Mining rare sequential patterns with Answer Set Programming (ASP)

Ahmed Samet\textsuperscript{1} \hspace{1cm} \textbf{Thomas Guyet}\textsuperscript{2} \hspace{1cm} Benjamin Negrevergne\textsuperscript{3}

\textsuperscript{1} Rennes University/IRISA
\textsuperscript{2} Agrocampus-Ouest/IRISA
\textsuperscript{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

⇒ 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)
  - $\langle d, a, b, c \rangle$: 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, $S$, consists in finding out all interesting patterns that occurs in the database

<table>
<thead>
<tr>
<th>SID</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$\langle d, a, b, c \rangle$</td>
</tr>
<tr>
<td>20</td>
<td>$\langle a, c, b, c \rangle$</td>
</tr>
<tr>
<td>30</td>
<td>$\langle a, b, c \rangle$</td>
</tr>
<tr>
<td>40</td>
<td>$\langle a, c, b \rangle$</td>
</tr>
<tr>
<td>50</td>
<td>$\langle a, c \rangle$</td>
</tr>
</tbody>
</table>
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, $S$, 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 $p = \langle abc \rangle$

<table>
<thead>
<tr>
<th>SID</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$\langle d, a, b, c \rangle$</td>
</tr>
<tr>
<td>20</td>
<td>$\langle a, c, b, c \rangle$</td>
</tr>
<tr>
<td>30</td>
<td>$\langle a, b, c \rangle$</td>
</tr>
<tr>
<td>40</td>
<td>$\langle a, c, b \rangle$</td>
</tr>
<tr>
<td>50</td>
<td>$\langle a, c \rangle$</td>
</tr>
</tbody>
</table>
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:
  $\text{seq}(10,d,1)$, $\text{seq}(10,a,2)$, $\text{seq}(10,b,3)$, $\text{seq}(10,c,4)$.

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, $S$, consists in finding out all interesting patterns that occur in the database

A pattern $p$ occurs in sequence $s$ when $p$ is a sub-sequence of $s$

- example: occurrences of $p = \langle abc \rangle$

### Interesting patterns?

- **Frequent patterns**: occur in at least $\sigma$ sequences of $S$
- **Rare patterns**: occur in at most $\sigma$ sequences of $S$

$\text{supp}(p) = |\{s \in S | s \preceq p\}| < \sigma$
Let $\sigma \in \mathbb{N}^+$ be a frequency threshold and $S = \{s^i\}$ a dataset of sequences.

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

- Rare patterns
  $$RP = \{p| supp(p) < \sigma \}$$

- Non-zero-rare patterns
  $$RP = \{p| supp(p) > 0 \land supp(p) < \sigma \}$$

- Minimal rare patterns
  $$mRP = \{p| supp(p) < \sigma \land \forall p' < p, supp(p') \geq \sigma \}$$
Rare sequential patterns and minimal rare patterns

<table>
<thead>
<tr>
<th>Seq.</th>
<th>$s^{10}$</th>
<th>$s^{20}$</th>
<th>$s^{30}$</th>
<th>$s^{40}$</th>
<th>$s^{50}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle d, a, b, c \rangle$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle a, c, b, c \rangle$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle a, b, c \rangle$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle a, c, b \rangle$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\langle a, c \rangle$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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]

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

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

⇒ enumerate all sequential patterns and their supported sequences
Embedding encoding

- **Encoding strategy named “fill-gaps” (vs “skip-gaps”)**
- \( \text{occ}(s, x, u) : \langle p_i \rangle_{1 \leq i \leq x} \sqsubseteq \langle s_j \rangle_{1 \leq j < u} \)
- 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
- \( \text{rare}(l) \) 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 \text{rare}(L) :- L=1..L, \text{patlen}(L),
15 \quad \#count\{ S : \text{occ}(S, IL, LS), \text{seqlen}(S,LS) \} < \text{th}.
16 \text{rare}(L) :- \text{rare}(L-1), L<=PL, \text{patlen}(PL).
17 :- \text{not rare}(L), \text{patlen}(L).
18 :- \text{not support}(S) : \text{seq}(S,_,_).
Minimal rare patterns

- A pattern of length \( n \) is minimal rare iff any sub-pattern is rare.
- It is sufficient to evaluate rarity of sub-pattern of length \( n - 1 \).

Encoding principle

- Evaluate the rarity for each sub-pattern of length \( n - 1 \).
- \( \text{sprare}(u, l) \) sub-pattern \( \langle p_i \rangle_{1 \leq i \leq l \land i \neq u} \) is rare.
- Require to evaluate support for all sub-pattern in the same answer set.
- Encoding trick: sub-pattern embeddings are similar before position \( u \) (not recomputed).

Diagram:

\[ \text{a, b, d, c} \]

- \( u = 2 \):
- \( u = 3 \):

\[ \text{a, b, d, c} \]
Minimal rare patterns II

<table>
<thead>
<tr>
<th>Atom</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>suppat(u)</td>
<td>( u ) is an identifier of a sub-pattern ( u = \langle p_i \rangle_{1 \leq i \leq n}, i \neq u )</td>
</tr>
<tr>
<td>spocc(s,u,l,x)</td>
<td>( \langle u_i \rangle_{1 \leq i \leq x} \subseteq \langle s_j \rangle_{1 \leq j &lt; l}, ) where ( u \leq l \leq n+1 )</td>
</tr>
<tr>
<td>sprare(u,x)</td>
<td>( \langle u_i \rangle_{1 \leq i \leq x} ) is rare</td>
</tr>
</tbody>
</table>

\[
\text{suppat}(U) :- U=1..L, \text{patlen}(L), L>1. \\
\text{spocc}(S,1,2,P) :- \text{seq}(S,P,1), \text{pat}(2,1), \text{not support}(S). \\
\text{spocc}(S,U,U,P) :- \text{suppat}(U), \text{occ}(S,U-1,P), \text{not support}(S). \\
\text{spocc}(S,U,L,P+1) :- \text{spocc}(S,U,L,P), \text{seq}(S,P+1,\_). \\
\text{spocc}(S,U,L+1,P+1) :- \text{spocc}(S,U,L,P), \text{seq}(S,P+1,L), \text{pat}(L+1,L). \\
\text{sprare}(U,U-1) :- \text{suppat}(U). \\
\text{sprare}(U,L) :- \text{sprare}(U,L-1), L<=LP, \text{patlen}(LP). \\
\text{sprare}(U,IL) :- \text{suppat}(U), IL=U+1..L, \text{patlen}(L), \# \text{count}\{ S: \text{spocc}(S,U,IL,LS), \text{seqlen}(S,LS); \text{S: support}(S) \} < \text{th}. \\
\text{sprare}(U,L), \text{patlen}(L). 
\]
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)
  ⇒ expected results: MRSM has better results (better than procedural approach), exponential increasing wrt threshold and sequence lengths
- 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/aspreseqmining/
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)

  ⇒ expected results: MRSM has better results (better than procedural approach), exponential increasing wrt threshold and sequence lengths

- 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

**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 ($\text{maxlen} = 3$)

→ Too many pattern to be analyzed

**Explore additional constraints**
- Selection of almost-rare patterns ($f_{min} \leq \text{supp}(p) \leq \sigma$)
- Use of a $\text{maxgap}$ 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

\[ \sigma = 10\%, \ maxlen = 3, \ f_{min} = 5\% \text{ and } \max\text{gap} = 3 \]

→ pattern number = 133
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
  → 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)
Exponential increase of time wrt threshold

ASP-RSM is constant: most of the time is spent to enumerate patterns and the number of patterns 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

Mean sequence length ($\lambda$)
$|\mathcal{D}| = 250 - \sigma = 20\%$

- **ASP-MRSM**
- **ASP-RSM**
- **MRG-EXP**
- **Apriori-Rare**

- exponential increase of time wrt mean sequence length


