 itemsets
Two-step procedure: Comparison

Offline processing

<table>
<thead>
<tr>
<th>Slow</th>
<th>Two-step procedure</th>
<th>Fast</th>
</tr>
</thead>
</table>

MH method

Online processing

Complexity depends on the measure $m$:

<table>
<thead>
<tr>
<th>Measure $m(X)$</th>
<th>Preprocessing</th>
<th>$k$ realizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{supp}(X, D)$</td>
<td>$O(</td>
<td>I</td>
</tr>
<tr>
<td>$\text{supp}(X, D) \times</td>
<td>X</td>
<td>$</td>
</tr>
<tr>
<td>$\text{supp}_+(X, D) \times (</td>
<td>D^-</td>
<td>- \text{supp}_-(X, D))$</td>
</tr>
<tr>
<td>$\text{supp}(X, D)^2$</td>
<td>$O(</td>
<td>I</td>
</tr>
</tbody>
</table>

Preprocessing time may be prohibitive
Two-step procedure: Comparison

Complexity depends on the measure $m$:

<table>
<thead>
<tr>
<th>Measure $m(X)$</th>
<th>Preprocessing</th>
<th>$k$ realizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$supp(X, \mathcal{D})$</td>
<td>$O(</td>
<td>\mathcal{I}</td>
</tr>
<tr>
<td>$supp(X, \mathcal{D}) \times</td>
<td>X</td>
<td>$</td>
</tr>
<tr>
<td>$supp_+(X, \mathcal{D}) \times (</td>
<td>\mathcal{D}^-</td>
<td>- supp_-(X, \mathcal{D}))$</td>
</tr>
<tr>
<td>$supp(X, \mathcal{D})^2$</td>
<td>$O(</td>
<td>\mathcal{I}</td>
</tr>
</tbody>
</table>

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 individual 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 ⇒ itemset is enough.
    - Some chemicals have the same 2-d graph representation and totally different odor qualities (e.g., isomers) ⇒ 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)
An efficient algorithm for mining frequent sequence with constraint programming.

Origami: Mining representative orthogonal graph patterns.

Nir Ailon.
An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami.
Mining association rules between sets of items in large databases.

Stefano Bistarelli and Francesco Bonchi.
Interestingness is not a dichotomy: Introducing softness in constrained pattern mining.
Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti.
Free-sets: A condensed representation of boolean data for the approximation of
frequency queries.

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

Behrouz Babaki, Tias Guns, and Siegfried Nijssen.
Constrained clustering using column generation.
In International Conference on AI and OR Techniques in Constriant Programming

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

Tijl De Bie.
Maximum entropy models and subjective interestingness: an application to tiles
in binary databases.

Tijl De Bie.
Subjective interestingness in exploratory data mining.
In Advances in Intelligent Data Analysis XII - 12th International Symposium,
Abdelhamid Boudane, Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi.
A sat-based approach for mining association rules.

Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli.
Fast generation of best interval patterns for nonmonotonic constraints.

Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli.
Revisiting pattern structure projections.

Mario Boley, Maike Krause-Traudes, Bo Kang, and Björn Jacobs.
Creedoscalable and repeatable extrinsic evaluation for pattern discovery systems by online user studies.

Francesco Bonchi and Claudio Lucchese.
Extending the state-of-the-art of constraint-based pattern discovery.

Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner.
Direct local pattern sampling by efficient two-step random procedures.
Mario Boley, Sandy Moens, and Thomas Gärtner.
Linear space direct pattern sampling using coupling from the past.

Mansurul Bhuiyan, Snehasis Mukhopadhyay, and Mohammad Al Hasan.
Interactive pattern mining on hidden data: a sampling-based solution.

Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel.
One click mining: Interactive local pattern discovery through implicit preference and performance learning.

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.


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.

Moonjung Cho, Jian Pei, Haixun Wang, and Wei Wang.
Preference-based frequent pattern mining.

Toon Calders, Christophe Rigotti, and Jean-François Boulicaut.
A survey on condensed representations for frequent sets.

Ming-Wei Chang, Lev-Arie Ratinov, Nicholas Rizzolo, and Dan Roth.
Learning and inference with constraints.

Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tou Ng.
A lattice-based approach to query-by-example spoken document retrieval.

James Cussens.
Bayesian network learning by compiling to weighted MAX-SAT.
Duen Horng Chau, Jilles Vreeken, Matthijs van Leeuwen, and Christos Faloutsos, editors.

Tijl De Bie.
Subjective interestingness in exploratory data mining.

Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, and Luc De Raedt.
Interactive learning of pattern rankings.

Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut.
Granularity of co-evolution patterns in dynamic attributed graphs.

Vladimir Dzyuba and Matthijs van Leeuwen.
Learning what matters—sampling interesting patterns.

Vladimir Dzyuba, Matthijs van Leeuwen, and Luc De Raedt.
Flexible constrained sampling with guarantees for pattern mining.

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.

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.

Johannes Fürnkranz and Eyke Hüllermeier.
Preference Learning.

Ada Wai-Chee Fu, Renfrew W.-w. Kwong, and Jian Tang.
Mining $N$-most interesting itemsets.

Frédéric Flouvat, Jérémy Sanhes, Claude Pasquier, Nazha Selmaoui-Folcher, and Jean-François Boulicaut.
Improving pattern discovery relevancy by deriving constraints from expert models.


Arianna Gallo, Tijl De Bie, and Nello Cristianini.


Floris Geerts, Bart Goethals, and Taneli Mielikäinen.


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


Liqiang Geng and Howard J Hamilton.


Bart Goethals, Sandy Moens, and Jilles Vreeken.


Arnaud Giacometti and Arnaud Soulet.


Arnaud Giacometti and Arnaud Soulet.


Arnaud Giacometti and Arnaud Soulet.


Bernhard Ganter and Rudolf Wille.


Bart Goethals and Mohammed Javeed Zaki, editors.

Jiawei Han and Yongjian Fu.

Wilhelmiina Hämäläinen and Matti Nykänen.


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

Microsoft research Redmond, WA, 2009.

Mohammad Al Hasan and Mohammed J. Zaki.


Tomasz Imielinski and Heikki Mannila.


Thorsten Joachims.


Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi.

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


Mehdi Khiari, Patrice Boizumault, and Bruno Crémilleux.


Arno J. Knobbe and Eric K. Y. Ho.


Amina Kemmar, Samir Loudni, Yahia Lebbah, Patrice Boizumault, and Thierry Charnois.

Jerry Kiernan and Evimaria Terzi.


Sergei O. Kuznetsov.


B. Liu, W. Hsu, and Y. Ma.


Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski.


Sandy Moens and Mario Boley.

Proper database covers instantly by recursive tile sampling.

Sandy Moens and Bart Goethals.


Randomly sampling maximal itemsets.

Marianne Mueller and Stefan Kramer.


Shinichi Morishita and Jun Sese.


Dominance programming for itemset mining.

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

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

Exploratory mining and pruning optimizations of constrained associations rules.

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


Constraint-based pattern mining.

Siegfried Nijssen and Albrecht Zimmermann.


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


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


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


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


Luc De Raedt, Tias Guns, and Siegfried Nijssen.
Stefan Rueping.


Luc De Raedt and Albrecht Zimmermann.


Xuehua Shen and ChengXiang Zhai.

Active feedback in ad hoc information retrieval.
Hannu Toivonen et al.


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.


Willy Ugarte, Patrice Boizumault, Bruno Crémilleux, Alban Lepailleur, Samir Loudni, Marc Plantevit, Chedy Raïssi, and Arnaud Soulet.

Takeaki Uno.


Matthijs van Leeuwen.


Matthijs van Leeuwen, Tijl De Bie, Eirini Spyropoulou, and Cédric Mesnager.


Matthijs van Leeuwen and Arno J. Knobbe.


Matthijs van Leeuwen and Antti Ukkonen.


Jilles Vreeken and Arno Siebes.
Dong Xin, Hong Cheng, Xifeng Yan, and Jiawei Han.


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

