Mining rare sequential patterns with Answer Set Programming (ASP)

Ahmed Samet[1] **Thomas Guyet**[2] Benjamin Negrevergne[3]

[1] Rennes University/IRISA
[2] Agrocampus-Ouest/IRISA
[3] LAMSADE, Université Paris-Dauphine

ILP 2017 –

Motivation: toward declarative sequence analytics (I)

Sequential data – Patient care pathway

- Timestamped events (hospital stays, drugs deliveries, consultations)
- Sequences descriptions (patient sex & age, chronic long-term illness)
- background knowledge about drugs and diseases (taxonomies)

FOL facts encoding:

```
patient(0).

sex(0,m).
birthYear(0,1960).

delivery(0,1,3585053,1).
delivery(0,1,3599730,1).
delivery(0,154,3599730,1).
delivery(0,346,3599730,1).
delivery(0,350,3599730,2).

disease(0,380,g409).
disease(0,380,k700).
```

Sequential data analysis

Use of data analytics techniques to extract meaningful information from sequences (*e.g.* care pathways) to carry epidemiological studies

- statistics
- pattern mining (e.g. *sequential pattern mining*)
- visual analytics tools

Declarative pattern mining with ASP

Declarative pattern mining

- Pattern mining: extracting knowledge hidden in structured data
- Declarative pattern mining: use of a declarative programming to encode a pattern mining task
  - Constraint programming [GDN+17, NG15]
  - SAT encoding [JSS15]
  - Logic programming [GGQ+16, KAP15]

ASP – Answer Set Programming

- Expressive first order logic syntax, including
  - constraints on atoms sets (aggregates)
  - optimization directives
- Efficient state-of-the-art solvers (Potassco suite – clingo solver [GKK+11])
- Existing work on pattern mining with ASP
  - → frequent itemset mining [Jär11]
  - → frequent sequential pattern mining [GGQ+16]

Contributions

- This work is part of a research project aiming at developing a **versatile sequential pattern mining approach based on ASP**
  - versatile: offer flexibility to easily specify a (complex) data analysis based on pattern mining (*elaboration tolerant*)
  - sequential: data is set of events sequences
  - ASP: we explore Answer Set Programming as declarative programming paradigm
- **existing sequential pattern mining ASP encodings** [GQM+17] mainly based on the notion of frequent sequential patterns

$\Rightarrow$ The focus of this article is the encoding of **rare sequential patterns** mining

- rare sequential patterns
- non-zero-rare sequential patterns
- minimal rare sequential patterns
- constrained rare sequential patterns

Challenge: encoding sequential pattern mining tasks

Mining **rare** sequential patterns

- It raises interesting encoding problems
  - the naive encoding of rarity is not efficient: it requires specific encodings
  - too many patterns: study of an efficient condensed representation
- It is new sequential pattern mining task to illustrate the expressive power of declarative pattern mining
  - few approach to extract rare itemsets [KR16]
  - rare sequential patterns not addressed in literature

Rare sequential patterns objectives

- data analysis: extracting exceptional behaviors in dense datasets
- data cleaning: identify unexpected behaviors to remove from the database or to correct

Sequential pattern mining backgrounds

- A sequence $S = \langle s_i \rangle$ is an ordered list of items (could be itemsets)
  - $\rightarrow$ $\langle d, a, b, c \rangle$: sequence of 4 items
  - $\rightarrow$ encoded facts for sequence 10:

    seq(10,**d**,1) . **seq**(10,**a**,2) . **seq**(10,**b**,3) . **seq**(10,**c**,4) .
- A sequential pattern $p = \langle p_i \rangle_{1 \leq i \leq n}$ is a sequence of length n
- Mining sequential patterns in a sequences dataset, $\mathcal{S}$, consists in finding out all **interesting** patterns that occurs in the database

SID	sequence
10	$\langle d, a, b, c \rangle$
20	$\langle a, c, b, c \rangle$
30	$\langle a, b, c \rangle$
40	$\langle a, c, b \rangle$
50	$\langle a, c \rangle$

Sequential pattern mining backgrounds

- A sequence $S = \langle s_i \rangle$ is an ordered list of items (could be itemsets)
  - → $\langle d, a, b, c \rangle$: sequence of 4 items
  - → encoded facts for sequence 10:

    seq(10,**d**,1) . **seq**(10,**a**,2) . **seq**(10,**b**,3) . **seq**(10,**c**,4) .
- A sequential pattern $p = \langle p_i \rangle_{1 \leq i \leq n}$ is a sequence of length n
- Mining sequential patterns in a sequences dataset, $\mathcal{S}$, consists in finding out all **interesting** patterns that occurs in the database
- A pattern p occurs in sequence s when p is a sub-sequence of s
  - → example: occurrences of $p = \langle abc \rangle$

SID	sequence
10	$\langle d, a, b, c \rangle$
20	$\langle a, c, b, c \rangle$
30	$\langle a, b, c \rangle$
40	$\langle a, c, b \rangle$
50	$\langle a, c \rangle$

Sequential pattern mining backgrounds

- A sequence $S = \langle s_i \rangle$ is an ordered list of items (could be itemsets)
  - → $\langle d, a, b, c \rangle$: sequence of 4 items
  - → encoded facts for sequence 10:

    seq(10,**d**,1) . **seq**(10,**a**,2) . **seq**(10,**b**,3) . **seq**(10,**c**,4) .
- A sequential pattern $p = \langle p_i \rangle_{1 \leq i \leq n}$ is a sequence of length n
- Mining sequential patterns in a sequences dataset, $\mathcal{S}$, consists in finding out all **interesting** patterns that occurs in the database
- A pattern p occurs in sequence s when p is a sub-sequence of s
  - → example: occurrences of $p = \langle abc \rangle$

SID	sequence
10	$\langle d, a, b, c \rangle$
20	$\langle a, c, b, c \rangle$
30	$\langle a, b, c \rangle$
40	$\langle a, c, b \rangle$
50	$\langle a, c \rangle$

Interesting patterns?

- Frequent patterns: occur in at least σ sequences of $\mathcal{S}$
- Rare patterns: occur in at most σ sequences of $\mathcal{S}$

$$supp(\boldsymbol{p}) = |\{\boldsymbol{s} \in \mathcal{S} | \boldsymbol{s} \preceq p\}| < \sigma$$

Rare sequential patterns and minimal rare patterns

Let $\sigma \in \mathbb{N}^+$ be a frequency threshold and $\mathcal{S} = \{\boldsymbol{s}^i\}$ a dataset of sequences.

$$supp(\boldsymbol{p}) = |\{\boldsymbol{s} \in \mathcal{S} | \boldsymbol{s} \preceq p\}|$$

- Rare patterns
$$RP = \{\boldsymbol{p} | supp(\boldsymbol{p}) < \sigma\}$$
- Non-zero-rare patterns
$$RP = \{\boldsymbol{p} | supp(\boldsymbol{p}) > 0 \wedge supp(\boldsymbol{p}) < \sigma\}$$
- Minimal rare patterns
$$mRP = \{\boldsymbol{p} | supp(\boldsymbol{p}) < \sigma \wedge \forall \boldsymbol{p}' \prec \boldsymbol{p}, supp(\boldsymbol{p}') \geq \sigma\}$$

Rare sequential patterns and minimal rare patterns

	s^{10}	s^{20}	s^{30}	s^{40}	s^{50}
Seq.	$\langle d, a, b, c \rangle$	$\langle a, c, b, c \rangle$	$\langle a, b, c \rangle$	$\langle a, c, b \rangle$	$\langle a, c \rangle$

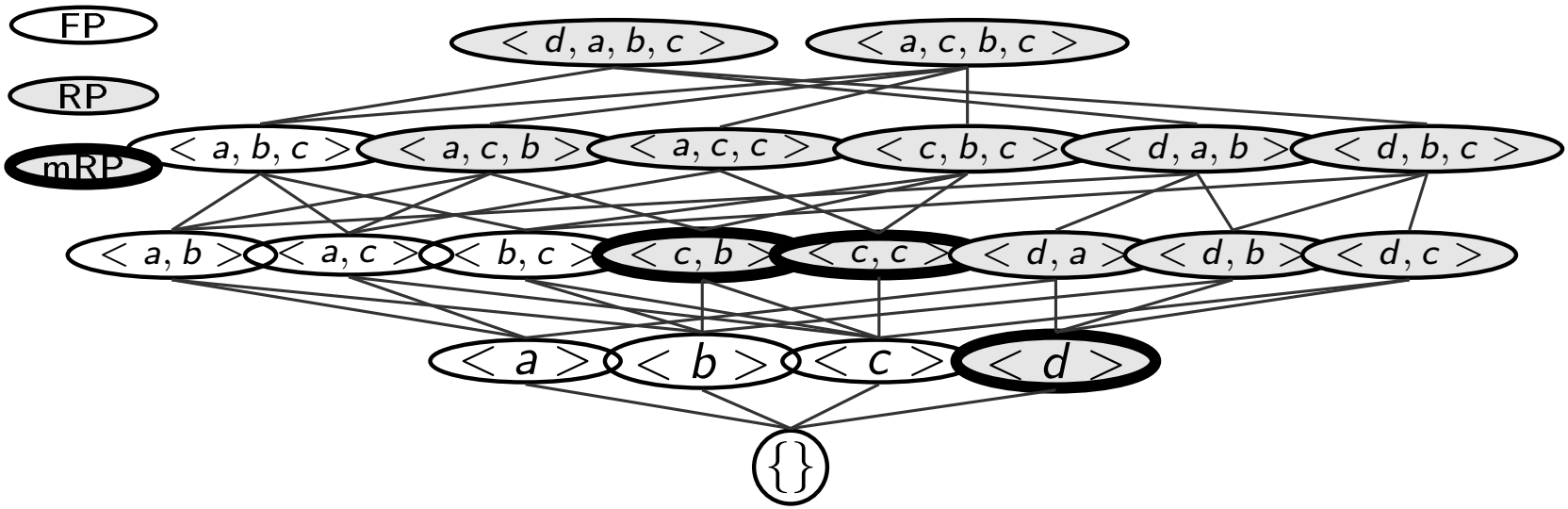


Figure: Lattice of sequential patterns of dataset $\mathcal{D}$. Zero-rare patterns are omitted for sake of clarity. In white: frequent patterns, in grey: rare patterns, surrounded: minimal rare patterns for $\sigma = 2$.

A glance at the encodings from [GGQ+16]

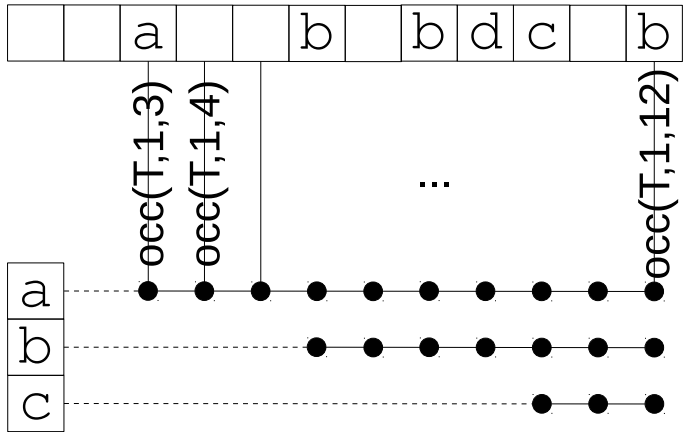
Atom	Meaning
seq(s,x,e)	e is an event at position x in sequence $\boldsymbol{s}$
item(e)	item e belongs to some $\boldsymbol{t} \in \mathcal{D}$
patpos(x)	$1 \leq x$ refers to the position x of an item p_x in pattern $\boldsymbol{p}$
pat(x,e)	$p_x = e$ is the item at position x in pattern $\boldsymbol{p}$
support(s)	$\langle p_i \rangle_{1 \leq i \leq m} \sqsubseteq \langle s_j \rangle_{1 \leq j \leq n}$, that is, $\boldsymbol{p} \sqsubseteq \boldsymbol{s}$
occ(s,l,x)	$\langle p_i \rangle_{1 \leq i \leq x} \sqsubseteq \langle s_j \rangle_{1 \leq j < l}$, where $1 \leq l \leq n+1$

```
1  item(I) :- seq(_, _,I).
2  seqlen(S,L) :- seq(S,L,_), not seq(S,L+1,_).
3
4  patpos(1).
5  { patpos(X+1) } :- patpos(X), X<maxlen.
6  patlen(L) :- patpos(L), not patpos(L+1).
7  1 { pat(X,I) : item(I) } 1 :- patpos(X).
8
9  occ(S,1,P) :- pat(1,I), seq(S,P,I).
10 occ(S,L,P) :- occ(S, L,   P-1), seq(S,P,_).
11 occ(S,L,P) :- occ(S, L-1, P-1), seq(S,P,C), pat(L,C).
12
13 support(S) :- occ(S, L, LS), patlen(L), seqlen(S,LS).
```

$\Rightarrow$ enumerate all sequential patterns and their supported sequences

Embedding encoding

```
8  ...
9  occ(S,1,P):- pat(1,I), seq(S,U,I).
10 occ(S,L,P):- occ(S, L,   U-1), seq(S,U,_).
11 occ(S,L,P):- occ(S, L-1, U-1), seq(S,U,C), pat(L,C).
12 ...
```



- Encoding strategy named “fill-gaps" (vs “skip-gaps")
- occ(s,x,u): $\langle p_i \rangle_{1 \leq i \leq x} \sqsubseteq \langle s_j \rangle_{1 \leq j < u}$
- a pattern item that has been "recognized" at position u is at position $u + 1$
- the item at the position x of the pattern is recognized at position u when the items at position $x - 1$ has been recognized and s_u holds p_x.

Encoding rare pattern constraints

Naive encoding of the rarity constraint

```
:- # count { S : support ( S ) } >= th .
```

Improved version of the rarity constraint

- if a prefix-pattern of length l is rare, then all patterns of length l', $l' > l$, are rare
- rare(l) means that the subpattern $\langle p_j \rangle_{1 \leq j \leq l}$ is rare
- evaluates independently the support of each of pattern prefixes

```
14 rare(IL) :- IL=1..L, patlen(L),
15             #count{ S : occ(S, IL, LS), seqlen(S,LS) } < th.
16 rare(L)  :- rare(L-1), L<=PL, patlen(PL).
17 :- not rare(L), patlen(L).
18 :- not support(S) : seq(S,_,_).
```

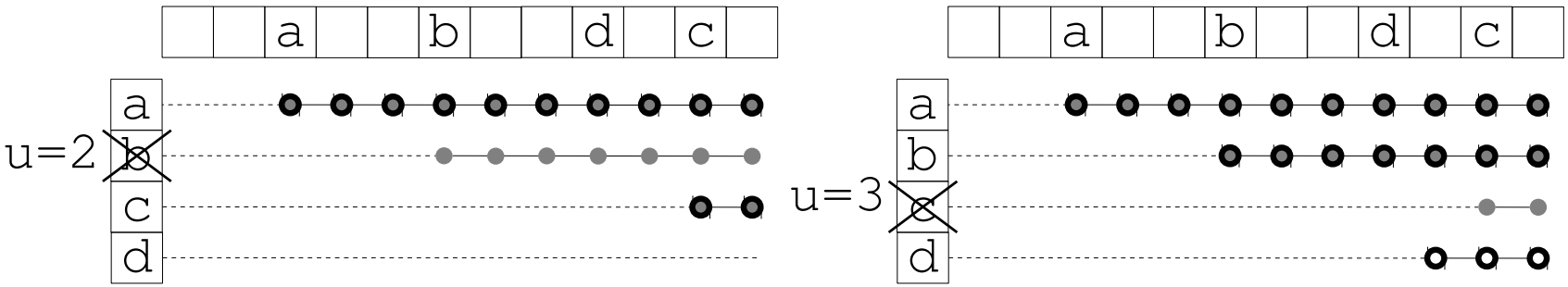
Minimal rare patterns I

Minimal rare patterns

- a pattern of length n is minimal rare iff any sub-pattern is rare
- it is sufficient to evaluate rarity of subpattern of length $n-1$

Encoding principle

- evaluate the rarity for each subpattern of length $n-1$
  - $\rightarrow$ **sprare**$(\mathbf{u}, \mathbf{l})$ subpattern $\langle p_i \rangle_{1 \leq i \leq l \wedge i \neq u}$ is rare
- require to evaluate support for all subpattern in the same answer set
- encoding trick: subpattern embeddings are similar before position u (not recomputed)



Minimal rare patterns II

Atom	Meaning
suppat(u)	u is an identifier of a sub-pattern $u = \langle p_i \rangle_{1 \leq i \leq n,\ i \neq u}$
spocc(s,u,l,x)	$\langle u_i \rangle_{1 \leq i \leq x} \sqsubseteq \langle s_j \rangle_{1 \leq j < l}$, where $u \leq l \leq n+1$
sprare(u,x)	$\langle u_i \rangle_{1 \leq i \leq x}$ is rare

```
20 suppat(U) :- U=1..L, patlen(L), L>1.
21
22 spocc(S,1,2,P) :- seq(S,P,I), pat(2,I), not support(S).
23 spocc(S,U,U,P) :- suppat(U), occ(S,U-1,P), not support(S).
24 spocc(S,U,L  ,P+1) :- spocc(S,U,L,P), seq(S,P+1,_).
25 spocc(S,U,L+1,P+1) :- spocc(S,U,L,P), seq(S,P+1,I),
26                       pat(L+1,I).
27
28 sprare(U,U-1):- suppat(U).
29 sprare(U,L)  :- sprare(U, L-1), L<=LP, patlen(LP).
30 sprare(U, IL):- suppat(U), IL=U+1..L, patlen(L),
31                #count{ S: spocc(S,U,IL,LS), seqlen(S,LS);
32                        S: support(S) } < th.
33 :- sprare(U,L), patlen(L).
```

Experiments and results

Experiments

- Runtime and memory efficiency were evaluated on synthetic datasets
- Flexibility of the declarative approach has been evaluated on real dataset

Source codes of the ASP programs, of algorithms and of the dataset generator are available at

`https://sites.google.com/site/aspseqmining/`

Experiments and results

Experiments

- Runtime and memory efficiency were evaluated on synthetic datasets
  - Controlled generation of synthetic datasets (same principle as IBM Quest Synthetic Data Generator)
  - Comparison with adapted Apriori-Rare [Sza14] and MRG-Exp [SVN10] algorithm for sequential pattern mining (encoded in Matlab)

  $\Rightarrow$ expected results: MRSM has better results (better than procedural approach), exponential increasing wrt threshold and sequence lengths Results
- Flexibility of the declarative approach has been evaluated on real dataset

Source codes of the ASP programs, of algorithms and of the dataset generator are available at

`https://sites.google.com/site/aspseqmining/`

Experiments and results

Experiments

- Runtime and memory efficiency were evaluated on synthetic datasets
  - Controlled generation of synthetic datasets (same principle as IBM Quest Synthetic Data Generator)
  - Comparison with adapted Apriori-Rare [Sza14] and MRG-Exp [SVN10] algorithm for sequential pattern mining (encoded in Matlab)

  $\Rightarrow$ expected results: MRSM has better results (better than procedural approach), exponential increasing wrt threshold and sequence lengths Results
- Flexibility of the declarative approach has been evaluated on real dataset
  - dataset from care pathway analysis
  - evaluate pattern number reduction under additional constraints

Source codes of the ASP programs, of algorithms and of the dataset generator are available at

`https://sites.google.com/site/aspseqmining/`

Case study: care pathway analytics I

Case study

- Care pathways exposed to anti-epileptic drugs who had epileptic seizures (inhospital diagnosis)
- Events are drug deliveries
  - 500 patients
  - total amount of $101,793$ events, with $|\mathcal{I}| = 3,671$.
- Motivations of rare patterns:
  - exclude patients with rare disease from a cohort datasets
  - identify possible unknown adverse drugs reactions

Preliminary result

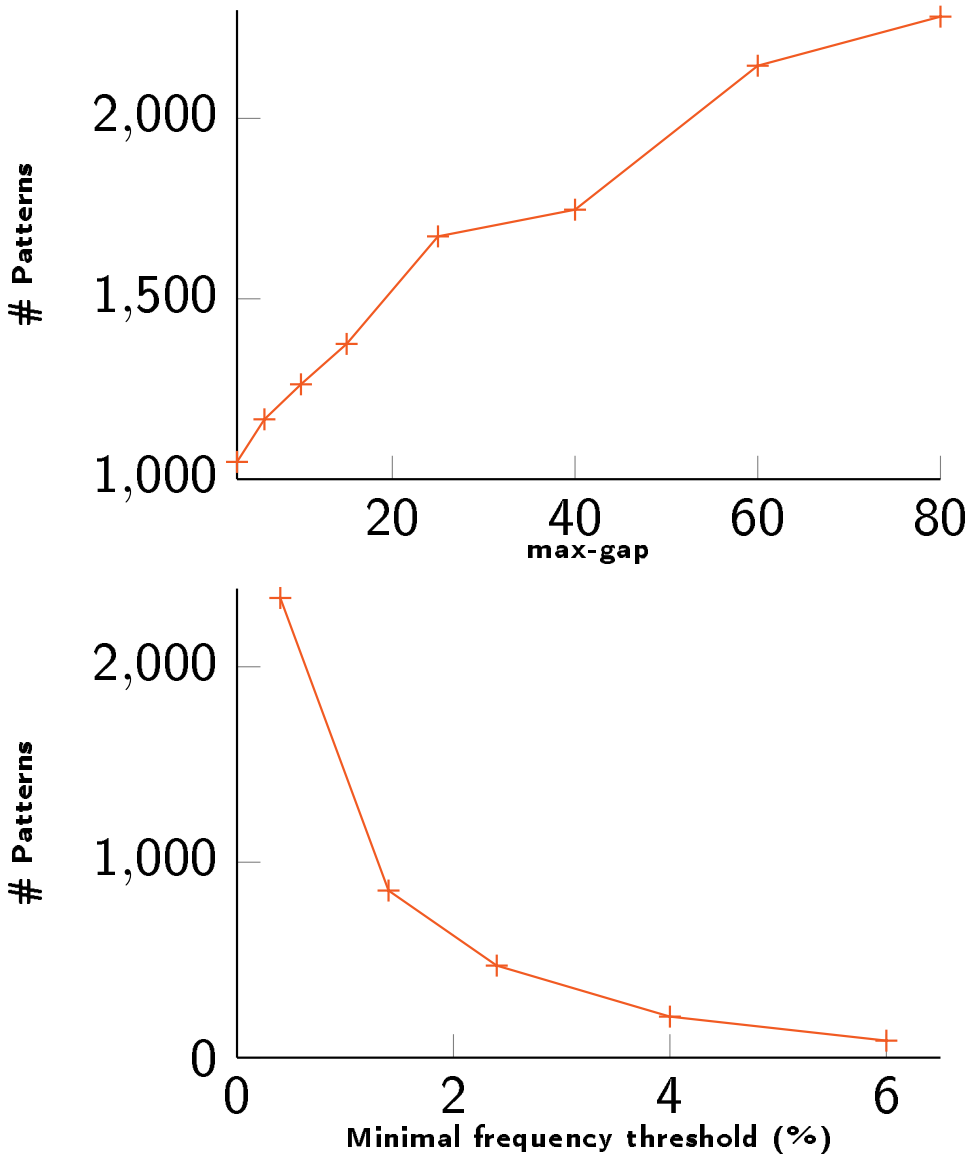
$\sigma = 10\%$: $7,758$ mRPs of length at most 3 ($maxlen = 3$)

$\rightarrow$ too many pattern to be analyzed

Explore additional constraints

- selection of almost-rare patterns ($f_{min} \leq supp(\boldsymbol{p}) \leq \sigma$)
- use of a *maxgap* constraints (from 3 to 25 events)

Case study: care pathway analytics II



- smaller maxgap constraints reduces the pattern number
- minimal frequency threshold reduces exponentially the pattern number

$\sigma = 10\%$, $maxlen = 3$, $f_{min} = 5\%$ and $maxgap = 3$

$\rightarrow$ pattern number $= 133$

Conclusion and perspectives

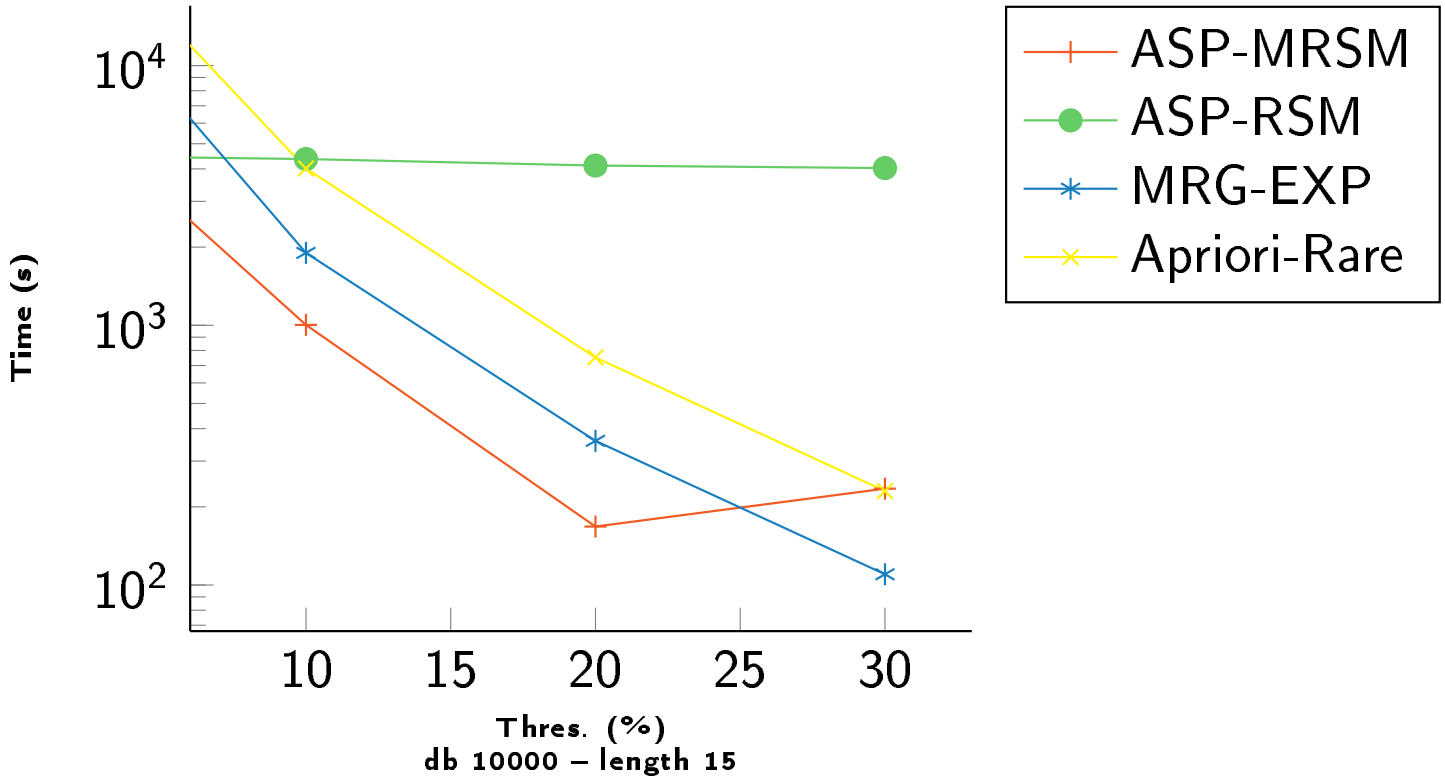
Conclusions

- we introduced (minimal) rare sequential pattern mining
- we proposed ASP encodings for extracting them
- experiments shown that ASP encoding are better on (not tuned) procedural algorithms
  - $\rightarrow$ required computing resources prevent from processing large datasets
- a case study illustrates the use of additional constraint to reduce the number of patterns
  - practical case study with tries/errors pattern mining task design
  - illustrate the effective versatility of our declarative approach

Perspectives

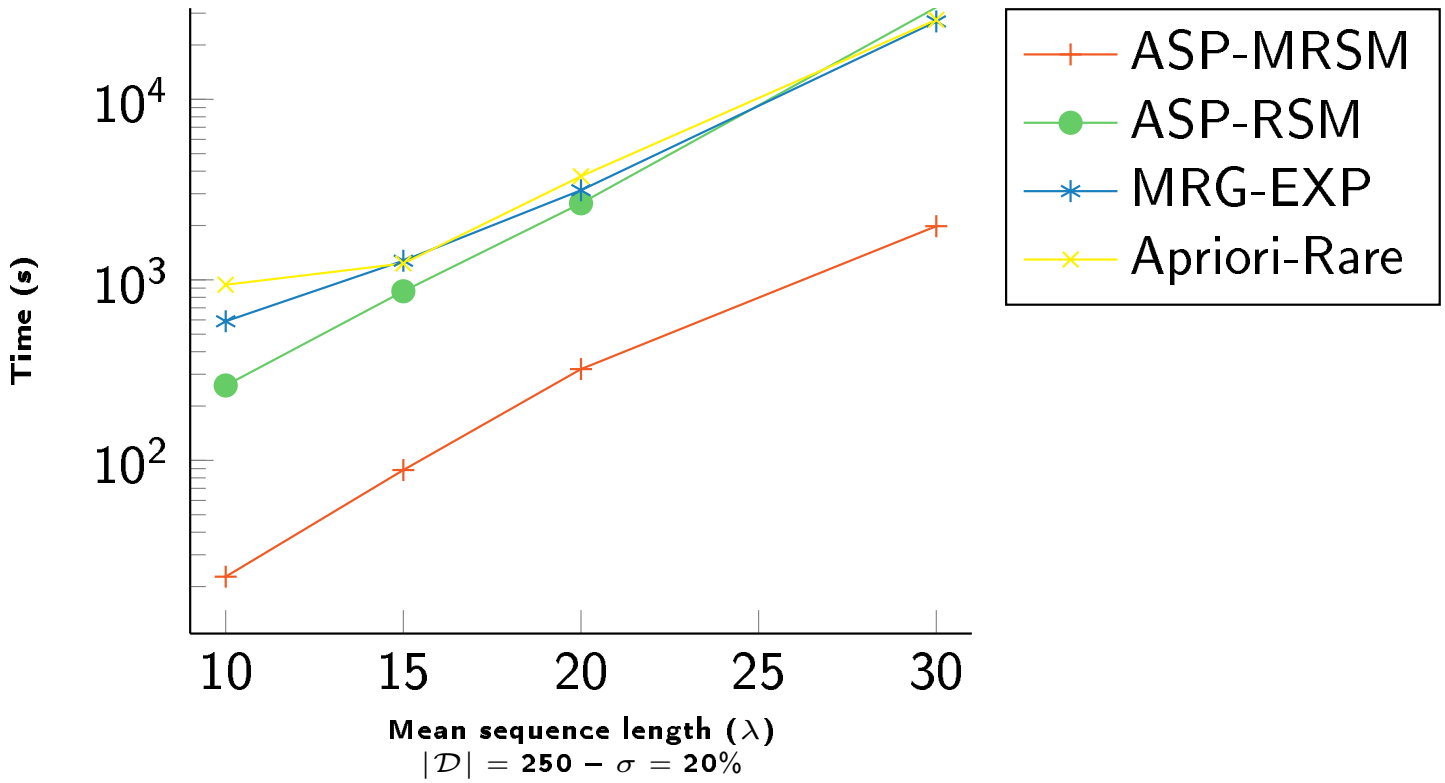
- Integrate our various encoding in an easy to use framework
- Using hybrid-ASP for improving resources consumptions (and increase processable data size)

Synthetic datasets results back



- Exponential increase of time wrt threshold
- ASP-RSM is constant: most of the time is spend to enumerate patterns and the number of pattern does not evolve ($\approx 10^5$ patterns)
- ASP-MRSM is the most efficient approach but require more memory ($\approx 10^4$ patterns)

Synthetic datasets results back II



- exponential increase of time wrt mean sequence length

References I

Tias Guns, Anton Dries, Siegfried Nijssen, Guido Tack, and Luc De Raedt, *Miningzinc: A declarative framework for constraint-based mining*, Artif. Intell. 244 (2017), 6–29.

Martin Gebser, Thomas Guyet, René Quiniou, Javier Romero, and Torsten Schaub, *Knowledge-based sequence mining with ASP*, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), 2016, pp. 1497–1504.

M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider, *Potassco: The Potsdam answer set solving collection*, AI Communications 24 (2011), no. 2, 107–124.

Thomas Guyet, René Quiniou, Yves Moinard, Schaub, and Torsten, *Efficiency analysis of ASP encodings for sequential pattern mining tasks*, Advances in Knowledge Discovery and Management 8 (2017), to appear.

Matti Järvisalo, *Itemset mining as a challenge application for answer set enumeration*, Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning, 2011, pp. 304–310.

Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi, *Decomposition based SAT encodings for itemset mining problems*, Proceeding of the Pacific-Asia Conference Advances on Knowledge Discovery and Data Mining, Part II, 2015, pp. 662–674.

Nikos Katzouris, Alexander Artikis, and Georgios Paliouras, *Incremental learning of event definitions with inductive logic programming*, Machine Learning 100 (2015), no. 2-3, 555–585.

Yun Sing Koh and Sri Devi Ravana, *Unsupervised rare pattern mining: A survey*, Transactions on Knowledge Discovery from Data 10 (2016), no. 4, 1–29.

References II

Benjamin Negrevergne and Tias Guns, *Constraint-based sequence mining using constraint programming*, Proceedings of the International Conference on Integration of AI and OR Techniques in Constraint Programming, 2015, pp. 288–305.

Laszlo Szathmary, Petko Valtchev, and Amedeo Napoli, *Generating rare association rules using the minimal rare itemsets family*, International Journal on Software and Informatics 4 (2010), no. 3, 219–238.

Laszlo Szathmary, *Finding minimal rare itemsets with an extended version of the Apriori algorithm*, Proceedings of the International Conference on Applied Informatics, vol. 1, 2014, pp. 85–92.