Using Formal Concept Analysis to Infer Schemas for Semi-Structured Data

Bachelor thesis
Luca Liechti
Software Composition Group
Universität Bern
18.10.2016
Roadmap

- Structured vs. semi-structured data
- Goals of inference
- How Formal Concept Analysis helps
- Tools and resources used
- Example data sets
- Literature
Structured vs. semi-structured data

> “Semi-structured data is a form of structured data that does not conform with the formal structure of data models associated with relational databases or other forms of data tables, but nonetheless contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data.” (https://en.wikipedia.org/wiki/Semi-structured_data)

> e.g. XML, JSON, BibTeX

> Advantages and disadvantages compared to structured (relational) data

> Growing importance of NoSQL databases
Structured vs. semi-structured data

lib
id  title                           journal  year  vol  date  inst
1   The C Programming Language     NULL     1978  NULL  NULL  NULL
2   Inferring NoSQL schema         VLDB     2016  1    NULL  NULL
3   Hacking Evil Corp              NULL     NULL  NULL  09.05.2015  fsociety

auth
id  name
1   Brian W. Kernighan
2   Dennis M. Ritchie
3   John Doe
4   Elliot Anderson

ref
lib_id  auth_id
1       1
1       2
2       3
3       4
What we would like

<table>
<thead>
<tr>
<th>book</th>
<th>title</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The C Programming Language</td>
<td>1978</td>
</tr>
<tr>
<td></td>
<td>Harry Potter</td>
<td>1997</td>
</tr>
<tr>
<td></td>
<td>Random book</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>article</th>
<th>title</th>
<th>author</th>
<th>journal</th>
<th>year</th>
<th>vol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inferring NoSQL schema</td>
<td>John Doe</td>
<td>Inferring NoSQL schema</td>
<td>2016</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Are You Living In a Computer Simulation?</td>
<td>Nick Bostrom</td>
<td>Philosophical Quarterly</td>
<td>2003</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Random article</td>
<td>Random woman</td>
<td>Random journal</td>
<td>2010</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>thesis</th>
<th>title</th>
<th>author</th>
<th>date</th>
<th>institution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hacking Evil Corp</td>
<td>Elliot Alderson</td>
<td>09.05.2015</td>
<td>fsociety</td>
</tr>
<tr>
<td></td>
<td>Random thesis</td>
<td>Random student</td>
<td>01.01.2010</td>
<td>Oxford University</td>
</tr>
<tr>
<td></td>
<td>Other random thesis</td>
<td>Random man</td>
<td>02.02.2012</td>
<td>Bern University</td>
</tr>
</tbody>
</table>

... also a library table that manages the others, a book-author reference table, etc.

18.10.2016

nulls!
How can we get it?

Problem: We do not know what an item is (book, article, thesis, or something else)!

We could «look inside» the data and search for patterns

But what can we learn about a semi-structured dataset without doing that?

How can we cluster (and thus minimize the number of NULLs) semi-structured data based only on attributes/keys/tags and not on content?
An alternative visualisation of our library

<table>
<thead>
<tr>
<th>Title</th>
<th>_id</th>
<th>auth</th>
<th>titl</th>
<th>year</th>
<th>jrn1</th>
<th>volm</th>
<th>date</th>
<th>inst</th>
</tr>
</thead>
<tbody>
<tr>
<td>The C Programming Language</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inferring NoSQL schema</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hacking Evil Corp</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Harry Potter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random book</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are You Living In a Computer Simulation?</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random article</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Random thesis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other random thesis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formal Concept Analysis

Context := (G, M, I) where G = objects, M = attributes, I = binary relation between G, M.
Concept := (A, B), A ⊆ G, B ⊆ M, all As have all attributes in B; these are found in all As.

Ganter, Wille: Formal Concept Analysis, p. 18f.
Using Formal Concept Analysis to Infer Schemas for Semi-Structured Data

The concept lattice

The three circled nodes each have support 3. All other nodes have support 0.
What can we learn from concepts?

- Quick structural overview over datasets
- Number and support of concepts
- Neighbouring concepts = potentially identical
- Outlier detection (e.g. Lindig 2015)

Problems:
- Which concepts do we merge in a data transformation?
- Which concepts/attributes do we eliminate in a data cleansing effort?
- Calculating the number of concepts is very expensive. Many algorithms run in \(O(|G|^2|M||L|)\) \((O = \#\text{objects}, M = \#\text{attributes}, L = \text{size of lattice})\)
Tools and resources used

- Wrote parsers for XML and BibTeX, will add JSON. Output formats: .slf and .ctxt
- Visualised obtained lattices with the following open source software:
  - ConExp (http://conexp.sourceforge.net/) (rather clearly the most powerful tool)
  - LatticeMiner (https://sourceforge.net/projects/lattice-miner/)
  - Galicia (http://www.iro.umontreal.ca/~galicia/)
- Analysed semi-structured datasets from (among others):
  - zbMATH (https://zbmath.org/) (BibTeX export of search results)
  - dblp (http://dblp.uni-trier.de/) (XML export of search results)
  - http://shelah.logic.at/eindex.html (Shela’s bibliography; BibTeX)
  - The SCG bibliography (BibTeX)
Example data sets

- Already very well-structured dataset
- Concepts need little or no transformation
- How do we algorithmically decide whether a dataset is already well-structured?

- 607 objects, of which
  - 13 Books and Theses
  - 393 Conference and Workshop Papers
  - 15 Editorship
  - 19 Informal Publications
  - 155 Journal Articles
  - 10 Parts in Books or Collections
  - 2 Reference Works

Search results from dblp; ConExp visualisation
Example data sets II

- article objects belong to many nodes
- Yet all nodes who have an author are articles!
- (Except article76 😞)
- How do we algorithmically find this node?
- 200 objects, of which
  - 177 article
  - 1 inbook
  - 22 incollection

shela’s bibliography; ConExp visualisation
Example data sets III

> Basically a hot mess
> Observe that many attributes enter the lattice only in the very last row
> Starting point for data cleansing?
> *Is there anything we can do to cleanse the dataset?*

> 200 objects, of which
> 41 article
> 26 book
> 8 incollection
> 82 inproceedings
> 4 manual
> 5 mastersthesis
> 9 misc
> 6 phdthesis
> 17 techreport
> 2 unpublished

CG bibliography (part); ConExp visualisation
I want your semi-structured datasets!
Literature