Finding Errors in New Object in Formal Contexts

Artem Revenko[1][2][*], Sergei O. Kuznetsov[2], and Bernhard Ganter[1]

[1] Technische Universität Dresden
Zellescher Weg 12-14, 01069 Dresden, Germany
[2] National Research University Higher School of Economics
Pokrovskiy bd. 11, 109028 Moscow, Russia
artem_viktorovich.revenko@mailbox.tu-dresden.de,
skuznetsov@hse.ru, bernhard.ganter@tu-dresden.de

Abstract. Classification of possible errors in formal contexts is given and the possibilities of exploring them are discussed. An approach for finding errors of some classes in formal contexts is introduced. This approach may be used in attempt to find errors in an object that is to be added to the context. The idea is based on finding those implications from an implication basis of the context, which are not respected by the object. It is noted that addressing such a problem directly may lead to an intractable solution. Alternative approach based on closing subsets of the intent of an object is considered in order to be able to find solution in polynomial time and deal with inconsistent combination of attributes. Examples are provided.

Keywords: formal context, implication, error exploration

1 Introduction

The work is motivated by the idea of building multi-user system based on Formal Concept Analysis methods. It would be different from QED project [QED] in the way that information should not be formalised as mathematical expressions and from Wikipedia in the way that information can somehow be inferenced by computer. In such a multi-user system error finding tools are absolutely necessary. In this work only errors in new objects (not yet added to the context) are considered. Throughout the text we assume that objects in the context are checked by an expert and correct. We attempt to find errors in new objects based on the information already in the context. This is actually the first step to building error finding tools.

2 Main Definitions

Let G and M be given sets. Let $I \subseteq G \times M$ be a binary relation between G and M. Triple $\mathbb{K} := (G, M, I)$ is called a *(formal) context.*
Set G is called a set of *objects.* Set M is called a set of *attributes.*

[*] We thank Sergei Obiedkov for discussion and useful remarks

Consider mappings $\varphi\colon 2^G \to 2^M$ and $\psi\colon 2^M \to 2^G$: $\varphi(A) := \{m \in M \mid gIm \text{ for all } g \in A\}$, $\psi(B) := \{g \in G \mid gIm \text{ for all } m \in B\}$. For any $A_1, A_2 \subseteq G$, $B_1, B_2 \subseteq M$ one has

1. $A_1 \subseteq A_2 \Rightarrow \varphi(A_2) \subseteq \varphi(A_1)$
2. $B_1 \subseteq B_2 \Rightarrow \psi(B_2) \subseteq \psi(B_1)$
3. $A_1 \subseteq \psi\varphi(A_1)$ and $B_1 \subseteq \varphi\psi(B_1)$

Mappings φ and ψ define a *Galois connection* between $(2^G, \subseteq)$ and $(2^M, \subseteq)$, i.e. $\varphi(A) \subseteq B \Leftrightarrow \psi(B) \subseteq A$. Usually, instead of φ and ψ a single notation $(\cdot)'$ is used. $(\cdot)'$ is sometimes called a *derivation operator.* For object $g \in G$ the set $g' = \{m \in M | gIm\}$ is called an *intent* of g and is denoted $\mathrm{int}(g)$. Similarly, for attribute $m \in M$ the set m' is called an *extent* of m and is denoted $\mathrm{ext}(m)$. Let $\overline{M} = \{\overline{m} \mid m \in M\}$ and $\overline{I} = \{(g, m) \mid g \in G, m \in M, (g, m) \notin I\}$. Triple $\overline{\mathbb{K}} := (G, \overline{M}, \overline{I})$ is called a *complementary (formal) context.*
Let $B \subseteq M, g \in G, \overline{B} := \{\overline{b} \mid b \in B\}$. $\overline{B} \subseteq \mathrm{int}(g)$ means that in $\mathbb{K} = (G, M, I)$ object g is not related to all attributes from B.

An *implication* of $\mathbb{K} := (G, M, I)$ is defined as a pair (A, B), written $A \to B$, where $A, B \subseteq M$. A is called a *premise*, B is called a *conclusion.* Implication $A \to B$ is *respected by a set of attributes* N if $A \nsubseteq N$ or $B \subseteq N$. Implication $A \to B$ holds (is valid) in $\mathbb{K}$ if $A' \subseteq (B \cup A)'$ or $A' \subseteq B'$, i.e. every object, that has all the attributes from A, also has all the attributes from B. The implications of $\mathbb{K}$ satisfy *Armstrong rules*:

$$\frac{}{X \to X}\ , \quad \frac{X \to Y}{X \cup Z \to Y}\ , \quad \frac{X \to Y, Y \cup Z \to W}{X \cup Z \to W}$$

A *support* of an implication is the set of object, whose intents respect this implication.

An *implication basis* of context $\mathbb{K}$ is defined as a set $\mathfrak{L}$ of implications of $\mathbb{K}$, from which any valid implication for $\mathbb{K}$ can be deduced by the Armstrong rules and none of the proper subsets of $\mathfrak{L}$ has this property.
A minimal implication basis is an implication basis minimal in the number of implications. A minimal implication basis was defined in [GD86] and is known as *canonical implication basis.* In paper [Gan84] the premises of implications from canonical base were characterized in terms of pseudo-intents. A subset of attributes $P \subseteq M$ is called *pseudo-intent*, if $P \neq P''$ and for every such pseudo-intent Q such that $Q \subset P$, one has $Q'' \subset P$. Canonical implication basis looks then as follows: $\{P \to (P'' \setminus P) \mid P$ - pseudo-intent$\}$.

3 Classification and Exploration of Errors

Every object in the context is described by the set of attributes, that are related to this object. In the "real world" there may exist dependencies between attributes. Consider possible cases in terms of implications:

1. Valid in "real world" dependency $A \to B,\ A, B \subseteq M$ is not respected by an object

2. Valid in “real world” dependency $A \to \overline{B},\ A, B \subseteq M$ is not respected by an object
3. Combination of two above cases, i.e. valid in “real world” dependency $A \to \overline{B} \wedge C,\ A, B, C \subseteq M$ is not respected by an object
4. Valid in “real world” dependency $A \to b \vee c,\ A \subseteq M, b, c \in M$ is not respected by an object
5. Valid in “real world” dependency $A \to \mathbf{F}$, where $A \subseteq M, \mathbf{F}$ is any logical formula not considered above, is not respected by an object (for example, $\mathbf{F} = a \vee (b \wedge \overline{c})$)

Unfortunately, it turns out that not all possible errors might be found using implications of a context. Namely, case 4 corresponds to reducible object in a context, while it is known that reducible objects change neither the lattice nor the implication basis of a context (definitions of reducible objects and lattices of contexts are not given in this paper for the sake of compactness, for definitions and further information see [GW99]).
Every implication $A \to B$ can be regarded as a conjunction of implications $A \to B_1$ and $A \to B_2,\ B_1 \cup B_2 = B$. Thereby, in Case 5 in $\mathbf{F}$ top level conjunctions can be dealt with easily. However, as in Case 4 we do not know how to reveal errors that have disjunctions in their conclusion.

Let $\mathcal{L}$ be a set of all implications valid for a context $\mathbb{K}$. From any implication basis any valid implication for the context should be deducible by definition. It means that if an object does not respect an implication from $\mathcal{L}$, then it should not respect an implication from implication basis of the context. Then an expert is asked whether this implication is valid. If he accepts this implication, then the object is an error. All the errors of the first class are caught using this approach.

4 An Example

The formal context on Fig. 1 shows the properties of convex quadrangles. The context is not full, i.e. not all possible convex quadrangles are considered, and some objects in the context are reducible (do not bring new information in an implication basis of the context). 7 attributes are considered. Attributes ‘has equal legs’ and ‘has equal angles’ require at least two angles/legs of a quadrangle to be equal. Some dependencies on attributes are obvious, for example, it is clear that if all angles are equal in a quadrangle then this quadrangle definitely has equal angles.
4 errors are presented on Fig 2. Errors are added to the context on Fig. 1 one at a time. One should treat an error as an object to be added to the context.
The context without errors on Fig. 1 is denoted $\mathbb{K}$, $(\cdot)'$ is the corresponding derivation operator.
The context of errors on Fig. 2 is denoted $\mathbb{K}_{\mathrm{e}}$, $(\cdot)^{\mathrm{e}}$ is the derivation operator for $\mathbb{K}_{\mathrm{e}}$.

Example 1. $\{$Error 1$\}^{\mathrm{e}}$ =$\{$has equal legs, at least 3 different angles, all legs equal$\}$
$\{$Error 1$\}^{\mathrm{e}''}$ =$\{$all angles equal, all legs equal, at least 3 different angles, at least 3 different legs, has equal angles, has equal legs, has right angle$\}$

Convex quadrangles	has equal legs	has equal angles	has right angle	all legs equal	all angles equal	at least 3 different angles	at least 3 different legs
Square	×	×	×	×	×		
Rectangle	×	×	×		×		
Quadrangle						×	×
Rhombus	×	×		×			
Parallelogram	×	×					
Rectangular trapezium		×	×			×	×
Quadrangle with 2 equal legs and right angle	×		×			×	×
Isosceles trapezium	×	×				×	
Rectangular trapezium with 2 equal legs	×	×	×			×	×
Quadrangle with 2 equal angles		×				×	×
Quadrangle with 2 equal legs	×					×	×
Quadrangle with 2 equal legs and 2 equal angles	×	×				×	×

Fig. 1. Context of convex quadrangles $\mathbb{K}$

Errors	has equal legs	has equal angles	has right angle	all legs equal	all angles equal	at least 3 different angles	at least 3 different legs
Error1	×			×		×	
Error2	×		×	×	×		
Error3		×	×	×	×	×	×
Error4	×	×		×			×

Fig. 2. Context of errors $\mathbb{K}_e$

Canonical basis of the context on Fig. 1 looks as follows:

1. at least 3 different angles $\rightarrow$ at least 3 different legs
2. all angles equal $\rightarrow$ has equal angles, has equal legs, has right angle
3. all legs equal $\rightarrow$ has equal angles, has equal legs
4. has right angle, at least 3 different legs $\rightarrow$ at least 3 different angles
5. has equal angles, has equal legs, at least 3 different legs, all legs equal $\rightarrow$ has right angle,at least 3 different angles, all angles equal
6. has equal angles, has equal legs, at least 3 different legs, all angles equal, has right angle, at least 3 different angles $\rightarrow$ all legs equal
7. has right angle, has equal legs, all legs equal, has equal angles $\rightarrow$ all angles equal

Consider Error2.
$\{\text{Error 2}\}^{\text{e}}$ =$\{$has equal legs, has right angle, all legs equal, all angles equal$\}$
This object does not respect Implications 2 and 3. It is easy to see that both implications are valid in "real world". Thereby, an expert recognizes object as an error.

5 Improvements

Although this approach gives the needed result, there are some problems remaining. The problem of producing canonical basis with known algorithms is intractable. Recent theoretical results suggest that existing approaches for computing the stem base may not lead to algorithms with better worst-case complexity [DS11], [BK10]. One can use other bases (for example, advances were achieved in computing proper premises [RDB11]), but known algorithms are still too costly and non minimal bases do not guarantee to ask an expert minimal yet sufficient number of questions.
Since we are only interested in implication corresponding to an object, it is not necessary to compute a whole implication basis. Only the closure of object's intent may be considered. From monotonicity of the closure operator of the context it follows that we do not lose any attributes that are erroneously not related to an object. Nevertheless, the following case is possible. Let a set $H \subseteq M$ be the intent of an object such that $\nexists g \in G : H \subseteq g'$. In this case $H'' = M$ and the implication $H \rightarrow H'' \setminus H$ has empty support. This is the case if an object is an error of the second class, because in its intent impossible in "real world" combination of attributes is contained. Although it is not the best solution, but we can ask an expert if the combination of attributes in object's intent is consistent. In such a question we use information already input in the context, but an expert should consider all possible combinations of attributes to be excessive. Further on this question is not sufficient to reveal an error of the first class.
A better idea would be to investigate the subset of object's intent. Even if $H'' = M$, we are still able to reveal an error of the first class, because we examine all possible dependencies on the subset of object's intent that were satisfied by intents in the context. Searching through all the subsets of object's intent leads

to exponential time solution. Since we are only interested in such subsets that are contained in at least one intent in the context, we may consider only the intersections of object's intent with intents in the context. This allows us to find errors of the first class in polynomial time.
But we can do even better and replace the question about the consistency of set of attributes in object's intent with an implication. This idea is better because in case of implication an expert is explicitly shown an attribute that breaks the dependencies satisfied in the context. For this purpose we should consider complementary context. If we investigate the closures of the subsets of initial object's intent in the context $\mathbb{K}_\cup := \{G, M \cup \overline{M}, I \cup \overline{I}\}$, then we are also able to find errors of the second class in the very same manner, that we discussed before for the first class. Thus we are able to find all the errors of first three classes.

5.1 Pseudocode

Below is presented the pseudocode of the method described above.

```
inspect_with_negations($\mathbb{K}$:=(G, M, I), H$\subseteq$M)
1. Candidates = {object$'$$\cap$H | object$\in$G}
2. Candidates = {candidate$\in$Candidates |
                 $\nexists$c$\in$Candidates: candidate$\subseteq$c}
3. Result = $\emptyset$
4. for candidate in Candidates:
    5. if candidate$'^{\cup}'^{\cup}$ != candidate:
        6. Result.add(candidate $\to$ candidate$'^{\cup}'^{\cup}$$\setminus$(candidate$\cup$H))
7. return Result
```

H is the intent of an object to be added to the context $\mathbb{K}$. In the first line we compute the set of all subsets that could produce desired implication. Since closure operator is monotone we may consider only maximal elements of Candidates. In the second line we discard all the non-maximal elements. In line 4 - 6 we check if closure differs from the intersection, i.e. there are attributes in conclusion of future implication. Here $(\cdot)'^{\cup}$ denotes derivation operator of the context $\mathbb{K}_\cup$. There is no need to check if candidate$''$ is contained in H, because all the attributes from H$\setminus$candidate are not contained in the intent of at least one object, from which this candidate was generated in the first line.
It is possible to provide further optimisation. In the first line we can stop generating Candidates after all the maximal subsets satisfying condition were found.

6 Results

FCA package for Python written by Nikita Romashkin was used for implementation [Rom].
The name inspect_dg is used to denote the function implementing the method described in Section 3.

Inspecting Error1:

`inspect\_dg`

at least 3 different angles → at least 3 different legs

all legs equal → has equal angles, has equal legs

`inspect\_with\_negation`

has equal legs, at least 3 different angles → at least 3 different legs, $\overline{\text{all legs equal}}$

has equal legs, all legs equal → has equal angles, $\overline{\text{at least 3 different angles}}$

Both implications in the result of v without overlined attributes in conclusions are deducible from the two implications in the result of `inspect_dg`. The two implications in the result of `inspect_dg` with the intent of Error1 added in the premise are deducible from the implications in the result of `inspect_with_negation`. Considering a particular object and corresponding implications we can always add object's intent to the premise(s), because attributes from its intent are always related to the object.
Errors of the second class are not caught using canonical base. As described above such errors correspond to inconsistent in the context combination of attributes. Nevertheless, `inspect_with_negation` catches such errors.

Inspecting Error2:

`inspect\_dg`

all angles equal → has equal angles, has equal legs, has right angle

all legs equal → has equal angles, has equal legs

`inspect\_with\_negation`

has right angle, has equal legs, all legs equal, all angles equal → has equal angles

In this example we are able to ask even less number of questions to the expert using `inspect_with_negation` as with `inspect_dg`. This is the result of finding implications generated by maximal subsets of object's intent. Again, adding object's intent to the premises of implications in the result of `inspect_dg` makes both groups mutually deducible.

Inspecting Error3:

`inspect\_dg`

all angles equal → has equal angles, has equal legs, has right angle

all legs equal → has equal angles, has equal legs

`inspect\_with\_negation`

has equal angles, has right angle, at least 3 different legs, at least 3 different angles → $\overline{\text{all angles equal}}$, $\overline{\text{all legs equal}}$

has equal angles, has right angle, all legs equal, all angles equal → has equal legs, $\overline{\text{at least 3 different angles}}$, $\overline{\text{at least 3 different legs}}$

The case of Error3 is more or less the same, as the case of Error1. The two groups of implications are mutually deducible under the same conditions as before.

Inspecting Error4:

`inspect\_dg`

has equal angles, has equal legs, at least 3 different legs, all legs equal $\rightarrow$ has right angle, at least 3 different angles, all angles equal

`inspect\_with\_negation`

has equal angles, has equal legs, all legs equal $\rightarrow \overline{\text{at least 3 different legs}}$
has equal angles, has equal legs, at least 3 different legs $\rightarrow \overline{\text{all legs equal}}$

Error4 is a very special case when corresponding implication from canonical basis has empty support. In this case even if implication from the result of `inspect_dg` is rejected by an expert object may still be an error. This implication is in fact excessive, because the premise is not contained in any intent in the context and all attributes, that are not in premise, are in conclusion. Using `inspect_with_negation` we are able to ask an expert more sensible questions. Unfortunately, groups of implications are not deducible from each other in this example.

7 Conclusion

An algorithm for finding errors in new objects of the context was proposed. As opposed to finding not respected implications in an implication basis proposed algorithm finishes in polynomial time. Moreover, in case of inconsistent combination of attributes in object's intent it is possible to state more sensible questions. In some cases the number of produced questions to an expert is less than the number of not respected implications in the canonical basis of the context.

References

[BK10] M. Babin and S. O. Kuznetsov. Recognizing pseudo-intent is conp-complete. *Proc. 7th International Conference on Concept Lattices and Their Applications, University of Sevilla*, pages 294–301, 2010.

[DS11] Felix Distel and Barış Sertkaya. On the complexity of enumerating pseudo-intents. *Discrete Applied Mathematics*, 159(6):450–466, 2011.

[Gan84] B. Ganter. Two basic algorithms in concept analysis. *Preprint-Nr. 831*, 1984.

[GD86] J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Math. Sci. Hum*, 24(95):5–18, 1986.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis : Mathematical Foundations.* Springer, 1999.

[QED] The qed project. http://mizar.org/qed/.

[RDB11] Uwe Ryssel, Felix Distel, and Daniel Borchmann. Fast computation of proper premises. In Amedeo Napoli and Vilem Vychodil, editors, *International Conference on Concept Lattices and Their Applications*, pages 101–113. INRIA Nancy – Grand Est and LORIA, 2011.

[Rom] Nikita Romashkin. Python package for formal concept analysis. https://github.com/jupp/fca.