Evaluating Formal Concept Analysis Software for Anomaly Detection and Correction

Nassif Saab, Marianne Huchard, Pierre Martin

Evaluating Formal Concept Analysis Software for Anomaly Detection and Correction

Nassif Saab[1], Marianne Huchard[1], and Pierre Martin[2]

[1] LIRMM, Univ Montpellier, CNRS, Montpellier, France
{nassif.saab@lirmm.fr, marianne.huchard@lirmm.fr}
[2] CIRAD, UPR AIDA, F-34398 Montpellier, France
AIDA, Univ Montpellier, CIRAD, Montpellier, France
pierre.martin@cirad.fr

Abstract. Data cleaning is a process that precedes data mining. Particularly, in our dataset on pesticidal plant use, several types of anomalies were identified, ranging from incorrect values to a lack of data susceptible of causing users to draw wrong conclusions during its exploration. Literature presents three methods based on Formal Concept Analysis (FCA), *i.e.* implication rules computation, association rules computation, and attribute exploration, that may allow the detection and correction of anomalies. This paper evaluates 30 FCA-based software and their apposite features to the development of an anomaly detection and correction method applicable to our dataset. Results show that only ConExp and its reimplementations provide all three methods. Since the data model on plant use is relational but ConExp only allows formal contexts as input, this paper concludes on the importance of integrating Relational Concept Analysis (RCA) with ConExp in future work.

Keywords: Formal Concept Analysis · Software evaluation · Anomaly detection · Anomaly correction · Data cleaning

1 Introduction

Data cleaning is a basic operation of the knowledge discovery process [5]. Particularly, the Knomana Knowledge Base (KKB) [15] includes 45,000 descriptions of pesticidal plant use extracted from literature. In Knomana, a description of plant use is made of 71 data types, for example, the pesticidal plant name, the location and part of the plant used to build essential oils. Various anomalies were observed within a description, such as incorrect spellings of words, *e.g.* plant names, and wrong types of values, *e.g.* integers where strings are expected.

Additionally, computing the Duquenne-Guigues basis of implications [8] for Knomana allowed [14] to identify another type of anomaly, *i.e.* the lack of data that may cause users of KKB to draw wrong conclusions regarding plant use. For instance, the presence of only one pesticidal plant to treat a disease may prompt a user to conclude that this disease can only be treated using this plant. Thereupon, anomalies should be detected and corrected before providing the user with software to explore KKB.

To this end, we plan to develop an Anomaly Detection and Correction (ADC) software. Our ADC method will be based on Formal Concept Analysis (FCA) [7]. Since the Knomana data model employs ternary relationships [12], this method will rely on Relational Concept Analysis (RCA) [13], an extension of FCA intended for datasets conforming to the entity-relationship model. A preliminary step in this development process is to assess whether existing FCA-based software support ADC. Software that fit this description and meet other requirements, including being cross-platform, will be adapted for RCA in future work.

The objective of this paper is to identify such software through an evaluation of several criteria. Section 2 presents FCA-based ADC methods found in literature. Section 3 introduces the evaluated software, the dataset and the criteria used for the evaluation. Section 4 shows the results of the evaluation. Section 5 concludes the paper and describes the next step in our work.

2 FCA-based methods in literature

This section reviews the literature on ADC and introduces three ADC methods based on FCA. Literature presents various ADC approaches, including statistics [1], neural networks [9], deep learning [10], clustering [11], nearest neighbor search [16], and the following three FCA-based ADC methods.

The authors in [2] improve Resource Description Framework data by employing implications rules. Since the confidence percentage of an implication is 100%, the confidence of its inversion informs the user of its proximity to being a definition. Thus, an inverted rule with a high confidence suggests a need for additional data to complete the set of triples.

The method presented in [4] combines FCA and association rules to spot faults during the software debugging process. Execution traces are first used to compute association rules, which are then transformed into a formal context (with the source code line numbers as attributes) and the corresponding concept lattice is computed. Concepts located at the bottom of the lattice contain rules that are too specific to explain the error, and the ones at the top contain concepts common to all failed executions.

In [3], attribute exploration (detailed in [6]) is applied to a human healthcare dataset in order to show that knowledge acquired through the exploration process and knowledge provided by domain experts, if combined, can improve the classification accuracy. This is an interactive method based on the computation of the Duquenne-Guigues basis of implications. The role of the expert in this process is to validate each rule. When an invalid rule is presented, the expert provides a counterexample.

Each of these papers explores a method adapted to managing potential anomalies: [2] and [4] compute implication and association rules, respectively. The authors in [3] perform attribute exploration, thus supplementing the computed implication rules with user knowledge to correct the anomalies. Since we intend to include these three methods in our future ADC software, they are considered in the evaluation conducted in the sections that follow.

3 Methods

This section introduces the evaluated software, the dataset and the criteria used for the evaluation. The evaluated software are 30 of the listed FCA software on U. Priss' website[3]. Table 1 describes the assessed versions. Having a cross-platform software[4] facilitates its distribution, and therefore, verifying this criterion is important for the ADC software development described in Section 1.

Table 1. Description of the evaluated FCA software.

Software	Operating system	License	Programming language	Version
Camelis	Cross-platform	GNU GPL	Objective Caml	v.1.4.3, 18 Dec. 2009
Colibri Concepts	Cross-platform	GPL-2.0	C, Roff, Yacc, Lex, Bash	18 Sept. 2007
concepts by S. Bank	Cross-platform	MIT	Python	v.0.9.2, 8 June 2020
concepts.py by D. Endres	Cross-platform	*N/A*	Python	*N/A*
ConExp	Cross-platform	BSD	Java	v.1.3, 12 Sept. 2006
conexp-fx	Cross-platform	GPL-3.0	Java, CSS, Scala, Bash	v.5.5.1, 12 Sept. 2019
conexp-clj	Cross-platform	EPL-1.0	Clojure, Java, C#, Python	v.2.0.0-rc1, 30 June 2019
ConExp-NG	Cross-platform	GPL-3.0	Java	v.0.7.0, 05 Sept. 2014
FCA algorithms FCbO	Cross-platform	GPL-2.0	C	5 Oct. 2010
FCA algorithms IterEss	Cross-platform	GPL-2.0	C	29 Nov. 2017
FCA algorithms PCbO	Cross-platform	GPL-2.0	C	3 Mar. 2009
FCA4J	Cross-platform	Apache-2.0	Java	v.0.4, Mar. 2022
FcaBedrock	Windows NT	MIT	Visual Basic .NET	Build 2.8.28, 12 June 2014
fcaR	Cross-platform	GPL-3.0	R, C++	v.1.1.1 (CRAN)
FCART	Windows NT	*N/A*	*N/A*	v.0.9.5, Mar. 2016
FcaStone	Cross-platform	GNU GPL	Perl, HTML	Mar. 2022
GALACTIC	Cross-platform	BSD 3-Clause	Python	v.0.5.0.dev5, 14 Feb. 2022
Galicia	Cross-platform	GPL-2.0	Java	v.3.2, 13 Dec. 2005
Griff (Sarl3 library)	Cross-platform	LGPLv2	C, C++, Ruby	v.0.9, 06 Jan. 2006
In-Close4	Cross-platform	MIT	C++	18 July 2017
Lattice Miner	Cross-platform	Apache-2.0	Java	v.2.0, 14 Apr. 2017
Lattice navigator	Windows NT	AGPL-3.0	C#	v.3.6.6.0, 27 July 2011
MIW	Cross-platform	*N/A*	Python	24 Dec. 2014
OpenFCA	Windows NT	MIT[5]	ActionScript, C#, JavaScript, HTML	10 Apr. 2011
Python FCA Tool	Cross-platform	LGPL	Python	v.0.0.1, 3 Mar. 2012
qdfca	Cross-platform	Unlicense	Ruby	v.1.0, 20 Feb. 2015
RCAExplore	Cross-platform	LGPL	Java	12 Oct. 2015
The Coron System	Cross-platform	Free Software	Java, Bash, Perl	v.0.8, 19 Jan. 2010
Tockit (CASS toolkit)	Cross-platform	BSD	Java	28 June 2007
ToscanaJ	Cross-platform	BSD-style	Java	v.1.7, 2012

N/A : Not Available

The evaluation was conducted using the example of formal context provided by [6] on eight European countries (Albania, Vatican, Switzerland, Austria, Cyprus, San Marino, Liechtenstein, and Sweden) and their memberships regarding seven organisations (European Union, Council of Europe, European Economic Area, European Free Trade Association, EU Custom Union, Eurozone, and Schengen area). Interest in this dataset derives from the size of the formal context, small enough not to crash the software while still outputting

[3] https://upriss.github.io/fca/fcasoftware.html
[4] A software executable on most Linux distributions, macOS, and Windows NT.
[5] Except the SpringGraph component of the software.

meaningful results. Additionally, the implications and the attribute exploration process are fully described by the authors.

Software were evaluated according to five criteria: the first two, *i.e.* context editing and lattice building, consider FCA capabilities. Each of the last three, *i.e.* computing implication rules, computing association rules, and performing attribute exploration, corresponds respectively to an ADC method presented in Section 2, *i.e.* a method introduced by [2], [4], or [3].

4 Results

This section presents the results of the evaluation. Table 2 lists 20 software that were tested, thus omitting ten software from Table 1 for the following reasons. In March 2022 when the evaluation was conducted, concepts.py was not available, The Coron System required the purchase of a license to access its source code, and the license of MIW was not found. FcaBedrock, FCART, Lattice navigator, and OpenFCA were excluded as they do not appear to be cross-platform. The three FCA algorithms are command-line tools that compute formal concepts and maximal frequent itemsets which appear in mining nonredundant association rules. Nonetheless, these algorithms do not generate rules *per se*, nor do they allow context editing, lattice building, or attribute exploring.

Table 2. FCA software and their anomaly detection and correction features.

Software	Context editing	Lattice building	Implication rules computing	Association rules computing	Attribute exploring
Camelis	x	x			
colibri-concepts		x			
concepts by S. Bank	x	x			
ConExp	x	x	x	x	x
conexp-fx	x	x	x	x	x
conexp-clj	x	x	x	x	x
ConExp-NG	x	x	x	x	x
FCA4J	x	x	x		
fcaR	x	x	x		
FcaStone	x	x			
GALACTIC	x	x	x		
Galicia	x	x	x	x	
Griff (Sarl3 library)	x	x	x		
In-Close4	x	x			
Lattice Miner	x	x	x	x	
Python FCA Tool	x	x	x		
qdfca		x			
RCAExplore	x	x			
Tockit (CASS toolkit)	x	x			
ToscanaJ		x			

Table 2 shows nine software allowing context editing and/or lattice building but without providing any of the three ADC methods. Nevertheless, these software hold other features[6]. For instance, ToscanaJ supports multi-level data analysis and provides a database viewer.

Whilst ConExp, FCA4J, Griff, and Python FCA Tools only compute the Duquenne-Guigues basis of implications, GALACTIC can also compute other bases of implications. Lattice Miner introduces the computation of implications with negation and the calculation of triadic implications. fcaR performs fuzzy formal concept analysis, thus computing implications and semantic closures of fuzzy sets. Galicia deduces implications by filtering the set of association rules and only keeping those with a confidence of 100%. Finally, ConExp and its reimplementations, *i.e.* -fx, -clj, and -NG, enable attribute exploration. ConExp relies on binary contexts to compute the implications and associations and perform attribute exploration.

5 Conclusion

This paper is a step prior to developing a software intended for the detection and correction of anomalies in the Knomana Knowledge Base. The evaluation of FCA-based software listed on U. Priss' website showed that ConExp and its reimplementations are the only cross-platform software allowing the computation of implication and association rules, as well as attribute exploration. In the literature on data cleaning, these three methods are used as basis to develop dataset-specific methods for anomaly detection and correction.

In the context of Knomana, anomalies are defined by missing data, as well as incorrect values that describe plant uses. We need to investigate the extent to which ConExp is appropriate for the detection and correction of the different types of anomalies found in Knomana. Since the Knomana data model employs ternary relationships to describe plant uses, we plan to mine this dataset through Relational Concept Analysis (RCA). Accordingly, we plan to integrate RCA with ConExp to obtain a data cleansing system applicable to Knomana.

Acknowledgments This work is supported by Montpellier University KIM (Key Initiatives MUSE) DATA & LIFE SCIENCES through an interdisciplinary internship grant[7].

References

1. Aggarwal, C.C.: Probabilistic and statistical models for outlier detection. In: Outlier Analysis, pp. 35–64. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-47578-3_2

[6] Additional information can be found at https://kodovnash.github.io/FCA_software/
[7] https://muse.edu.umontpellier.fr/key-initiatives-muse/

2. Alam, M., Buzmakov, A., Codocedo, V., Napoli, A.: Mining Definitions from RDF Annotations Using Formal Concept Analysis. In: International Joint Conference in Artificial Intelligence. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina (2015), https://hal.archives-ouvertes.fr/hal-01186204
3. Annapurna, J., Cherukuri, A.K.: Exploring attributes with domain knowledge in formal concept analysis. Journal of Computing and Information Technology **21**(2), 109 (2013). https://doi.org/10.2498/cit.1002114
4. Cellier, P.: Formal concept analysis applied to fault localization. In: Companion of the 13th international conference on Software engineering - ICSE Companion '08. ACM Press (2008). https://doi.org/10.1145/1370175.1370220
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM **39**(11), 27–34 (1996). https://doi.org/10.1145/240455.240464
6. Ganter, B., Obiedkov, S.: Conceptual Exploration. Springer Berlin Heidelberg (2016). https://doi.org/10.1007/978-3-662-49291-8
7. Ganter, B., Wille, R.: Formal Concept Analysis. Springer Berlin Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
8. Guigues, J.L., Duquenne, V.: Famille minimale d'implications informatives résultant d'un tableau de données binaires. Math. et Sci. Hum. **24**(95), 5–18 (1986)
9. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review **22**(2), 85–126 (2004). https://doi.org/10.1023/b:aire.0000045502.10941.a9
10. Kakanakova, I., Stoyanov, S.: Outlier detection via deep learning architecture. In: Proceedings of the 18th International Conference on Computer Systems and Technologies. ACM (2017). https://doi.org/10.1145/3134302.3134337
11. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. The VLDB Journal The International Journal on Very Large Data Bases **8**(3-4), 237–253 (2000). https://doi.org/10.1007/s007780050006
12. Mahrach, L., Gutierrez, A., Huchard, M., Keip, P., Marnotte, P., Silvie, P., Martin, P.: Combining implications and conceptual analysis to learn from a pesticidal plant knowledge base. In: Graph-Based Representation and Reasoning. pp. 57–72. Springer International Publishing (2021). https://doi.org/10.1007/978-3-030-86982-3_5
13. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: Relational concept analysis: mining concept lattices from multi-relational data. Annals of Mathematics and Artificial Intelligence **67**(1), 81–108 (2013). https://doi.org/10.1007/s10472-012-9329-3
14. Saoud, J., Gutierrez, A., Huchard, M., Marnotte, P., Silvie, P., Martin, P.: Explicit versus Tacit Knowledge in Duquenne-Guigues Basis of Implications: Preliminary Results (2021), https://hal.archives-ouvertes.fr/hal-03274757
15. Silvie, P.J., Martin, P., Huchard, M., Keip, P., Gutierrez, A., Sarter, S.: Prototyping a knowledge-based system to identify botanical extracts for plant health in subsaharan africa. Plants **10**(5) (2021). https://doi.org/10.3390/plants10050896
16. Yamanishi, K., ichi Takeuchi, J., Williams, G., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. Data Mining and Knowledge Discovery **8**(3), 275–300 (2004). https://doi.org/10.1023/b:dami.0000023676.72185.7c