Lattice Miner 2.0 - A FCA Tool

Rokia Missaoui and Kevin Emamirad

Laboratoire de Recherche sur l'Information Multimdia (LARIM), Gatineau QC, Canada

Abstract

- *Lattice Miner* 2.0 is the latest release of a Formal Concept Analysis (FCA) software tool for the construction, visualization and manipulation of concept lattices. It allows the generation of formal concepts and association rules as well as the transformation of formal contexts and the construction and manipulation of concept lattices via exploration, approximation, projection and selection.
- The newly added procedures include the computation of implications with negation as well as the production of triadic association rules, including implications.

Implications with negation

To compute implications with negation from a given context $K = (G, M, I)$, the following theorem in [1] exploits the set $\mathbf{\Sigma}_{\mathcal{K}}$ of key-based implications of the context $K|\tilde{K}$ to infer implications with a non-null support.

Theorem

Let $Ax \subseteq M\tilde{M}$. *Then,* $Ax \rightarrow M\tilde{M} \backslash Ax\ [0] \Leftrightarrow A \rightarrow \tilde{x}\ [sup]$ *where* $sup = |A'|/|G|$ *and* G *is a set of objects.*

This is done in *Lattice Miner* by

1. first computing the context $K|\tilde{K}$,
2. identifying the left-hand side of key-based implications, which are the minimal generators of the infimum, and
3. finally generating the whole set of implications (with or without negation) by creating for each key Ax, $p = |Ax|$ implications where each element in Ax is negated and shifted from the left to the right of the generated implication.

Generation of the keys for the lattice of context $K|\tilde{K}$

```
Input: Context K = (G, M, I)
Output: 𝒦 : Generators of the infimum of the lattice of context K|K̃
INT ← ∅   {Set of co-atoms intentions}
O ← ∅   {Processed objects}
while G ≠ O do
  for all o ∈ G and o ∉ O do
    X ← o''   {Closure of the object o}
    Y ← Intent(o) ∪ (M \ Intent(o))~
    O := O ∪ X
    INT := INT ∪ {Y}
  end for
end while
𝒦 ← JEN(M ∪ M̃, INT)
return 𝒦
```

Triadic concept analysis

Triadic concept analysis was originally introduced by Lehmann and Wille [2] as an extension to FCA, to analyze data described by three sets K_1 (objects), K_2 (attributes) and K_3 (conditions) together with a 3-ary relation $Y \subseteq K_1 \times K_2 \times K_3$. $K = (K_1, K_2, K_3, Y)$ is called a *triadic context*. A triple (a_1, a_2, a_3) in Y means that object a_1 possesses attribute a_2 under condition a_3.

K	P	N	R	K	S
1	abd	abd	ac	ab	a
2	ad	bcd	abd	ad	d
3	abd	d	ab	ab	a
4	abd	bd	ab	ab	d
5	ad	ad	abd	abc	a

Types of Triadic Implications

According to Biedermann [3], a *triadic implication* has the form $(A \rightarrow D)_C$ and holds if "*whenever* A *occurs under all conditions in* C, *then* D *also occurs under the same conditions*". Later on, Ganter and Obiedkov [4] extended Biedermann's definition by proposing three types of implications:

1. *attribute×condition* implications (AxCIs),
2. *conditional attribute* implications (CAIs) (e.g. $\{K\} \xrightarrow{ad} \{P, R\}$),
3. *attributional condition* implications (ACIs).

All these variants of triadic implications and association rules are produced by our tool using procedures in [5, 6].

New procedure for CAI/ACI triadic implication generation

From the CAI implication defintion we know that $A \xrightarrow{C} D$ iff $(A \rightarrow D)_k$ for each $k \in C$:

1. Slice the triadic context into dyadic contexts based on the conditions in K_3
2. Compute Biedermann implications for each of these conditions
3. Put implications with the same premise in seperate buckets, then decompose each implication with p attributes in the right side of the implication with as many implications with just one attribute in the consequence.
4. Cumulate conditions for each implication that have the same premise and same condition part.
5. Group conclusions of implications that has the same premise
6. Compute the support of the obtained implications

Conclusion

We plan to have a Web-based release of *Lattice Miner* 2.0 using modern Web technologies. This will bring the following advantages:

1. the application can be used from a laptop or even a tablet without installing any further software
2. it can be deployed to a private or public cloud to benefit from the provider offerings such as more computational power, auto-scaling or managed containers and databases, and
3. the execution of some of the algorithms can be parallelized on different nodes.

Finally, by using modern tools and technologies we can speed up the development and improvement of this software project

Acknowledgements

- *Lattice Miner* has benefited from the contribution of many students and visitors, including Léonard Kwuida (Bern University) and Ludovic Thomas (INSA Rennes). We would like to thank all of them.

References

[1] R. Missaoui, L. Nourine, and Y. Renaud, "Computing Implications with Negation from a Formal Context," *Fundam. Inf.*, vol. 115, pp. 357–375, December 2012.

[2] F. Lehmann and R. Wille, "A Triadic Approach to Formal Concept Analysis," in *Proceedings of the Third International Conference on Conceptual Structures: Applications, Implementation and Theory*, pp. 32–43, 1995.

[3] K. Biedermann, "How Triadic Diagrams Represent Conceptual Structures," in *ICCS*, pp. 304–317, 1997.

[4] B. Ganter and S. A. Obiedkov, "Implications in Triadic Formal Contexts," in *ICCS*, pp. 186–195, 2004.

[5] R. Missaoui and L. Kwuida, "Mining Triadic Association Rules from Ternary Relations," in *9th International Conference ICFCA*, pp. 204–218, May 2011.

[6] E. Rodriguez-Lorenzo, P. Cordero, M. Enciso, R. Missaoui, and A. Mora, "An axiomatic System for Conditional Attribute Implications in Triadic concept analysis," *International Journal of Intelligent Systems*, February 2017.

{rokia.missaoui, emak01}@uqo.ca