Mining Fuzzy Concept Lattices

CMMSE 2022

M. Ojeda-Hernández, **D. López-Rodríguez**, Á. Mora, P. Cordero

Departamento de Matemática Aplicada

UNIVERSIDAD DE MÁLAGA

Table of Contents

Introduction and Motivation

- Within the classical FCA framework, the knowledge extracted from a binary table of data (formal context) is essentially represented as two complementary entities: the **concept lattice** and a basis of context-valid **implications**.
- The concept lattice represents an exhaustive analysis of the **closed sets** of objects and attributes, establishing a hierarchical biclustering among them.
- The **number of concepts can be exponential** in the size of the input context and the problem of determining this number is #P-complete.
- This is true in the binary case, and the problem in **the fuzzy case** is even worse.

	a	b	c	d	e
o1	0	0	½	0	1
o2	1	½	½	0	½
o3	½	½	0	½	½

Table 1: Fuzzy formal context

	(a, 0)	(a, ½)	(a, 1)	(b, 0)	(b, ½)	(b, 1)	(c, 0)	(c, ½)	(c, 1)	(d, 0)	(d, ½)	(d, 1)	(e, 0)	(e, ½)	(e, 1)
o1	1	0	0	1	0	0	1	1	½	1	0	0	1	1	1
o2	1	1	1	1	1	½	1	1	½	1	0	0	1	1	½
o3	1	1	½	1	1	½	1	0	0	1	1	½	1	1	½

Table 2: Original formal context scaled as in [Belohlavek and Konecny, 2017].

One might expect that with **native fuzzy** approaches the resulting algorithms could be more efficient.

Algorithms to Compute the Concept Lattice

- Determining maximal rectangles [Norris, 1978].
- First proper approach (in FCA): NextClosure algorithm [Ganter, 1984].
  - Introduces the *lectic* order in 2^M.
  - Polynomial delay $\mathcal{O}(|G||M|^2)$ searching all the intents. (note that G is the set of objects and M the set of attributes)
- Lindig's NextNeighbour builds the concept lattice from top to bottom, exploring the lower neighbours of the previously computed concepts.
- Close-by-one (CbO) [Kuznetsov, 1993]: builds a tree recursively adding new attributes to the already computed closures (by intersecting extents).
- Krajča et al. [2010]: "the major issue of widely-used algorithms for computing formal concepts is that some concepts are computed multiple times which brings significant overhead".
- Fast Close-by-One (FCbO) [Krajča et al., 2010] includes an additional canonicity test to avoid coputing concepts multiple times.
- **InClose** [Andrews, 2017] uses the canonicity test of the CbO family and incremental closure computation to reduce the number of operations.
- ...

The Fuzzy Setting

Let $\mathbb{L} = (L, 0, 1, \wedge, \vee, \otimes, \rightarrow)$ be a finite complete residuated lattice and $\mathbb{K} = (G, M, I)$ a formal context:

- G is the set of objects,
- M the set of attributes and
- $I(x, y) \in L$ is the degree to which object x possesses attribute y.

The concept-forming operators in this fuzzy case are ${}^{\uparrow} : L^G \rightarrow L^M$, ${}^{\downarrow} : L^M \rightarrow L^G$, defined as:

$$A^{\uparrow}(m) := \bigwedge_{g \in G} (A(g) \rightarrow I(g, m))$$

$$B^{\downarrow}(g) := \bigwedge_{m \in M} (B(m) \rightarrow I(g, m))$$

Extensions

Previously Known

- Adaptation of the NextClosure algorithm to the fuzzy case
- Using scaling to wrap algorithms for the *crisp* case.

Our Proposal

Refactoring the InClose family of algorithms for fuzzy sets.

The InClose Algorithms

Basic InClose algorithm:

- Suppose a concept candidate (A, B) is given.
- Find its children concepts as follows: for each $m \in M$:
  - Build $C := A \cap \{m\}^{\downarrow}$. This is an extent of the formal context.
  - If $C = A$, we update $B := B \cup \{m\}$ (the intent $A^{\uparrow}$ is computed incrementally in B).
  - Otherwise, the algorithms check that the extent C is canonical (i.e. it has not appeared before in the computations). If it is canonical, then repeat this procedure with the concept candidate (C, D) where $D := B \cup \{m\}$.

This recursive algorithm is proven to compute all the concepts if we start with the concept candidate $(G, \varnothing)$.
Several improvements are incorporated to avoid repetition of canonicity tests (InClose2), to inherit empty intersections (InClose4) and to inherit failed canonicity tests (InClose5).

Adaptation to the Fuzzy Case

- Instead of an attribute $\{m\}$, we now have to study all combinations $\{{}^{l_1}/m\}, \ldots, \{{}^{l_n}/m\}, \{m\}$, where we consider $L = \{0 < l_1 < \ldots < l_n < 1\}$.
- The fuzzy algorithm does not only need to run over all attributes, but over all **grades** in L.
  - To avoid repeated computations, we run over L in descending order.
- A new partial canonicity test is designed to take into account the adaptation of the *crisp* canonicity test to the *fuzzy* situation:
  - In the crisp setting, the canonicity test checks the intent up to a given attribute $j \in M$ (not including it).
  - In the fuzzy setting, we must take into consideration that the intent including j (with a given degree) can have appeared before.

Algorithm 1: InClose4b_ChildConcepts(A, B, y, P, $\mathbb{C}$)

Input: A: An extent; B: The intent corresponding to A, that will be completed in this execution; y: index of the attribute where to start the exploration of this branch; P: record for empty intersections; $\mathbb{C}$: the global variable where to accumulate the computed concepts.

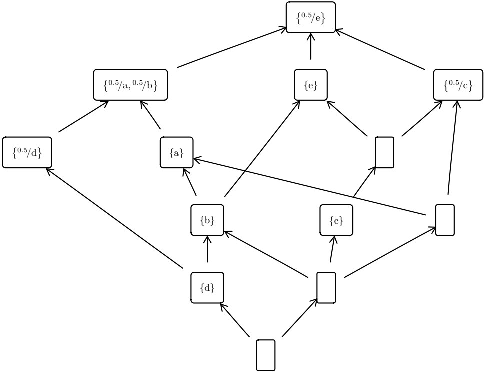
```
Q := ∅
for j ∈ {y + 1, ..., |M|} do
    for k ∈ {n, ..., 1} do
        g := l_k
        if B ∩ {m_j} ⊊ {g/m_j} and (P ∩ {m_j} = ∅ or {g/m_j} ⊊ P ∩ {m_j}) then
            C := A ∩ {g/m_j}^↓
            if C^↑ ∩ {m_j} = {g/m_j} then
                if C = ∅ then
                    P := (P ∖ {m_j}) ∪ {g/m_j}
                else
                    if C = A then
                        B := B ∪ {g/m_j}
                    else
                        if B ∩ M_j = C^{↑j} then
                            Q := Q ∪ {(C, j, k)}
ℂ := ℂ ∪ {(A, B)}
for (C, j, k) ∈ Q do
    D := B ∪ {l_k/m_j}
    InClose4b_ChildConcepts(C, D, j, P, ℂ)
```

Some Results

We use the same example as before:

	a	b	c	d	e
o1	0	0	½	0	1
o2	1	½	½	0	½
o3	½	½	0	½	½

Table 3: Formal context



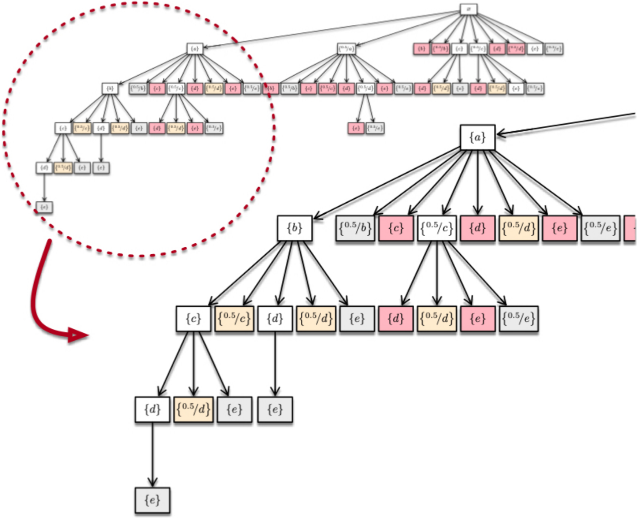


Figure 1: Order of computations performed by the fuzzy version of InClose2.

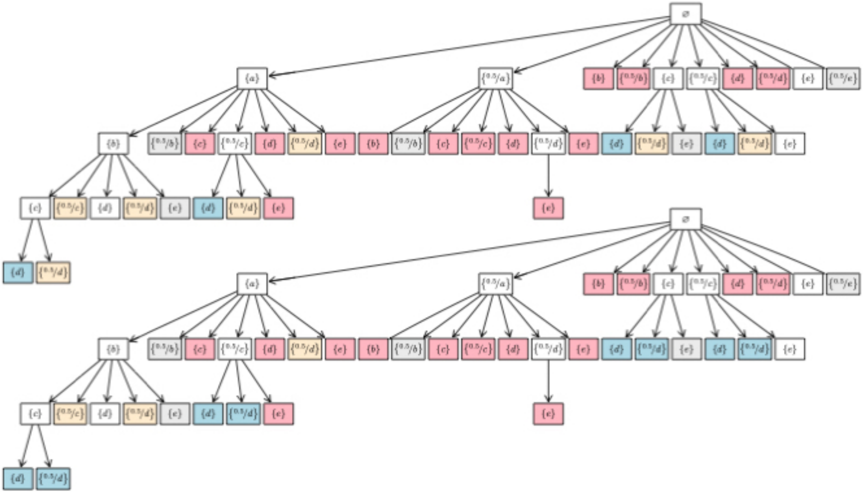


Figure 2: Order of computations performed by the fuzzy version of InClose4a (up) and InClose4b (down).

Table 4: Number of computations performed by each of the algorithms for the dataset.

Algorithm	Partial tests	Full tests	#Intents	#Extents
FuzzyInClose2	49	29	97	50
FuzzyInClose4a	41	29	89	42
FuzzyInClose4b	33	25	74	42

Conclusions and Future Works

- We have introduced extensions of algorithms of the InClose family to the fuzzy framework.
  - Different optimisations of the algorithm are taken into consideration.
  - The adaptation to the fuzzy setting requires to include partial canonicity tests and restructuring the construction of the computation tree.
  - We provide an example that shows how the different optimisations allow to improve the number of operations performed.
- Future work:
  - To study several optimisations to the InClose4 family, taking advantage of the structure of the degrees in L.
  - To explore generalisations of other algorithms, such as the FastCbO family or the NextNeighbour or NextPriorityConcept, for the fuzzy setting, along with different optimisations that could alleviate the greater computational cost when compared to the binary case.
  - To adapt these algorithms to compute the canonical basis of implications in this fuzzy setting.

References I

Simon Andrews. Making use of empty intersections to improve the performance of CbO-type algorithms. In *In.Conference on Formal Concept Analysis*, pages 56–71. Springer, 2017.

Radim Belohlavek and Jan Konecny. Fixpoints of fuzzy closure operators via ordinary algorithms. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, 2017.

Bernhard Ganter. Two basic algorithms in concept analysis. FB4-Preprint 831. Darmstadt, Germany: Technische Hochschule Darmstadt, 1984.

Petr Krajča, Jan Outrata, and Vilém Vychodil. Advances in algorithms based on CbO. In *CLA*, volume 672, pages 325–337. Citeseer, 2010.

Sergei Kuznetsov. A fast algorithm for computing all intersections of objects in a finite semi-lattice. *Automatic Documentation and Mathematical Linguistics*, 27:11–21, 1993.

Eugene M Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23 (2):243–250, 1978.

Mining Fuzzy Concept Lattices

CMMSE 2022

M. Ojeda-Hernández, **D. López-Rodríguez**, Á. Mora, P. Cordero

Departamento de Matemática Aplicada

UNIVERSIDAD
DE MÁLAGA