Graph Pattern Generation and Selection based on Minimum Description Length

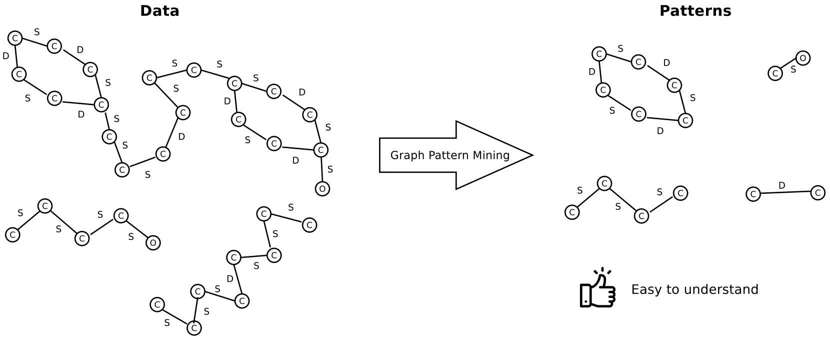
Francesco Bariatti Peggy Cellier Sébastien Ferré

Univ Rennes, INSA, CNRS, IRISA
name.surname@irisa.fr

07/07/2021

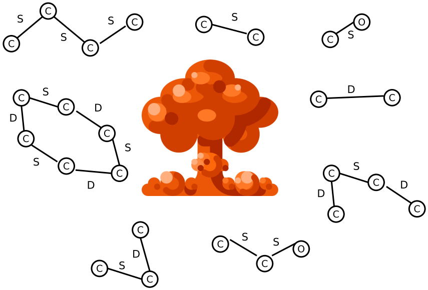
Table of Contents

Context: graph pattern mining



- A lot of data exists in graph form
  - Difficult for humans to interpret large datasets ☹
- Graph pattern mining: extract frequent structures from data
  - Easier for human to understand and interpret

Pattern explosion!



- Large number of patterns extracted even on small datasets!
- Still difficult for humans to interpret ☹
  - Difficult to analyze thousands or millions of patterns
  - Many redundant patterns

- Need to mine a *small* and *descriptive* subset of patterns

Minimum Description Length principle [Rissanen, 1978]

- From the domain of Information Theory

Informally

The model that describes the data the best is the model that compresses the data the best

Formally

Given a family of models $\mathcal{M}$ and some data D, the best model $M \in \mathcal{M}$ is the one that minimizes the *description length*

$$L(M, D) = \underbrace{L(M)}_{\text{Model}} + \underbrace{L(D|M)}_{\text{Data encoded with model}}$$

- In practice, need to define: possible models, encoding the data with a model, description length of model and encoded data
  - Some usual representations exist [Lee, 2001]

MDL for graph mining

VoG [Koutra et al., 2015]

👍 Few and descriptive patterns

☹ Non-labeled graphs

☹ Pre-defined pattern shapes only

Subdue [Cook and Holder, 1993]

👍 Patterns can have any shape

👍 Patterns can contain other patterns

☹ Loss of information: impossible to exactly reconstruct initial graph

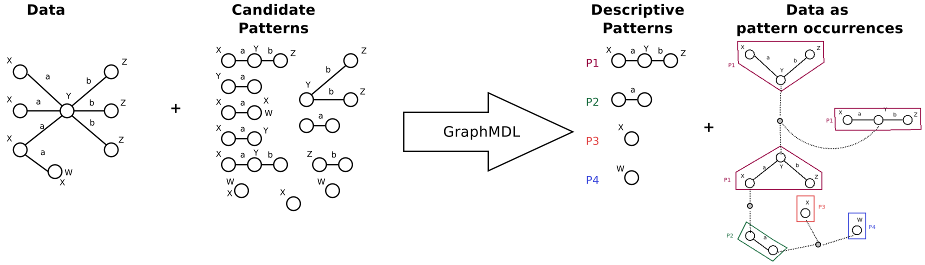
☹ Evaluate patterns individually: a "set of good patterns" and not a "good set of patterns"

Our goal

- Extract a small, **human-sized** set of **descriptive** patterns
  - Using MDL principle to guide the search
- No limits on pattern shapes
- The whole set of patterns is evaluated
  - Take into account interactions between patterns

1. Introduction
2. GraphMDL: pattern selection
   - Goal
   - Intuition
   - Experimental Evaluation
3. GraphMDL+: pattern generation and selection
   - Motivation
   - Intuition
   - Experimental evaluation
4. Current work and conclusion

GraphMDL (IDA2020) [Bariatti et al., 2020]



Input:

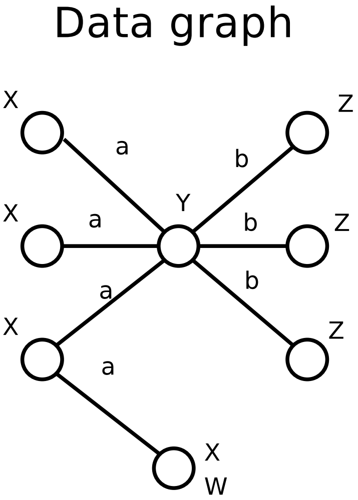
- Data graph
- Set of *candidate patterns* generated with classic graph mining algorithm

Output:

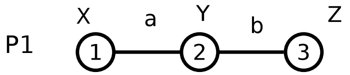
- Small set of patterns selected from the candidates
- Data encoded as pattern occurrences

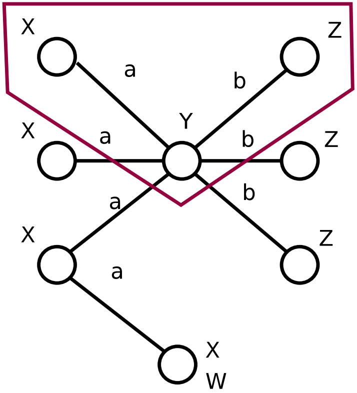
Introduced the notion of **ports** to represent data vertices at the border of several patterns

GraphMDL Intuition

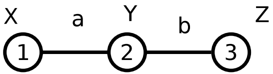


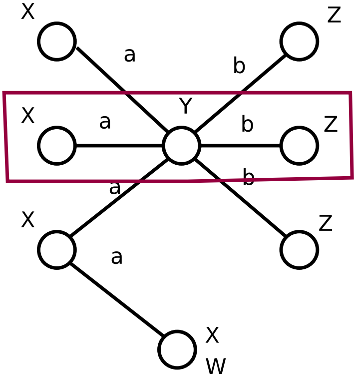
GraphMDL Intuition - patterns have occurrences in data



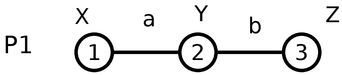


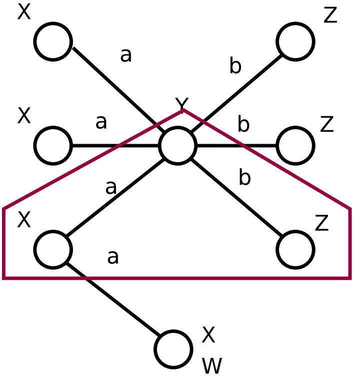
GraphMDL Intuition - patterns have occurrences in data

P1 

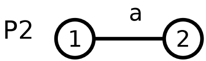


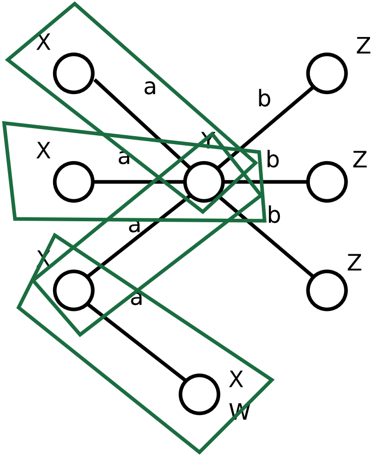
GraphMDL Intuition - patterns have occurrences in data



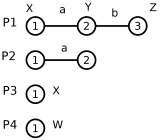


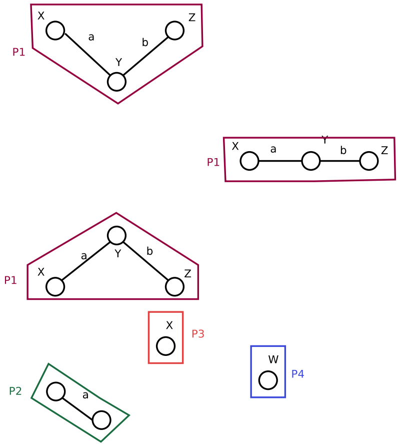
GraphMDL Intuition - patterns have occurrences in data





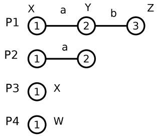
GraphMDL Intuition - data as a composition of pattern occurrences

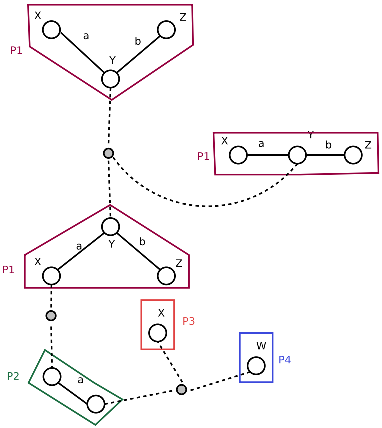


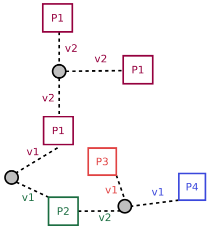


- Data graph described as a composition of pattern occurrences
- Lost connectivity

GraphMDL Intuition - notion of ports







- We call **ports** the vertices shared by multiple pattern occurrences
- Goal: finding the pattern set that gives the smallest description length
  - More details: F. Bariatti, P. Cellier, and S. Ferré, GraphMDL: Graph Pattern Selection Based on Minimum Description Length, IDA 2020

Quantitative evaluation

Dataset	gSpan support	Candidate count	GraphMDL selection	$L\% = \frac{L(M,D)}{L(M_0,D)}$
AIDS-CA	20%	2194	115	24.42%
AIDS-CA	15%	7867	123	21.64%
AIDS-CA	10%	20596	148	19.03%
AIDS-CM	20%	433	111	28.91%
AIDS-CM	15%	779	131	27.44%
AIDS-CM	10%	2054	163	24.94%
AIDS-CM	5%	9943	225	20.43%
UD-PUD-En	10%	164	162	39.55%
UD-PUD-En	5%	458	249	34.45%
UD-PUD-En	1%	6021	523	28.14%
UD-PUD-En	0%	233434	773	26.25%

- First two datasets: molecules
- Third dataset: dependency relationships between words in sentences

Quantitative evaluation

Dataset	gSpan support	Candidate count	GraphMDL selection	$L\% = \frac{L(M,D)}{L(M_0,D)}$
AIDS-CA	20%	2194	115	24.42%
AIDS-CA	15%	7867	123	21.64%
AIDS-CA	10%	20596	148	19.03%
AIDS-CM	20%	433	111	28.91%
AIDS-CM	15%	779	131	27.44%
AIDS-CM	10%	2054	163	24.94%
AIDS-CM	5%	9943	225	20.43%
UD-PUD-En	10%	164	162	39.55%
UD-PUD-En	5%	458	249	34.45%
UD-PUD-En	1%	6021	523	28.14%
UD-PUD-En	0%	233434	773	26.25%

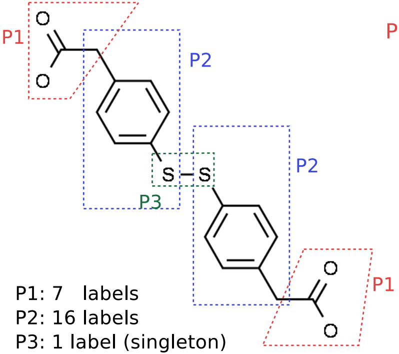
- Number of patterns greatly reduced
  - Candidates probably have redundancies, avoided by GraphMDL
- GraphMDL finds pattern that decrease description length
  - Good at finding descriptive patterns that compress the data

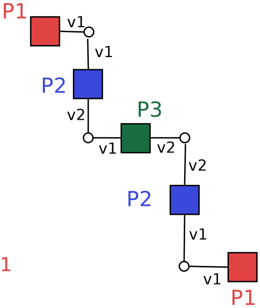
Quantitative evaluation

Dataset	gSpan support	Candidate count	GraphMDL selection	$L\% = \frac{L(M,D)}{L(M_0,D)}$
AIDS-CA	20%	2194	115	24.42%
AIDS-CA	15%	7867	123	21.64%
AIDS-CA	10%	20596	148	19.03%
AIDS-CM	20%	433	111	28.91%
AIDS-CM	15%	779	131	27.44%
AIDS-CM	10%	2054	163	24.94%
AIDS-CM	5%	9943	225	20.43%
UD-PUD-En	10%	164	162	39.55%
UD-PUD-En	5%	458	249	34.45%
UD-PUD-En	1%	6021	523	28.14%
UD-PUD-En	0%	233434	773	26.25%

- Number of patterns greatly reduced
  - Candidates probably have redundancies, avoided by GraphMDL
- GraphMDL finds pattern that decrease description length
  - Good at finding descriptive patterns that compress the data

Qualitative evaluation





- Both small and big patterns selected
- Known structure found: P1 is carboxylic acid
  - Without any previous chemistry knowledge!
- Ports clearly identify how patterns are connected

GraphMDL+ (SAC2021) [Bariatti et al., 2021]

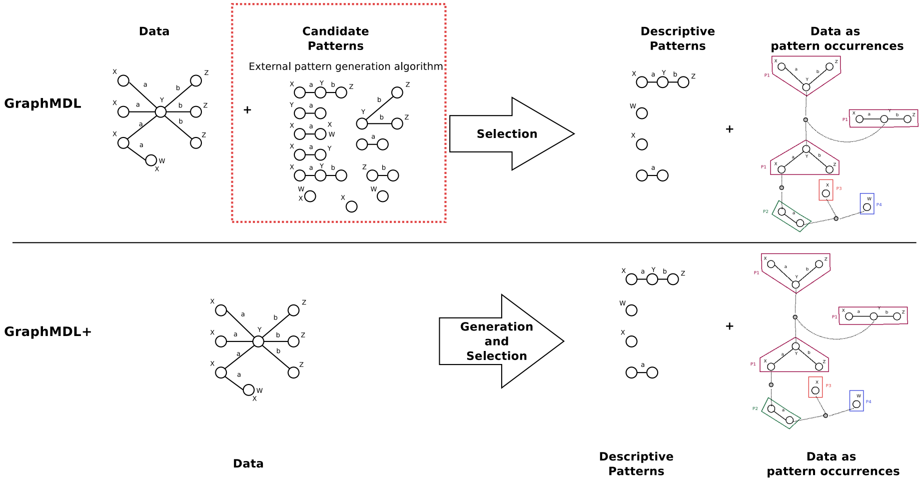
GraphMDL: select patterns from a set of candidates:

- ☹ Need to mine candidates beforehand
- ☹ Dependency on external pattern generation algorithm
  - Candidates restricted to what that algorithm can generate
- ☹ Need to sieve through many "useless" candidates
  - Pattern generation algorithm has no knowledge of what could be useful to GraphMDL
  - Large set of candidate patterns: large time needed to treat them

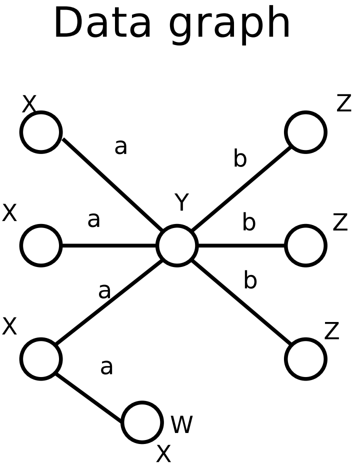
GraphMDL+: generate and select patterns

- 👍 Control over candidate patterns
- 👍 Independent from other approaches
- 👍 Parameter-free approach
- 👍 Anytime approach
  - Generation & selection can be stopped at any time and still give a solution

GraphMDL vs GraphMDL+

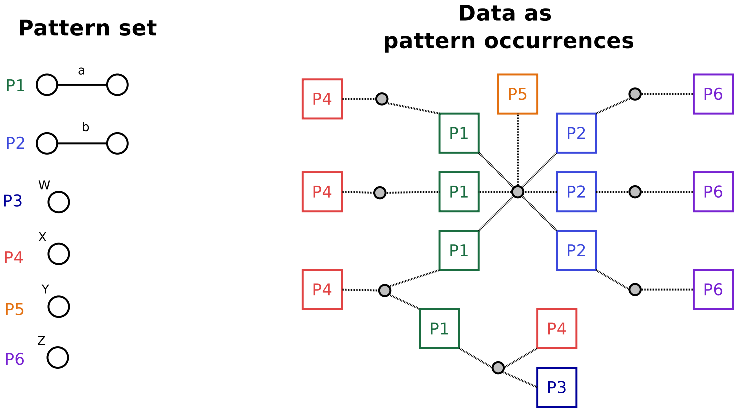


GraphMDL+ candidate generation overview



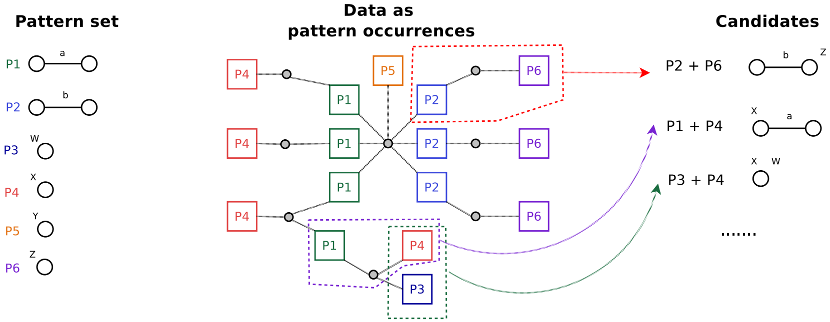
GraphMDL+ candidate generation overview

1. Encode the data with the current pattern set
   - First iteration: singleton-only pattern set



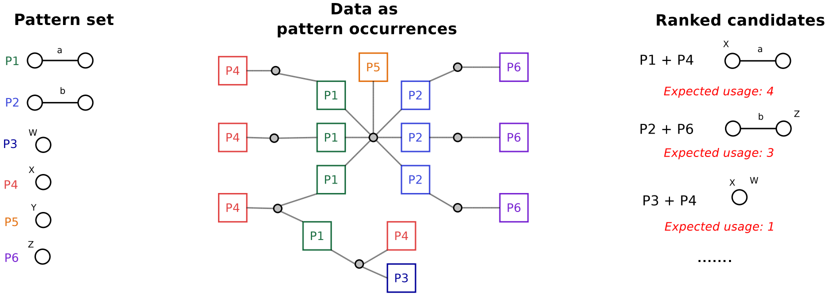
GraphMDL+ candidate generation overview

2. Create a candidate for each pair of patterns sharing a port
   - If they appear together, maybe they can be replaced with a pattern that merges them



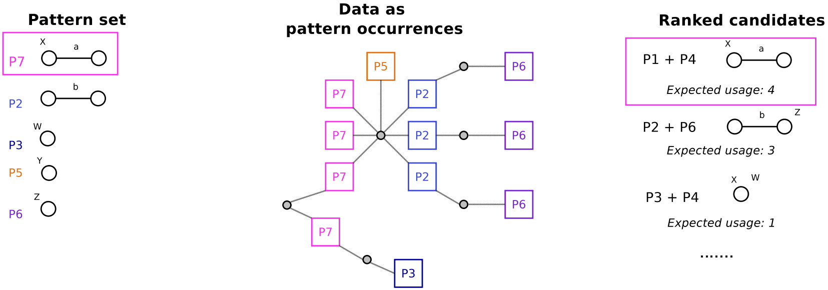
GraphMDL+ candidate generation overview

3. Rank all candidates according to heuristic



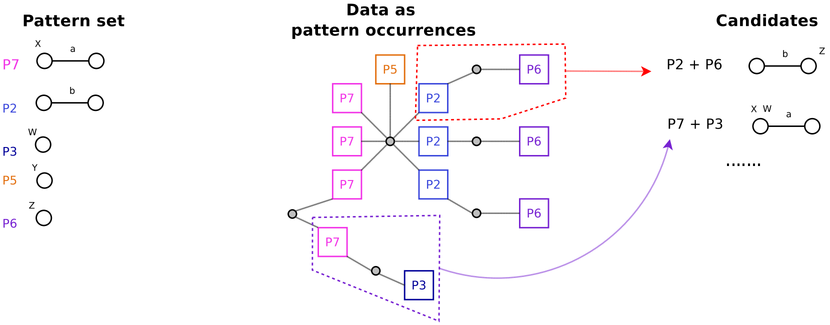
GraphMDL+ candidate generation overview

4. Try to add top candidate to pattern set, see if it improves according to MDL
   - <u>If it does</u>: candidate accepted
   - If it does not: test next candidate



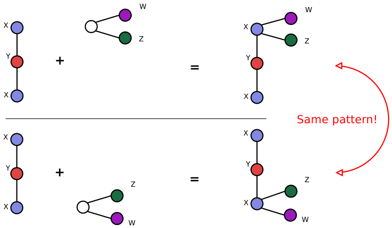
GraphMDL+ candidate generation overview

5. Repeat until no candidate improves pattern set



Main challenges

- Pattern merging formal definition
  - Allow any way of merging two patterns
  - More than one vertex per pattern may be merged
- Isomorphisms and automorphisms [Fortin, 1996]



Patterns that appear different may actually be equivalent

- Candidate ranking heuristic
  - Which candidate tested first? Avoid testing all candidates at each step
  - Evaluated experimentally

More details: F. Bariatti, P. Cellier, and S. Ferré, GraphMDL+: interleaving the generation and MDL-based selection of graph patterns, SAC 2021.

Quantitative comparison with GraphMDL

Dataset	GraphMDL		GraphMDL+		
	$L_1 = best\ L\%$	time	time for $L\% \leq L_1$	best $L\%$	time
AIDS-CA	19.15%	111m	36s	12.02%	77m
AIDS-CM	20.71%	170m	1m	14.68%	240m
Mutag	15.42%	158m	3s	10.73%	1m
PTC-FM	22.87%	102m	12s	22.01%	1m
PTC-FR	23.35%	27m	8s	22.62%	1m
PTC-MM	23.65%	129m	4s	22.12%	1m
PTC-MR	23.09%	18m	7s	21.43%	1m
UD-PUD-En (undir.)	26.84%	101m	3m	25.29%	152m

Quantitative comparison with GraphMDL

	GraphMDL		GraphMDL+		
Dataset	$L_1 = best\ L\%$	time	time for $L\% \leq L_1$	best $L\%$	time
AIDS-CA	19.15%	111m	36s	12.02%	77m
AIDS-CM	20.71%	170m	1m	14.68%	240m
Mutag	15.42%	158m	3s	10.73%	1m
PTC-FM	22.87%	102m	12s	22.01%	1m
PTC-FR	23.35%	27m	8s	22.62%	1m
PTC-MM	23.65%	129m	4s	22.12%	1m
PTC-MR	23.09%	18m	7s	21.43%	1m
UD-PUD-En (undir.)	26.84%	101m	3m	25.29%	152m

- 👍 Significantly less time than GraphMDL to attain equivalent results
  - GraphMDL needs to process *all* candidate patterns, but most of them are redundant and/or useless in terms of MDL
  - GraphMDL+ generates less candidates of higher quality

Quantitative comparison with GraphMDL

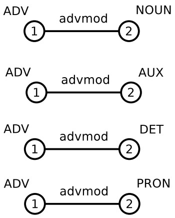
Dataset	GraphMDL $L_1 = best\ L\%$	GraphMDL time	GraphMDL+ time for $L\% \leq L_1$	GraphMDL+ best $L\%$	GraphMDL+ time
AIDS-CA	19.15%	111m	36s	12.02%	77m
AIDS-CM	20.71%	170m	1m	14.68%	240m
Mutag	15.42%	158m	3s	10.73%	1m
PTC-FM	22.87%	102m	12s	22.01%	1m
PTC-FR	23.35%	27m	8s	22.62%	1m
PTC-MM	23.65%	129m	4s	22.12%	1m
PTC-MR	23.09%	18m	7s	21.43%	1m
UD-PUD-En (undir.)	26.84%	101m	3m	25.29%	152m

- GraphMDL+ finds better patterns sets than GraphMDL
  - Given enough time and candidates GraphMDL would also find them, but in such a long time that it would not be practical

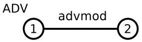
Qualitative comparison with GraphMDL

GraphMDL+ does not depend on external graph mining approaches

GraphMDL



GraphMDL+



Generalizes better

- GraphMDL could handle the pattern on the right, but the algorithm used for candidate generation[1] could not (forced all vertices to have a label)

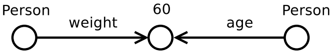
[1] gSpan [Yan and Han, 2002]

Current work: KG-MDL

Current work: applying GraphMDL+ to knowledge graphs

- Huge quantity of data available as knowledge graphs
- KGs can be human-readable but large: patterns can help extract knowledge

- ☹ KGs are "graphs" but adaptation is not straightforward



  - Should two entities be neighbours just because they have the same literal value for a property?
  - Should all lists be connected just because they all end with `rdf:nil` ?
- ☹ Scalability issues
  - Nodes degree tend to follow a power-law
  - Pathological patterns, e.g. "two things in the same city"

- Current results are very promising

Conclusion

- MDL principle is a powerful tool for graph mining
- We developed approaches to select small sets of descriptive patterns
  - Anytime and parameterless
- Notion of **ports** to encode the data using pattern occurrences
  - From the data point of view: vertices described by multiple patterns
  - From pattern point of view: “interface” to other patterns
- Extracted patterns allow for human interpretation of complex datasets

I am looking for a post-doctorate out of France starting in summer 2022!

francesco.bariatti@irisa.fr

Bariatti, F., Cellier, P., and Ferré, S. (2020).
GraphMDL: Graph Pattern Selection Based on Minimum Description Length.
In Berthold, M. R., Feelders, A., and Krempl, G., editors, *Advances in Intelligent Data Analysis XVIII*, Lecture Notes in Computer Science, pages 54–66. Springer International Publishing.

Bariatti, F., Cellier, P., and Ferré, S. (2021).
GraphMDL+: interleaving the generation and MDL-based selection of graph patterns.
In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, pages 355–363. Association for Computing Machinery.

Cook, D. J. and Holder, L. B. (1993).
Substructure Discovery Using Minimum Description Length and Background Knowledge.
Journal of Artificial Intelligence Research, 1:231–255.

Fortin, S. (1996).
The Graph Isomorphism Problem.
page 25.

Koutra, D., Kang, U., Vreeken, J., and Faloutsos, C. (2015).

Summarizing and understanding large graphs.
Statistical Analysis and Data Mining: The ASA Data Science Journal, 8(3):183–202.

Lee, T. C. M. (2001).
An Introduction to Coding Theory and the Two-Part Minimum Description Length Principle.
International Statistical Review, 69(2):169–183.

Rissanen, J. (1978).
Modeling by shortest data description.
Automatica, 14(5):465–471.

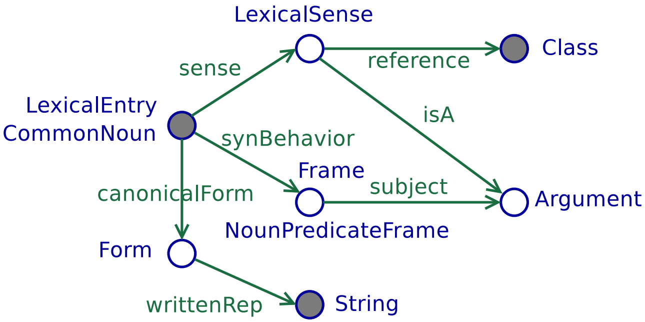
Yan, X. and Han, J. (2002).
gSpan: Graph-based substructure pattern mining.
In *2002 IEEE International Conference on Data Mining. Proceedings*, pages 721–724. Springer Berlin Heidelberg.

Image credits

-  made by Freepik, from `www.flaticon.com`
-  made by Dave Gandy, from `www.flaticon.com`
-  made by Adib Sulthon, from `www.flaticon.com`

- All other images: Francesco Bariatti

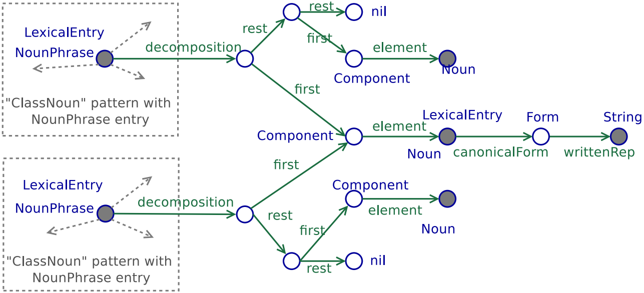
KG-MDL patterns - Lemon dataset



"ClassNoun" pattern with CommonNoun entry

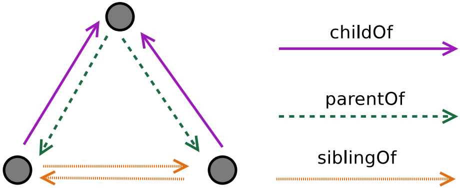
- Lemon dataset creates its data using some *design patterns*
- KG-MDL retrieved the design patterns by looking at the data
  - Fig. above: "ClassNoun" pattern, i.e. a lexical entry that is a common noun (e.g. "lake"), and whose meaning is a class (e.g. `dbo:Lake`)

KG-MDL patterns - Lemon dataset



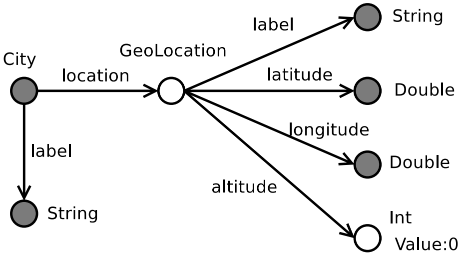
- A pair of NounPhrase (compound nouns) sharing the same first element whose meaning are distinct classes, e.g. `dbo:BloodVessel` and `dbo:BloodType`.
- Does not correspond to a design pattern used to create the graph, but highlights a common structure of the data

KG-MDL patterns - NTNames dataset



- Suggests axioms that appear in the RDFS/OWL schema
  - `childOf` inverse of `parentOf`
  - `siblingOf` symmetric
- Suggest axions that can not be expressed in RDFS/OWL
  - Two people with same parent are sibling

KG-MDL patterns - NTNames dataset



- How to represent geographical information of cities using the schema
- Altitude is always 0!? It is actually what happens in the data
  - Probably an error or a way to represent missing information
  - Impossible to know by just looking at the RDFS/OWL schema