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

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
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
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 humans 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
Introduction

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

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

- 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

<table>
<thead>
<tr>
<th>VoG [Koutra et al., 2015]</th>
<th>Subdue [Cook and Holder, 1993]</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎉 Few and descriptive patterns</td>
<td>🎉 Patterns can have any shape</td>
</tr>
<tr>
<td>🙁 Non-labeled graphs</td>
<td>🎉 Patterns can contain other patterns</td>
</tr>
<tr>
<td>🙁 Pre-defined pattern shapes only</td>
<td>🙁 Loss of information: impossible to exactly reconstruct initial graph</td>
</tr>
<tr>
<td></td>
<td>🙁 Evaluate patterns individually: a “set of good patterns” and not a “good set of patterns”</td>
</tr>
</tbody>
</table>
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

Data graph

X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
X \rightarrow Y \rightarrow Z
GraphMDL Intuition - patterns have occurrences in data
GraphMDL Intuition - patterns have occurrences in data
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
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

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

- **First two datasets:** molecules
- **Third dataset:** dependency relationships between words in sentences
## Quantitative evaluation

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

- 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

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

- 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
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+ (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**

Data

GraphMDL

Candidate Patterns

External pattern generation algorithm

GraphMDL+

Data

Generation and Selection

Descriptive Patterns

Selection

Data as pattern occurrences

Descriptive Patterns

Data as pattern occurrences
GraphMDL+: pattern generation and selection

Intuition

GraphMDL+ candidate generation overview

Data graph

X

a

b

Y

Z

Z

X

a

b

Z

Z

Z

W

a

X
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

1. Rank all candidates according to heuristic

Pattern set

- P1
- P2
- P3
- P4
- P5
- P6

Data as pattern occurrences

Ranked candidates

- P1 + P4  
  Expected usage: 4

- P2 + P6  
  Expected usage: 3

- P3 + P4  
  Expected usage: 1

...
GraphMDL+ candidate generation overview

Try to add top candidate to pattern set, see if it improves according to MDL
- **If it does**: candidate accepted
- **If it does not**: test next candidate
GraphMDL+ candidate generation overview

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

### Quantitative comparison with GraphMDL

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GraphMDL</th>
<th>GraphMDL+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1 = best \ L%$</td>
<td>time for $L% \leq L_1$</td>
</tr>
<tr>
<td>AIDS-CA</td>
<td>19.15%</td>
<td>111m</td>
</tr>
<tr>
<td>AIDS-CM</td>
<td>20.71%</td>
<td>170m</td>
</tr>
<tr>
<td>Mutag</td>
<td>15.42%</td>
<td>158m</td>
</tr>
<tr>
<td>PTC-FM</td>
<td>22.87%</td>
<td>102m</td>
</tr>
<tr>
<td>PTC-FR</td>
<td>23.35%</td>
<td>27m</td>
</tr>
<tr>
<td>PTC-MM</td>
<td>23.65%</td>
<td>129m</td>
</tr>
<tr>
<td>PTC-MR</td>
<td>23.09%</td>
<td>18m</td>
</tr>
<tr>
<td>UD-PUD-En (undir.)</td>
<td>26.84%</td>
<td>101m</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GraphMDL</th>
<th>GraphMDL+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_1 = \text{best } L%$</td>
<td>time</td>
</tr>
<tr>
<td>AIDS-CA</td>
<td>19.15%</td>
<td>111m</td>
</tr>
<tr>
<td>AIDS-CM</td>
<td>20.71%</td>
<td>170m</td>
</tr>
<tr>
<td>Mutag</td>
<td>15.42%</td>
<td>158m</td>
</tr>
<tr>
<td>PTC-FM</td>
<td>22.87%</td>
<td>102m</td>
</tr>
<tr>
<td>PTC-FR</td>
<td>23.35%</td>
<td>27m</td>
</tr>
<tr>
<td>PTC-MM</td>
<td>23.65%</td>
<td>129m</td>
</tr>
<tr>
<td>PTC-MR</td>
<td>23.09%</td>
<td>18m</td>
</tr>
<tr>
<td>UD-PUD-En (undir.)</td>
<td>26.84%</td>
<td>101m</td>
</tr>
</tbody>
</table>

- 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 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]
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
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


Summarizing and understanding large graphs.  

An Introduction to Coding Theory and the Two-Part Minimum Description Length Principle.  

Modeling by shortest data description.  

Yan, X. and Han, J. (2002).  
gSpan: Graph-based substructure pattern mining.  
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
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).
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.
Suggests axioms that appear in the RDFS/OWL schema
- `childOf` inverse of `parentOf`
- `siblingOf` symmetric

Suggest axioms that can not be expressed in RDFS/OWL
- Two people with same parent are sibling
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