Declarative AI and Digital Forensics: Activities and Results within the DigForASP project

Francesca A. Lisi and Gioacchino Sterlicchio

Department of Computer Science
Lab of Knowledge Acquisition and Machine Learning (LACAM)
A$\mathcal{R}\forall$ Group

francesca.lisi@uniba.it

Ital-IA 2023 - Pisa, 29/05/2023

Overview

Introduction I

Digital Forensics

Focus on the phase of *Evidence Analysis*:

- Examination and aggregation of evidence, collected from various electronic devices, about crimes and criminals in order to reconstruct **events**, **event sequences** and scenarios related to a crime.
- Results are then made available to law enforcement, investigators, intelligence agencies, public prosecutors, lawyers and judges

Introduction II

Digital Forensics: Research challenges for AI

- Fragmented knowledge
- Complex scenarios (space, time, causality, uncertainty, etc.)
- Big data
- Explainability

Introduction III

The DigForASP project

https://digforasp.uca.es/

- Formal and verifiable AI methods and techniques for Evidence Analysis [Costantini et al., 2019b]
- Preference for logic-based AI methods for explainability reasons, *e.g.*, NM reasoning with ASP [Costantini et al., 2019a]
- Not only deductive reasoning!

cost
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

DIGFORASP

The case study

- The DigForASP dataset of mobile phone records
- The problem of phone call analysis

The case study I

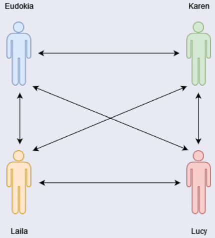
The DigForASP dataset of mobile phone records

Four Excel files with the following schema:

- *Type*: what kind of operation the user has performed (*e.g.*, incoming/outgoing call or SMS);
- *Caller*: who makes the call or sends an SMS;
- *Callee*: who receives the call or SMS;
- *Street*: where the operation has taken place;
- *Time*: when the operation has taken place (ISO format HH:MM:SS);
- *Duration*: how long the operation has been (ISO format HH:MM:SS);
- *Date*: when the operation has taken place (format: day, month, year).

The case study II

The problem of phone call analysis



1. From the Eudokia Makrembolitissa dataset, would it be possible to find her accomplices Karen Cook McNally or/and Laila Lalami?
2. From the Eudokia Makrembolitissa, Karen Cook McNally and Laila Lalami dataset, would it be possible to find Lucy Delaney?
3. Do same people gather physically often?
4. **When X calls Y, do always Y calls Z shortly afterwards?**
5. At the time of the crime, who was at the same location, or called by Eudokia Makrembolitissa?
6. The day before, who spoke with Eudokia Makrembolitissa? Or any other suspect?

Analysing Phone Calls with Declarative Pattern Mining

1. Mining Sequential Patterns
   [Lisi and Sterlicchio, 2022a, Lisi and Sterlicchio, 2022b]
2. Mining Contrast Patterns [Lisi and Sterlicchio, 2023]

Analysing Phone Calls with Declarative Pattern Mining I

What is declarative pattern mining?

- Pattern mining within a declarative framework, e.g.
  - Constraint Programming (CP) [De Raedt et al., 2010, Guns et al., 2017]
  - Boolean Satisfiability (SAT) [Jabbour et al., 2015]
  - Answer Set Programming (ASP) [Gebser et al., 2016, Guyet et al., 2018]
- DPM covers many pattern mining tasks such as sequence mining [Negrevergne and Guns, 2015, Gebser et al., 2016] and frequent itemset mining [Jabbour et al., 2015, Guns et al., 2017].

Analysing Phone Calls with Declarative Pattern Mining II

ASP in a nutshell

- Logic programming paradigm under answer set (or "stable model") semantics [Brewka et al., 2011]
- Highly declarative and expressive programming language, oriented towards difficult search problems.
- Used in a wide variety of applications in different areas like problem solving, configuration, information integration, security analysis, agent systems, semantic web, and planning.
- In ASP, search problems are reduced to computing answer sets, and an ASP solver (i.e., a program for generating stable models) is used to find solutions.

Mining sequential patterns I

Sequential Pattern Mining [Mooney and Roddick, 2013]

- Let Σ be the *alphabet*, *i.e.*, the set of items.
- An *itemset* $A = \{a_1, a_2, \ldots, a_m\} \subseteq \Sigma$ is a finite set of items.
- A *sequence* s is of the form $s = \langle s_1 s_2 \ldots s_n \rangle$ where each s_i is an itemset, and n is the length of the sequence.
- Given two sequences $s = \langle s_1 \ldots s_m \rangle$ and $t = \langle t_1 \ldots t_n \rangle$ with $m \leq n$, we say that s *is contained in* t, $s \sqsubseteq t$, if $s_i \subseteq t_{e_i}$ for $1 \leq i \leq m$ and an increasing sequence $(e_1 \ldots e_m)$ of positive integers $e_i \in [n]$, called an *embedding* of s in t.
  - E.g., we have $\langle a(cd) \rangle \sqsubseteq \langle ab(cde) \rangle$ relative to embedding $(1, 3)$.

Mining sequential patterns II

Sequential Pattern Mining (contd.)

- A *database* D is a multiset of sequences over Σ.
- The *cover* of p is the set of sequences in D that contain p: $cover(p, D) = \{t \in D | p \sqsubseteq t\}$. The number of sequences in D containing p is called its *support*, *i.e.*, $supp(p, D) = |cover(p, D)|$.
- For an integer k, the problem of *frequent sequence mining* is about discovering all sequences p such that $supp(p, D) \geq k$. We often call p a (sequential) pattern, and k is also referred to as the (minimum) *support threshold*.

Mining sequential patterns III

Example of sequential pattern mining

Id	Sequence
1	$\langle d\ a\ b\ c \rangle$
2	$\langle a\ c\ b\ c \rangle$
3	$\langle a\ b\ c \rangle$
4	$\langle a\ b\ c \rangle$
5	$\langle a\ c \rangle$
6	$\langle b \rangle$
7	$\langle c \rangle$

For $k = 2$ we can see how $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$, $\langle a\ b \rangle$, $\langle a\ c \rangle$, $\langle b\ c \rangle$ e $\langle a\ b\ c \rangle$ are common patterns in the following database $\mathcal{D}$

Mining sequential patterns IV

Pre-processing of phone records: From data to events

Each record has been transformed into a fact *seq_event(t, p, e)* where:

- *t* identifies the sequence by date,
- *p* defines the position of *e* within *t*.
- *e* represents the event, which is made up of:
  - *Type*: type of event ("in_sms", "redirect", "out_call", etc.);
  - *Caller*: the name of the caller;
  - *Callee*: the name of the callee;
  - *Street_a*, *Street_b*: the geo-location of the event;
  - the *(hour, minute, seconds)* triple: indicates the moment in time when the event occurred;
  - *Weekday*: the day of the week (0 = Monday, ..., 6 = Sunday);
  - *Duration*: duration, expressed in seconds, of the operation described by *Type*.

Mining sequential patterns V

Pre-processing of phone records: From events to sequences

- Additional pre-processing is required to create simpler and easier to analyze sequences.
- The idea is to create sequences whose identifier refers to a particular day describing what events on that day happened.
- Two types of sequences have been identified:

  Communication sequences The event *e* is the *(Caller, Callee)* pair.

  Localization sequences The event *e* is the *(Street_a, Street_b)* pair.

Mining sequential patterns VI

An example of communication sequences

```
avg_len_sequences(53).
number_of_sequences(164).
max_len_sequences((1,2,2041),129).
seq((1,9,2040),1,(eudokia_makrembolitissa,florence_violet_mckenzie)).
seq((1,9,2040),2,(eudokia_makrembolitissa,florence_violet_mckenzie)).
seq((1,9,2040),3,(florence_violet_mckenzie,eudokia_makrembolitissa)).
.
.
seq((2,9,2040),1,(annie_dillard,eudokia_makrembolitissa)).
seq((2,9,2040),2,(eudokia_makrembolitissa,irena_jordanova)).
seq((2,9,2040),3,(eudokia_makrembolitissa,irena_jordanova)).
.
.
```

Mining sequential patterns VII

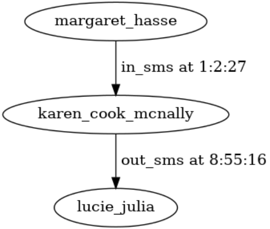
ASP encoding for sequential pattern mining

```
item(I) :- seq(_, _,(I, _, _)).

% sequential pattern generation
patpos(1).
0 { patpos(Ip+1) } 1 :- patpos(Ip), Ip<maxlen.
patlen(L) :- patpos(L), not patpos(L+1).
1 { pat(Ip,I): item(I) } 1 :- patpos(Ip).

% pattern embeddings
occ(T,1,P) :- seq(T,P,(I, _, _)), pat(1,I).
occ(T,L,P) :- occ(T, L,   P-1), seq(T,P,_).
occ(T,L,P) :- occ(T, L-1, P-1), seq(T,P,(C, _, _)), pat(L,C).

% frequency constraint
seqlen(T,L) :- seq(T,L,_), not seq(T,L+1,_).
support(T) :- occF(T, L, LS), patlen(L), seqlen(T,LS).
:- { support(T) } < th.

% pattern information
len_support(N) :- N = #count{T : supp(T)}.
pat_information(T, (Pos, C) , Type, Time) :- supp(T), pat(Pos, C), seq(T, P, (C, Type, Time)), occ(T, Pos, P).

% constraint for specific db with none line
:- pat(_, (none, _)). :- pat(_, (_, none)).

% constraint for minimum pattern lenght
:- #count{T : pat(T, _)} < minlen.
```

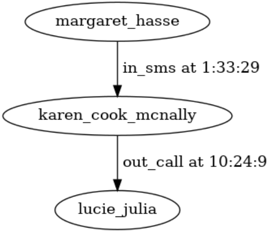
Mining sequential patterns VIII

An example of sequential pattern

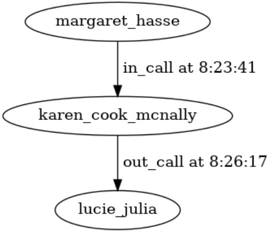
```
Answer: 1
pat(1,(margaret_hasse,karen_cook_mcnally))
pat(2,(karen_cook_mcnally,lucie_julia))
support((8,9,2040)) support((9,9,2040)) support((12,9,2040))
pat_information((8,9,2040),(1,(margaret_hasse,karen_cook_mcnally)),in_sms(simple),(1,0,55))
pat_information((8,9,2040),(1,(margaret_hasse,karen_cook_mcnally)),in_sms(simple),(1,2,27))
pat_information((8,9,2040),(2,(karen_cook_mcnally,lucie_julia)),out_sms(simple),(8,55,9))
pat_information((8,9,2040),(2,(karen_cook_mcnally,lucie_julia)),out_sms(simple),(8,55,16))
pat_information((9,9,2040),(1,(margaret_hasse,karen_cook_mcnally)),in_sms(simple),(1,33,29))
pat_information((9,9,2040),(2,(karen_cook_mcnally,lucie_julia)),out_call(simple),(10,24,9))
pat_information((12,9,2040),(1,(margaret_hasse,karen_cook_mcnally)),in_call(simple),(8,23,41))
pat_information((12,9,2040),(2,(karen_cook_mcnally,lucie_julia)),out_call(simple),(8,26,17))
len_support(3)
```



Pattern of 8-9-2040



Pattern of 9-9-2040



Pattern of 12-9-2040

Mining contrast patterns I

Contrast Pattern Mining: the intuition

- Frequent pattern mining algorithms are used to discover statistically significant regularities in a set of transactions whereas the contrast pattern mining task is about detecting statistically significant differences (*contrast*) between two or more disjoint sets of transactions [Dong and Bailey, 2012].
- Class labels are introduced to partition the dataset.
- Halfway between characterization and discrimination.

Mining contrast patterns II

Contrast Pattern Mining: problem statement

Given:

- the transaction database $\mathcal{D}$ over the set of transactions T;
- the minimum absolute support threshold $minSupp \geq 0$;
- the minimum absolute support difference threshold $minDiff \geq 0$;
- the label $\alpha \in L$.

the problem of contrast pattern mining is to find all patterns $(P, diff(P, \alpha))$ such that:

1. $|P| \leq maxLength$;
2. $supp(P, T(\alpha)) \geq minSupp$;
3. $diff(P, \alpha) \geq minDiff$.

Mining contrast patterns III

Pre-processing of phone records

Class labels "in_sms", "out_sms", "in_call", "out_call", "config", "redirect", "gprs".

Features *caller*, *callee*, *street_a*, *street_b*, *time*, *weekday* and *duration*.

- *weekday* added: (0 = Monday, ..., 6 = Sunday).
- *duration* expressed in seconds.
- *time* discretized into four time slots :
  1. "morning": from 06:00:00 to 11:59:59;
  2. "afternoon" from 12:00:00 to 17:59:59;
  3. "evening" from 18:00:00 to 23:59:59;
  4. "night" from 00:00:00 to 05:59:59.

Depending on the analyst's needs, it is possible to consider (and encode) only the transactions related to specific days, months or years so as to subsequently carry out a more granular analysis. The transactions are sorted by date and time.

Mining contrast patterns IV

Example: ASP-encoded Karen's phone recordings from the morning of 07/09/2040 to the night of 08/09/2040.

```
class(t1,in_sms).
db(t1,caller(lauretta_ngcobo)). db(t1,callee(karen_cook_mcnally)).
db(t1,street_a(bowsprit_avenue)). db(t1,street_b(none)).
db(t1,date(7,9,2040)).
db(t1,time(morning)).
db(t1,weekday(4)).
db(t1,duration(0)).
.
.
class(t93,in_call).
db(t93,caller(lady_anne_halkett)). db(t93,callee(karen_cook_mcnally)).
db(t93,street_a(bigwood_court)). db(t93,street_b(none)).
db(t93,date(7,9,2040)). db(t93,time(evening)).
db(t93,weekday(4)). db(t93,duration(56)).
.
.
class(t113,out_sms).
db(t113,caller(karen_cook_mcnally)). db(t113,callee(karen_platt)).
db(t113,street_a(bayhampton_court)). db(t113,street_b(none)).
db(t113,date(8,9,2040)). db(t113,time(night)).
db(t113,weekday(5)). db(t113,duration(0)).
.
.
```

Mining contrast patterns V

ASP encoding for contrast pattern mining

```
% link facts to objects used in the encoding
item(I) :- db(_,I).
transaction(T) :- db(T,_).

% problem encoding (frequent itemset mining)
{in_pattern(I)} :- item(I).
in_support(T) :- {conflict_at(T,I) : item(I)} 0, transaction(T), class(T, class).
out_support(T) :- {conflict_out(T,I) : item(I)} 0, transaction(T), not class(T, class).
conflict_at(T,I) :- not db(T,I), in_pattern(I), transaction(T), class(T, class).
conflict_out(T,I) :- not db(T,I), in_pattern(I), transaction(T), not class(T, class).

% definition of absolute support difference (Dong et al.)
absolute_diff(D) :- N = #count{ T : in_support(T)}, M = #count{T : out_support(T)}, D = |N-M|.

% length constraint
:- maxLength+1 {in_pattern(I)}.
:- {in_pattern(I)} 0.

% frequency constraint
:- {in_support(T)} minSup-2.

% absolute growth-rate constraint
:- absolute_diff(D), D < minDiff.

% print an answer-set as made of facts built with in_pattern/1 and absolute_diff/1 predicates
#show in_pattern/1.
#show absolute_diff/1.
```

Mining contrast patterns VI

Example: Contrast patterns for the “in_call” class

```
Answer: 1
in_pattern(callee(karen_cook_mcnally)) absolute_diff(216)
Answer: 2
in_pattern(callee(karen_cook_mcnally)) in_pattern(time(afternoon)) absolute_diff(106)
Answer: 3
in_pattern(time(afternoon)) absolute_diff(130)
Answer: 4
in_pattern(time(morning)) absolute_diff(43)
Answer: 5
in_pattern(callee(karen_cook_mcnally)) in_pattern(time(morning)) absolute_diff(72)
```

Final remarks I

Summary

- Sequential and contrast pattern mining provide a suite of powerful tools for analysing evidence in the context of DF investigations.
- The ASP encoding makes the definition of algorithmic variants pretty easier, thanks to the expressive power of constraints.
- The results are encouraging, although they highlight some weaknesses as regards the scalability.

Final remarks II

Future work

- To explore several directions of improvement of the current work as regards efficiency and scalability
  - i.e., different choices for the encoding, the solver, and the computing platform
- To define new versions of the problems
- To benefit from a tighter interaction with DF experts
  - feedback as regards the validity and the usefulness of our work from DF viewpoint
  - suggestions for new interesting directions of applied research in this field

References I

Brewka, G., Eiter, T., and Truszczynski, M. (2011).
Answer set programming at a glance.
Communications of the ACM, 54(12):92–103.

Costantini, S., De Gasperis, G., and Olivieri, R. (2019a).
Digital forensics and investigations meet artificial intelligence.
Annals of Mathematics and Artificial Intelligence, 86(1):193–229.

Costantini, S., Lisi, F. A., and Olivieri, R. (2019b).
DigForASP: A European cooperation network for logic-based AI in digital forensics.
In Casagrande, A. and Omodeo, E. G., editors, *Proceedings of the 34th Italian Conference on Computational Logic, Trieste, Italy, June 19-21, 2019*, volume 2396 of *CEUR Workshop Proceedings*, pages 138–146. CEUR-WS.org.

De Raedt, L., Guns, T., and Nijssen, S. (2010).
Constraint programming for data mining and machine learning.
In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Dong, G. and Bailey, J. (2012).
Contrast data mining: concepts, algorithms, and applications.
CRC Press.

Gebser, M., Guyet, T., Quiniou, R., Romero, J., and Schaub, T. (2016).
Knowledge-based sequence mining with asp.
In *IJCAI 2016-25th International joint conference on artificial intelligence*, page 8. AAAI.

Guns, T., Dries, A., Nijssen, S., Tack, G., and De Raedt, L. (2017).
Miningzinc: A declarative framework for constraint-based mining.
Artificial Intelligence, 244:6–29.

References II

Guyet, T., Moinard, Y., Quiniou, R., and Schaub, T. (2018).
Efficiency analysis of asp encodings for sequential pattern mining tasks.
In *Advances in Knowledge Discovery and Management*, pages 41–81. Springer.

Jabbour, S., Sais, L., and Salhi, Y. (2015).
Decomposition based sat encodings for itemset mining problems.
In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 662–674. Springer.

Lisi, F. A. and Sterlicchio, G. (2022a).
Declarative pattern mining in digital forensics: Preliminary results.
In Calegari, R., Ciatto, G., and Omicini, A., editors, *Proceedings of the 37th Italian Conference on Computational Logic, Bologna, Italy, June 29 - July 1, 2022*, volume 3204 of *CEUR Workshop Proceedings*, pages 232–246. CEUR-WS.org.

Lisi, F. A. and Sterlicchio, G. (2022b).
Mining sequences in phone recordings with answer set programming.
In Bruno, P., Calimeri, F., Cauteruccio, F., Maratea, M., Terracina, G., and Vallati, M., editors, *Joint Proceedings of the 1st International Workshop on HYbrid Models for Coupling Deductive and Inductive ReAsoning (HYDRA 2022) and the 29th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2022) co-located with the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022), Genova Nervi, Italy, September 5, 2022*, volume 3281 of *CEUR Workshop Proceedings*, pages 34–50. CEUR-WS.org.

Lisi, F. A. and Sterlicchio, G. (2023).
A declarative approach to contrast pattern mining.
In Dovier, A., Montanari, A., and Orlandini, A., editors, *AIxIA 2022 – Advances in Artificial Intelligence*, pages 17–30, Cham. Springer International Publishing.

References III

Mooney, C. and Roddick, J. F. (2013).
Sequential pattern mining - approaches and algorithms.
ACM Comput. Surv., 45(2):19:1–19:39.

Negrevergne, B. and Guns, T. (2015).
Constraint-based sequence mining using constraint programming.
In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 288–305. Springer.