Preference-based Pattern Mining

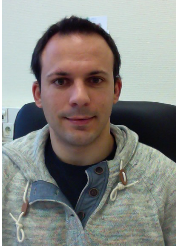
GDR-IA - GT Caviar

Bruno Crémilleux - GREYC CNRS UMR 6072

Orléans - May 27, 2019

Genesis of the talk and acknowledgments

With M. Plantevit and A. Soulet:
tutorial at ECML/PKDD 2016, ICFCA 2017, BDA 2017



Marc Plantevit
Univ. Lyon

Arnaud Soulet
Univ. Tours

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:

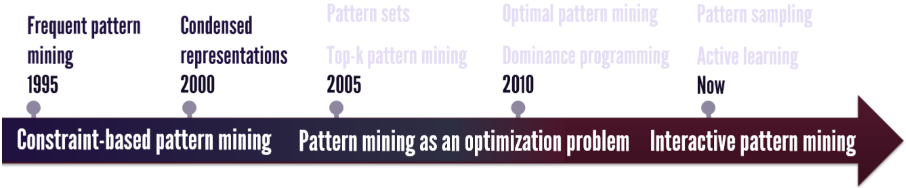
- automatically discover novel insights about the domain in which the data was measured
- use machine discoveries to synergistically boost human expertise
- e.g., understanding commonalities and differences among PET scans of Alzheimer's patients.

Roadmap

- constraint-based pattern mining
  (➥ not a tutorial, the goal is to illustrate the limits)
- pattern mining as an optimization problem
- interactive pattern mining

Each part:

- a period of the data mining story
- a take-home message



Constraint-based pattern mining: the toolbox and its limits

Data mining task: an example
Contrast patterns (1/2)

	d_1	d_2	d_3	d_4	d_5
mol_1	X				X
mol_2	X	X	X		X
mol_3				X	
mol_4	X		X		
mol_5	X		X	X	
mol_6	X		X		X
mol_7					X
mol_8		X			
mol_9	X	X			X
mol_{10}	X	X			

2 classes:

T: toxic

NT: non toxic

X : pattern
example: {d1,d2}

{d1,d2}: present/supported by chemicals [2,9,10]

Frequency:
$F(\{d1, d2\}) = 3$

Contrast patterns (2/2)

	d_1	d_2	d_3	d_4	d_5
mol_1	X				X
mol_2	X	X	X		X
mol_3				X	
mol_4	X		X		
mol_5	X		X	X	
mol_6	X		X		X
mol_7					X
mol_8		X			
mol_9	X	X			X
mol_{10}	X	X			

GR ("growth rate") to quantify a contrast:

$$GR_T(X) = \frac{|NT| \times F(X,T)}{|T| \times F(X,NT)}$$

{d1,d3} is present in:

- the toxic chemicals [2,4,5]
- the non-toxic chemicals [6]

$$GR_T(\{d1,d3\}) = \frac{5 \times 3}{5 \times 1} = 3$$

Emerging pattern: $GR_{clas}(X) \geq mingr$ (a constraint)

goal: given $mingr$, mining all emerging patterns.

What is the difficulty?

Let us consider a very simple description of chemicals:
➥ n binary descriptors (presence/absence of molecular fragments)

What is the size of the search space?

What is the difficulty?

Let us consider a very simple description of chemicals:
➡ n binary descriptors (presence/absence of molecular fragments)

What is the size of the search space? 2^n (it is easy to get huge...)

Example of computation time:

(1 micro-second is required to process one data)

Taille (n)	$log_2 n$	n	$n log_2 n$	n^2	2^n
10	3×10^{-6}	10×10^{-6}	30×10^{-6}	100×10^{-6}	10^{-3}
100	7×10^{-6}	100×10^{-6}	700×10^{-6}	0.01	
1000	10×10^{-6}	10^{-3}	0.01	1	
10 000	13×10^{-6}	0.01	0.13	1.7 minute	
100 000	17×10^{-6}	0.1	1.7	2.8 hours	

What is the difficulty?

Let us consider a very simple description of chemicals:

➥ n binary descriptors (presence/absence of molecular fragments)

What is the size of the search space? 2^n (it is easy to get huge...)

Example of computation time:

(1 micro-second is required to process one data)

Taille (n)	$log_2 n$	n	$nlog_2 n$	n^2	2^n
10	3×10^{-6}	10×10^{-6}	30×10^{-6}	100×10^{-6}	10^{-3}
100	7×10^{-6}	100×10^{-6}	700×10^{-6}	0.01	10^{14} centuries
1000	10×10^{-6}	10^{-3}	0.01	1	
10 000	13×10^{-6}	0.01	0.13	1.7 minute	
100 000	17×10^{-6}	0.1	1.7	2.8 hours	

What is the difficulty?

Let us consider a very simple description of chemicals:

➥ n binary descriptors (presence/absence of molecular fragments)

What is the size of the search space? 2^n (it is easy to get huge. . .)

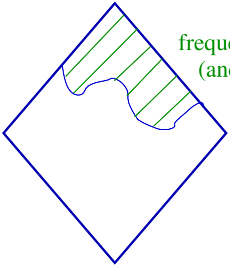
Example of computation time:

(1 micro-second is required to process one data)

Taille (n)	$log_2 n$	n	$nlog_2 n$	n^2	2^n
10	3×10^{-6}	10×10^{-6}	30×10^{-6}	100×10^{-6}	10^{-3}
100	7×10^{-6}	100×10^{-6}	700×10^{-6}	0.01	10^{14} centuries
1000	10×10^{-6}	10^{-3}	0.01	1	astronomic
10 000	13×10^{-6}	0.01	0.13	1.7 minute	astronomic
100 000	17×10^{-6}	0.1	1.7	2.8 hours	astronomic

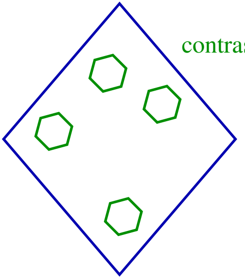
Heikki Mannila: "data mining is the art of counting"

Must we explore the whole search space?



➥ no

(it can be still expensive)



➥ a priori yes

Why contrast patterns do not follow anti-monotonicity property?

when a pattern is specialized, both the numerator and denominator decrease, but the numerator as well as the denominator can decrease the fastest

"solution": pruning according to "branch and bound"

Speeding the algorithms

- **pattern condensed representations** (Calders et al. Constraint-Based Mining and Inductive Databases 2004): do not count the frequency of all patterns (the frequencies of other patterns are deduced from the computed frequencies) ➠ equivalence classes
- **the FIM Era:** FIMI[1] Workshop@ICDM, 2003 and 2004
  - 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 was to earn ms according to their competitors!

What about the end-user (and the pattern interestingness)?
➥ constraints are a partial answer.

[1] Frequent Itemset Mining Implementations

Constraint-based data mining

Constraints are needed for:

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

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

Classes of constraints

There are several classes of constraints (e.g. (anti-)monotone, convertible, succinct constraints).

A large class of constraints: a lot of constraints can be decomposed into several pieces that are either monotone or anti-monotone.

- *primitive-based constraints* (Soulet et al. PAKDD 2005)
- *piecewise monotone and anti-monotone constraints* (Cerf at al. SDM 2008)
- *projection-antimonotonicity* (Buzmakov et al. ECML/PKDD 2015)

The “secret” of constraint-based pattern mining

- deduce (anti-)monotone constraints from the whole query
- take benefit from intervals/spaces where patterns have a same value according to interestingness measures
  ➥ pattern condensed representations

Toward declarativity

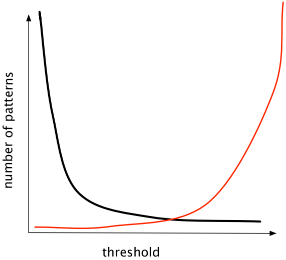
Why declarative approaches?

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

Declarative approaches:

- CP approaches (De Raedt et al. KDD 2008, Khiari et al. CP 2010, Guns et al. TKDE 2013, Dao et al. ECML/PKDD 2013, Dao et al AIJ 2017, Aoga et al. Constraints 2017,. . .)
- SAT approaches (Boudane et al. IJCAI 2016, Jabbour et al. CIKM 2013, Jabbour et al. PAKDD 2017, Dao et al IJCAI-ECAI 2018,. . .)
- ILP approaches (Mueller et al DS 2010, Babaki et al. CPAIOR 2014, Ouali et al. IJCAI 2016,. . .)
- ASP approaches (Gebser et al. IJCAI 2016,. . .)

Thresholding problem



- a too stringent threshold:
  trivial patterns
- a too weak threshold:
  too many patterns, unmanageable
  and diversity not necessary assured

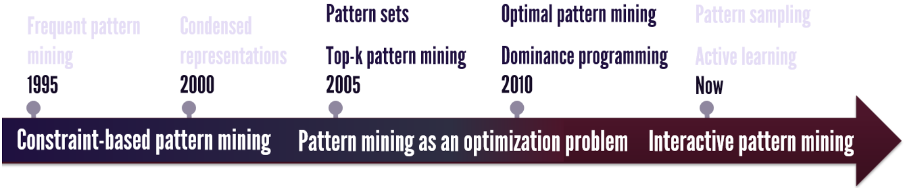
- some attempts to tackle this issue:
  - interestingness is not a dichotomy (Bistarelli and Bonchi ECML/PKDD 2005)
  - taking benefit from hierarchical relationships (Han and Fu TKDE 1999, Desmier et al. IDA 2014)
- but setting thresholds remains an issue in pattern mining.

Constraint-based pattern mining: issues

- 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 support the user to define relevant constraints independently of the pruning strategies used by the algorithms? how to design the proper constraints/preferences?

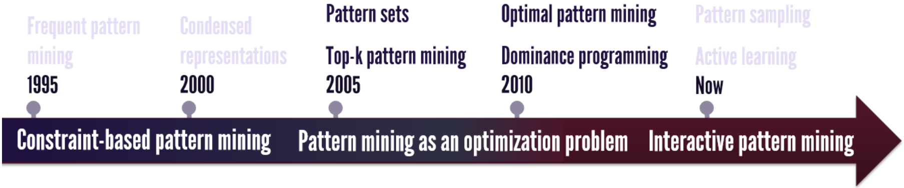
Take home message 1:

the need of preferences in pattern mining



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 talk

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

  in the next part: preferences are implicit

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

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

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

    With qualitative preferences: two patterns can be incomparable.

Measures

Many works on:

- interestingness measures (Geng et al. ACM Computing Surveys 2006)
- utility functions (Yao and Hamilton DKE 2006)
- statistically significant rules (Hämäläinen and Nykänen ICDM 2008)

Examples:

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

Putting the pattern mining task to an optimization problem

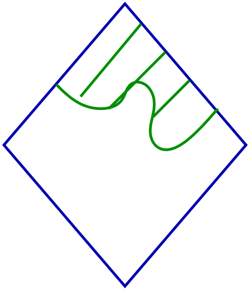
The most interesting patterns according to measures/preferences:

- **free/closed patterns** (Boulicaut et al. DAMI 2003, Bastide et al. SIGKDD Explorations 2000)
  ➥ given an equivalent class, I prefer the shortest/longest patterns
- **one measure**: top-k patterns (Fu et al. Ismis 2000, Jabbour et al. ECML/PKDD 2013)
- **several measures**: how to find a trade-off between several criteria?
  ➥ skyline patterns (Cho et al. IJDWM 2005, Soulet et al. ICDM 2011, van Leeuwen and Ukkonen ECML/PKDD 2013)
- **dominance programming** (Negrevergne et al. ICDM 2013), **optimal patterns** (Ugarte et al. ICTAI 2015)
- **subjective interest/interest according to a background knowledge** (De Bie DAMI 2011)

top-k pattern mining: an example

Goal: finding the k patterns maximizing an interestingness measure.

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



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

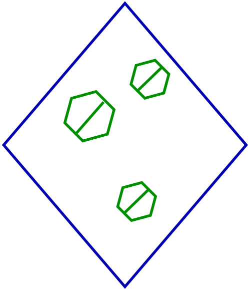
  ➥ easy due to the anti-monotone property of frequency

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

top-k pattern mining: an example

Goal: finding the k patterns maximizing an interestingness measure.

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



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

  ➥ easy due to the anti-monotone property of frequency

- the 3 patterns maximizing area: $BCDE$, BCD, CDE

  ➥ branch & bound

  (Zimmermann and De Raedt MLJ09)

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

top-k pattern mining
an example of pruning condition

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

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

Principle:

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

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

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

a pattern X must satisfy $frequency(X) \geq \frac{A}{L}$ to be inserted in $\mathcal{C}and$

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

Example with a depth first search approach:

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

top-k pattern mining in a nutshell

Advantages:

- compact
- threshold free
- best patterns

Drawbacks:

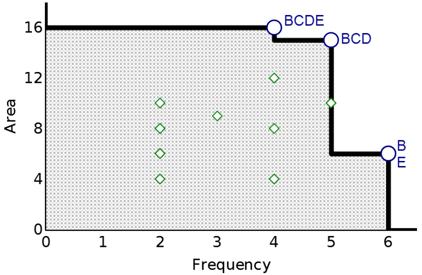
- complete resolution is costly, sometimes heuristic search (beam search)
  (van Leeuwen and Knobbe DAMI 2012)
- diversity issue: top-k patterns are often very similar
- several criteria must be aggregated
  ➥ skylines patterns: a trade-off between several criteria

Skypatterns (Pareto dominance)

Notion of skylines (database) in pattern mining (Cho at al. IJDWM 2005, Papadopoulos et al. DAMI 2008, Soulet et al. ICDM 2011, van Leeuwen and Ukkonen ECML/PKDD 2013)

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

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



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

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

Skylines vs skypatterns

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

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

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

Skypatterns: how to process?

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

Two approaches:

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

Dominance programming

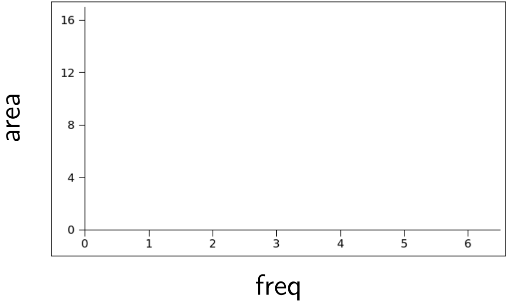
Dominance: a pattern is optimal if it is not dominated by another.
Skypatterns: dominance relation = Pareto dominance

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

Principle: increasingly reduce the dominance area by processing pairwise comparisons between patterns. Methods using Dynamic CSP (Negrevergne et al. ICDM 2013, Ugarte et al. CPAIOR 2014, AIJ 2017).

Dominance programming: example of the skypatterns

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



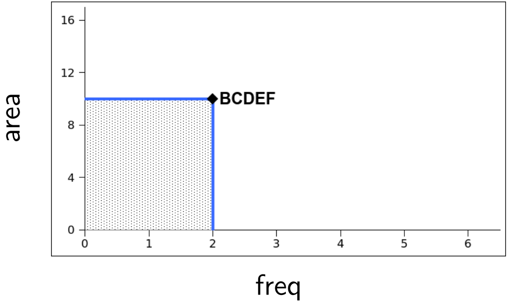
$$M = \{freq, area\}$$

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

$$Candidates =$$

Dominance programming: example of the skypatterns

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



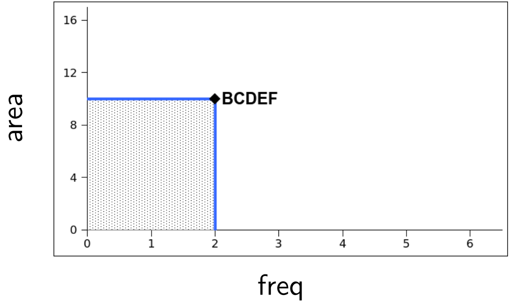
$$M = \{freq, area\}$$

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

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

Dominance programming: example of the skypatterns

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



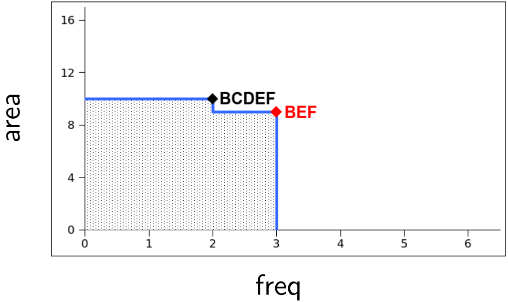
$$M = \{freq, area\}$$

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

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

Dominance programming: example of the skypatterns

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



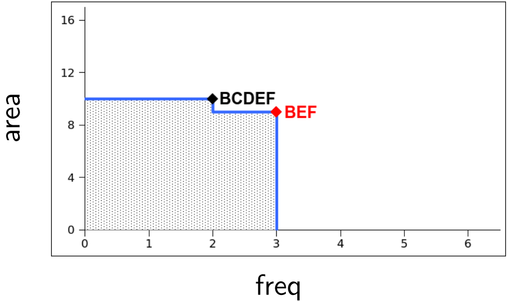
$$M = \{freq, area\}$$

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

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

Dominance programming: example of the skypatterns

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



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

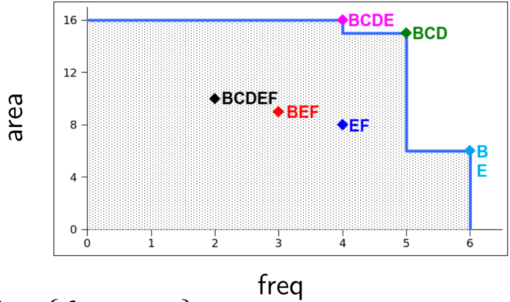
$$q(X) \equiv closed_{M'}(X) \wedge \neg(s_1 \succ_M X) \wedge \neg(s_2 \succ_M X)$$

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

Dominance programming: example of the skypatterns

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

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



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

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

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

Dominance programming: to sum up

The dominance programming framework encompasses many kinds of patterns:

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

maximal patterns $\subseteq$ closed patterns

top-k patterns $\subseteq$ skypatterns

A step further

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

➥ preference-based optimal patterns

In the following:
(1) define preference-based optimal patterns,
(2) show how many tasks of local patterns fall into this framework,
(3) deal with optimal pattern sets (not given in this talk).

Preference-based optimal patterns

A preference $\triangleright$ is a strict partial order relation on a set of patterns $\mathbb{S}$.
$x \triangleright y$ indicates that x is preferred to y

(Ugarte et al. ICTAI 2015): a pattern x is optimal (OP) according to $\triangleright$ iff $\nexists y_1, \ldots y_p \in \mathbb{S}, \forall 1 \leq j \leq p,\ y_j \triangleright x$

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

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

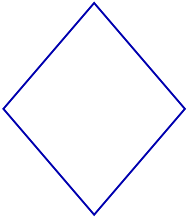
$$\left\{ x \in \mathbb{S} \mid \ \texttt{fundamental}(x) \ \wedge \nexists y_1, \ldots y_p \in \mathbb{S}, \forall 1 \leq j \leq p,\ y_j \triangleright x \right\}$$

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

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

Local patterns: examples

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



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

(Mannila et al. DAMI 1997)

Large tiles

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

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

Frequent sub-groups

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

Skypatterns

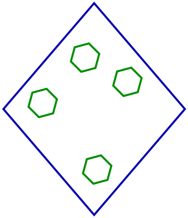
$$c(x) \equiv \quad \mathtt{closed}_M(x) \wedge \nexists\, y \in \mathbb{S} : y \succ_M x$$

Frequent top-k patterns according to m

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

Local (optimal) patterns: examples

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



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

(Mannila et al. DAMI 1997)

Large tiles

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

Frequent sub-groups

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

Skypatterns

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

Frequent top-k patterns according to m

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

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. SDM 2006)

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

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

Many usages:

- characterizing the differences and the norm between given components in the data - DIFFNORM (Budhathoki and Vreeken ECML/PKDD 2015)
- causal discovery (Budhathoki and Vreeken ICDM 2016)
- missing values (Vreeken and Siebes ICDM 2008)
- handling sequences (Bertens et al. KDD 2016)
- ...

and many other works on data compression/summarization (e.g. Kiernan and Terzi KDD 2008),...

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

Pattern mining as an optimization problem: concluding remarks

In the approaches indicated in this part:

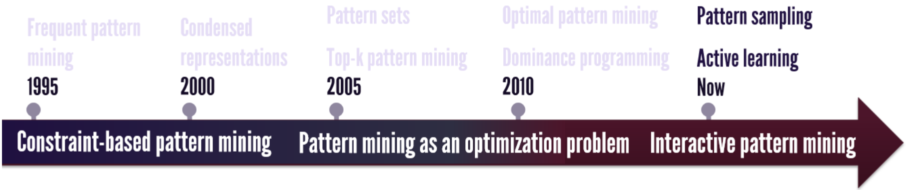
- measures/preferences are explicit and must be given by the user... (but there is no threshold :-)
- diversity issue: top-k patterns are often very similar
- complete approaches (optimal w.r.t the preferences):
  ➥ stop completeness "Please, please stop making new algorithms for mining *all* patterns"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

Take home message 2:

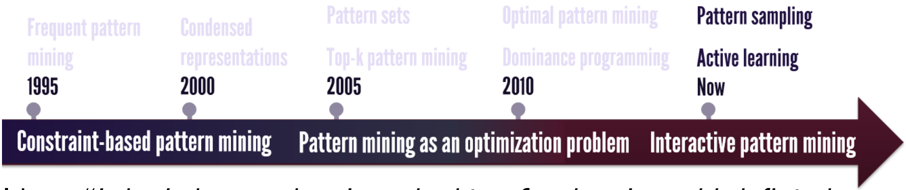
pattern mining can also be an optimization problem

(in this part, with explicit 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."*

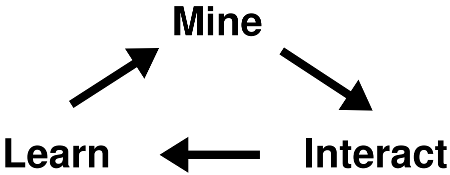
➠ preference acquisition

In this part:

- easier: no user-specified parameters (constraint, threshold or measure)
- better: learn user preferences from user feedback
- faster: instant pattern discovery (otherwise the user is discouraged)

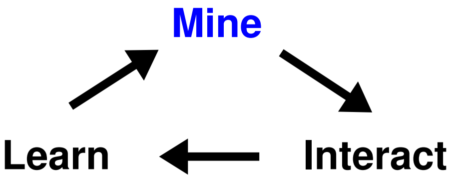
Interactive pattern mining: overview

Interactive data exploration using pattern mining (Van Leeuwen 2014)



Interactive pattern mining: overview

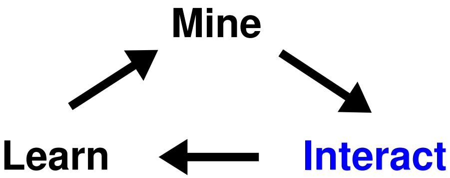
Interactive data exploration using pattern mining (Van Leeuwen 2014)



Mine: provide a sample of k patterns to the user (called the query $\mathcal{Q}$)

Interactive pattern mining: overview

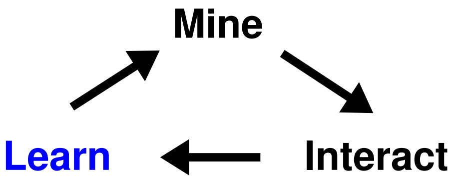
Interactive data exploration using pattern mining (Van Leeuwen 2014)



Interact: like/dislike or rank or rate the patterns user

GREYC

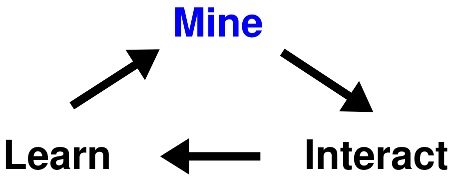
Interactive data exploration using pattern mining (Van Leeuwen 2014)



Learn: generalize user feedback for building a preference model

Interactive pattern mining: overview

Interactive data exploration using pattern mining (Van Leeuwen 2014)



Mine (again!): provide a sample of k patterns benefiting from the preference model

Interactive pattern mining: example: characterizing fraudsters

$\mathcal{D}^1$: unknown set of data prefered by the user.

We assume that the user knows if a given pattern is relevant or not w.r.t. $\mathcal{D}^1$

Goal: mining all patterns characterizing $\mathcal{D}^1$

what the user wants:

Trans.	Items				Classe
t_1	A	B			E	1
t_2	A	B				1
t_3		B	C	D		0
t_4		B	C			0

(Giacometti and Soulet IDA 2017)

Interactive pattern mining: example: characterizing fraudsters

$\mathcal{D}^1$: unknown set of data prefered by the user.

We assume that the user knows if a given pattern is relevant or not w.r.t. $\mathcal{D}^1$

Goal: mining all patterns characterizing $\mathcal{D}^1$

what the data are:

Trans.	Items					Classe
t_1	A	B			E	
t_2	A	B				
t_3		B	C	D		
t_4		B	C			

(Giacometti and Soulet IDA 2017)

Interactive pattern mining: example: characterizing fraudsters

$\mathcal{D}^1$: unknown set of data prefered by the user.

We assume that the user knows if a given pattern is relevant or not w.r.t. $\mathcal{D}^1$

Goal: mining all patterns characterizing $\mathcal{D}^1$

what we propose:

Trans.	Items				w
t_1	A	B			E	1
t_2	A	B				1
t_3		B	C	D		0
t_4		B	C			0

(Giacometti and Soulet IDA 2017)

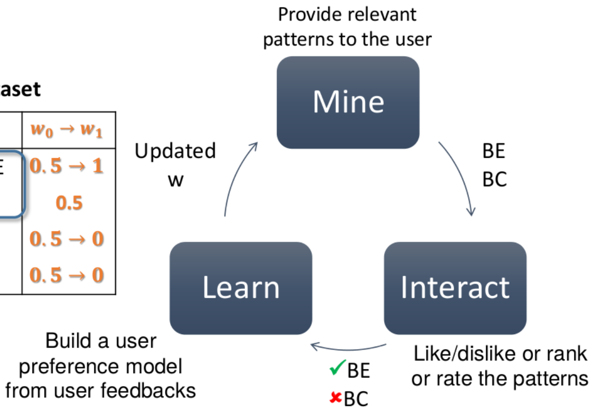
Learning preferences within the interactive pattern mining framework

[van Leeuwen, 2014]

How?

Unlabelled dataset

	Trans.	Items	$w_0 \rightarrow w_1$
$\mathcal{D}^1$	t1	A B E	$0.5 \rightarrow 1$
$\mathcal{D}^1$	t2	A B	0.5
	t3	B C D	$0.5 \rightarrow 0$
	t4	B C	$0.5 \rightarrow 0$



Source: Hewasinghage et al. EGC 2017

Challenges in this example

1. **A two-way learning problem:**
   - *system to user:* the user learns new knowledge from the database through the patterns provided by the system (frequent patterns in $\mathcal{D}^1$)
     ➥ fast extraction and quality of patterns are needed to maintain a satisfactory interaction
   - *user to system:* the system learns the user preferences (here represented by weights of items) from her feedback
     ➥ diversity is needed to discover the user preferences

2. **Which patterns to propose:**
   pattern sampling according to (Boley et al. KDD 2011): fast and random (which guarantees a good diversity)

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?

Research problem:

- expressivity of the preference model
- ease of learning of the preference model

Weighted product model:

- a weight on items $\mathcal{I}$
- score for a pattern X = product of weights of items in X
- (Bhuiyan et al. CIKM 2012, Dzyuba et al. PAKDD 2017)

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

Learn (preference model): how user preferences are represented?

Research problem:

- expressivity of the preference model
- ease of learning of the preference model

Feature space model:

- features:
  - assumption about the user preferences
  - the more, the better
- examples:
  - expected and measured frequency (Xin et al. KDD 2006)
  - attributes, coverage, chi-squared, length and so on (Dzyuba et al. ICTAI 2013)
- mapping between a pattern X and a set of features

Interact (user feedback): how user feedback are represented?

Research 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. CIKM 2012, Dzyuba et al. PAKDD 2017)

pattern	feedback
A	like
AB	like
BC	dislike

Interact (user feedback): how user feedback are represented?

Research problem:

- simplicity of user feedback (binary feedback $>$ graded feedback)
- accuracy of user feedback (binary feedback $<$ graded feedback)

Feature space model:

- ordered feedback (ranking) (Xin et al. KDD 2006, Dzyuba et al. ICTAI 2013)

$$A \succ AB \succ BC$$

- graded feedback (rate) (Rueping ICML 2009)

pattern	feedback
A	0.9
AB	0.6
BC	0.2

Learn (preference learning method): how user feedback are generalized to a model?

Weighted product model:

Counting likes and dislikes for each item: $\omega = \beta^{(\text{\#like - \#dislike})}$

(Bhuiyan et al. ICML 2012, Dzyuba et al. PAKDD 2017)

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

Feature space model: ➠ learning to rank

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 KDD 02), AdaRank (Xu et al. SIGIR 07),...

Learn (active learning problem): how are selected the set of patterns (query $\mathcal{Q}$)?

Research problem:

- mining the most relevant patterns according to *Quality*
- querying patterns that provide more information about preferences

Heuristic criteria:

- **local diversity:** diverse patterns among the current query $\mathcal{Q}$
- **global diversity:** diverse patterns among the different queries $\mathcal{Q}_i$ (i.e. taking into account the story of the queries)
- **density:** dense regions are more important

Learn (preference learning method): what method is used to mine the pattern query $\mathcal{Q}$

GREYC

Research problem:

- instant discovery for facilitating the iterative process
- preference model integration for improving the pattern quality
- pattern diversity for completing the preference model

Approaches:

- post-processing: re-ranking the patterns with the updated quality (Rueping ICML 2009, Xin et al. KDD 2006) ; clustering as heuristic for improving the local diversity (Xin et al. KDD 2006)
- optimal pattern mining: beam search based on reweighing subgroup quality measures for finding the best patterns (Dzyuba et al. ICTAI 2013)
- pattern sampling: (Bhuiyan et al. CIKM 2012, Dzyuba et al. PAKDD 2017): randomly draw pattern with a distribution proportional to their updated quality, then sampling as heuristic for diversity and density

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. DS 2014)

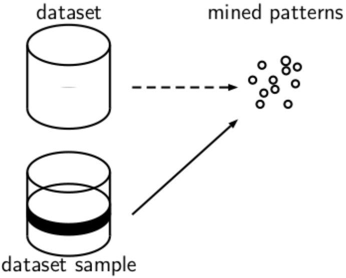
➠ few seconds between the query and the answer

Methods:

- ~~sound and complete pattern mining~~
- beam search subgroup discovery methods
- Monte Carlo tree search (Bosc et al. ECML/PKDD 2016)
- pattern sampling

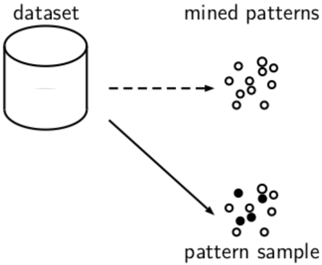
Dataset sampling vs pattern sampling

Dataset sampling



Finding all patterns from a transaction sample
➠ input space sampling

Pattern sampling



Finding a pattern sample from all transactions
➠ output space sampling

Two-step procedure: toy example

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

infeasible

~~Mine all frequent patterns~~

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

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

Pick 14 itemsets

Direct sampling

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

Two-step procedure: toy example

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

infeasible

~~Mine all frequent patterns~~

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

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

1. Pick a transaction proportionally to ω

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

2. Pick an itemset uniformly

Pattern sampling based on SAT

(Dzyuba et al. DMKD 2017)

Principle:

- used samplers based on random hash functions and XOR-sampling from the SAT community
- the sampling combines strong constraints (XOR constraints dividing the search space into 2^n cells) and a weak constraint (e.g. weighting w.r.t the frequency)

Method:

- a cell (here numbered 101) is defined by a set of n XOR constraints:

$$X_1 \otimes X_2 \otimes X_4 = 1$$
$$X_0 \otimes X_1 \otimes X_3 \otimes X_4 = 0$$
$$X_0 \otimes X_2 \otimes X_4 = 1$$

X_i belongs to the pattern language

- draw a pattern from the patterns satisfying the XOR constraints
  ➠ add it to the sample
- update the set of XOR constraints, repeat

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.

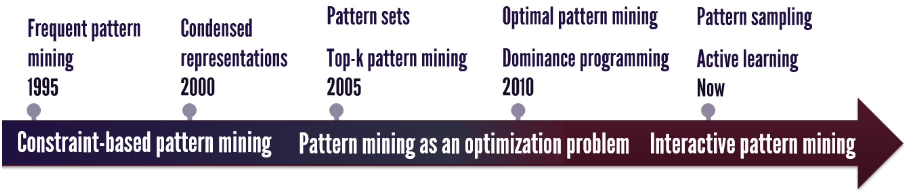
Take home message 3:

I don't know what I am looking for...

➥ interactive pattern mining

(➠ preference acquisition)

Preference-based pattern mining

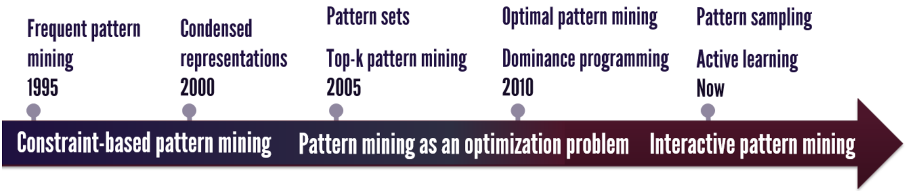


User preferences are more and more prominent...

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

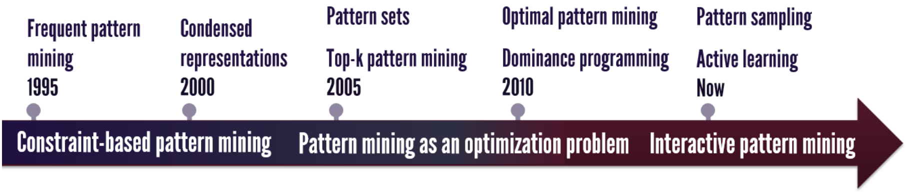


User preferences are more and more prominent...

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...

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

Pattern mining in the AI field

- **cross-fertilization between data mining and constraint programming/SAT/ILP** (De Raedt et al. KDD 2008): 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:

Tijl de Bie (Ghent University, Belgium)

Albert Bifet (Télécom ParisTech, Paris)

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

Wouter Duivesteijn (Ghent University, Belgium
& 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)