Analysis of Large Data Sets using Formal Concept Lattices

Simon Andrews and Constantinos Orphanides

Conceptual Structures Research Group
Communication and Computing Research Centre
Faculty of Arts, Computing, Engineering and Sciences
Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk corphani@my.shu.ac.uk

Abstract. Formal Concept Analysis (FCA) is an emerging data technology that has applications in the visual analysis of large-scale data. However, data sets are often too large (or contain too many formal concepts) for the resulting concept lattice to be readable. This paper complements existing work in this area by describing two methods by which useful and manageable lattices can be derived from large data sets. This is achieved though the use of a set of freely available FCA tools: the context creator *FcaBedrock* and the concept miner *In-Close*, that were developed by the authors, and the lattice builder *ConExp*. In the first method, a sub-context is produced from a data set, giving rise to a readable lattice that focuses on attributes of interest. In the second method, a context is mined for 'large' concepts which are then used to re-write the original context, thus reducing 'noise' in the context and giving rise to a readable lattice that lucidly portrays a conceptual overview of the large set of data it is derived from.

1 Introduction

It has been shown that a variety of data sets can be converted into formal contexts [7,2] by a process of discretising and Booleanising the data. However, data sets of only modest size can produce contexts containing hundreds of thousands of formal concepts [9], making the resulting concept lattices unreadable and unmanageable. Perhaps more pertinent than size, however, is the density of and 'noise' in a context; factors that increase the number of formal concepts. There is also an issue in computing large numbers of formal concepts; much of the existing software are not capable of carrying out this task on a large scale. Tools such as ToscanaJ [5] and Concept Explorer (ConExp) [14] exist that compute and visualise concept lattices but are not designed to do so for large numbers of concepts.

This paper describes two ways in which concept lattices can be produced from data sets: 1) by creating sub-contexts by restricting the conversion of the data to information of interest, and 2) by removing relatively small concepts from a context to reduce 'noise', so that a readable, yet still meaningful, concept lattice can be produced.

2 Analysis of Sub-Contexts from Data Sets

FcaBedrock[1] is a freely available tool developed by the authors that converts *csv* format data files into formal context *cxt* files and FIMI data format files [3]. It reads a data file and automatically converts each many-valued data attribute in the file to formal attributes. The process is guided by the user in deciding how the data set should be interpreted. The user can specify, for example, discrete ranges for continuous data attributes and what names should be given in the cxt file to the formal attributes. The user can also create sub-contexts by restricting the conversion to only the data attributes of interest for a particular analysis. Such meta-data, used to guide the conversion process, is stored in a separate file called a *Bedrock* file. These files can be loaded into FcaBedrock to repeat the conversion, or allow changes in the meta-data to be made to produce different sub-contexts for alternative analyses.

2.1 Attribute Exclusion

To illustrate the production of concept lattices from data sets using sub-contexts, the well-known *Mushroom* and *Adult* data sets from the *UCI Machine Learning Repository* [4] will be used. The Mushroom data set contains data of 8124 edible and poisonous mushrooms of the families *agaricus* and *lepiota*. It is many-valued categorical data with attributes describing properties such as stalk shape, cap colour and habitat. As an example of the agaricus family of mushrooms, Figure 1 is of a meadow agaricus.

Fig. 1. A meadow agaricus mushroom (source `www.50birds.com/gmushrooms1.htm`)

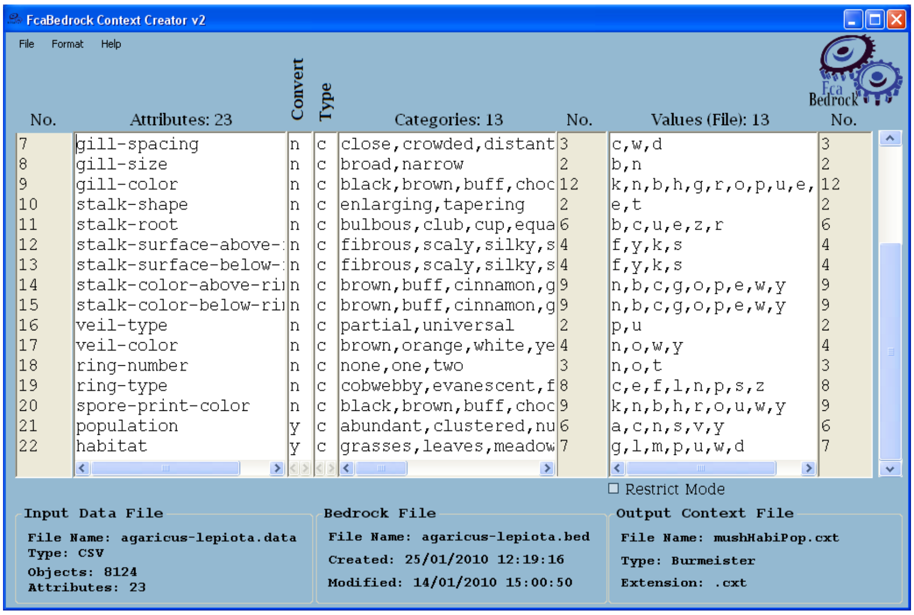
The data set was converted by FcaBedrock using all of the attributes and their categories. The resulting cxt file was processed by a formal concept miner developed by one of the authors, called *In-Close*[2] [1], generating over 220,000 concepts; far too many to visualise.

However, let us say we are interested in the relationship between mushroom habitat and population type. Figure 2 shows the meta-data for the Mushroom data set loaded into FcaBedrock. The *Convert* column was used to select only habitat and population to convert. In this way, a Mushroom sub-context was created containing formal attributes only for habitat and population. There were 13 formal attributes in all, corresponding to the 7 categories of habitat: *grasses*,

[1] `https://sourceforge.net/projects/fcabedrock`

[2] `https://sourceforge.net/projects/inclose`

leaves, *meadows*, *paths*, *urban*, *waste* and *woods*, and 6 categories of population type: *abundant*, *clustered*, *numerous*, *scattered*, *several* and *solitary*.



No.	Attributes: 23	Convert	Type	Categories: 13	No.	Values (File): 13	No.
7	gill-spacing	n	c	close,crowded,distant	3	c,w,d	3
8	gill-size	n	c	broad,narrow	2	b,n	2
9	gill-color	n	c	black,brown,buff,choc	12	k,n,b,h,g,r,o,p,u,e,	12
10	stalk-shape	n	c	enlarging,tapering	2	e,t	2
11	stalk-root	n	c	bulbous,club,cup,equa	6	b,c,u,e,z,r	6
12	stalk-surface-above-:	n	c	fibrous,scaly,silky,s	4	f,y,k,s	4
13	stalk-surface-below-:	n	c	fibrous,scaly,silky,s	4	f,y,k,s	4
14	stalk-color-above-ri	n	c	brown,buff,cinnamon,g	9	n,b,c,g,o,p,e,w,y	9
15	stalk-color-below-ri	n	c	brown,buff,cinnamon,g	9	n,b,c,g,o,p,e,w,y	9
16	veil-type	n	c	partial,universal	2	p,u	2
17	veil-color	n	c	brown,orange,white,ye	4	n,o,w,y	4
18	ring-number	n	c	none,one,two	3	n,o,t	3
19	ring-type	n	c	cobwebby,evanescent,f	8	c,e,f,l,n,p,s,z	8
20	spore-print-color	n	c	black,brown,buff,choc	9	k,n,b,h,r,o,u,w,y	9
21	population	y	c	abundant,clustered,nu	6	a,c,n,s,v,y	6
22	habitat	y	c	grasses,leaves,meadow	7	g,l,m,p,u,w,d	7

☐ Restrict Mode

Input Data File
File Name: agaricus-lepiota.data
Type: CSV
Objects: 8124
Attributes: 23

Bedrock File
File Name: agaricus-lepiota.bed
Created: 25/01/2010 12:19:16
Modified: 14/01/2010 15:00:50

Output Context File
File Name: mushHabiPop.cxt
Type: Burmeister
Extension: .cxt

Fig. 2. Mushroom habitat/population sub-context being created by FcaBedrock

This cxt file was then processed by ConExp to produce the concept lattice in Figure 3. In ConExp, the size of the node in the lattice can be made proportional to the number of own objects (mushrooms), so it can be seen that, for example, clusters of mushrooms are found in similar numbers in woods, leaves and waste ground, but not in other habitats. Solitary mushrooms are most likely to be found in woods, although they can occasionally be found in paths, urban areas and grassland.

A similar approach is applied by TocscanaJ [5], where an individual categorical attribute or a pair of continuous attributes are scaled to produce a lattice. A further attribute can be added as a nested lattice. The nested results are not always easy to interpret, however, sometimes requiring some visual cross-referencing of diagrams, and adding further lattice 'nests' does not appear possible or practical.

2.2 Object Exclusion

A second analysis was carried out, this time using the Adult data set [4]. This data set is US Census data of 32,561 adults, with attributes such as age, edu-

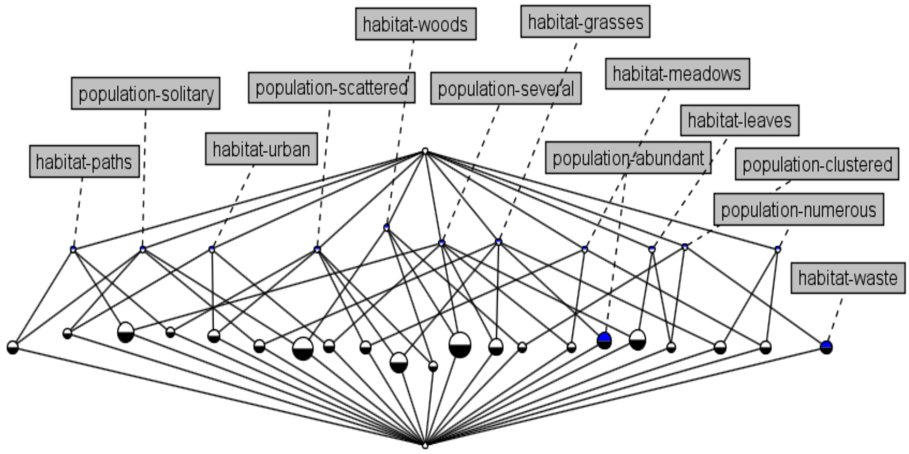
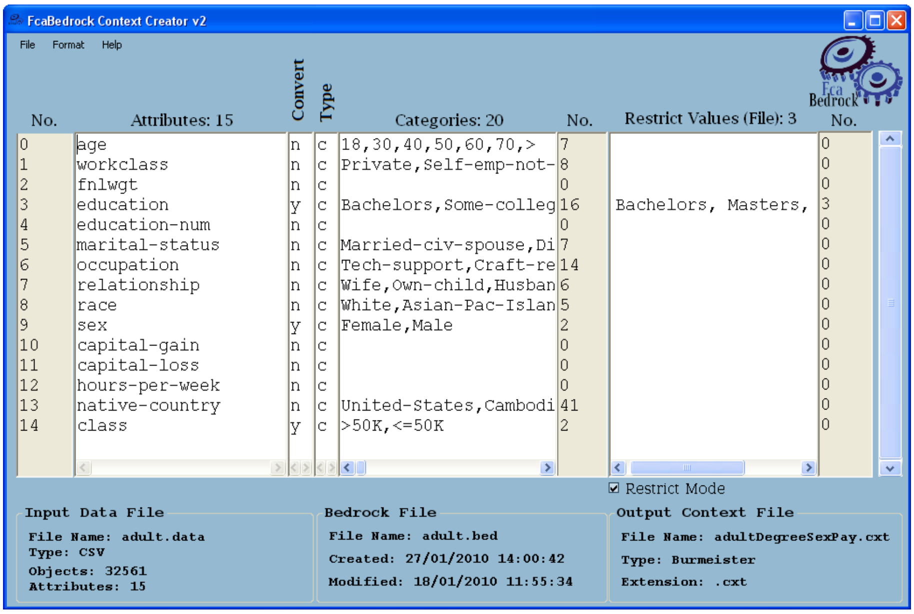


Fig. 3. Mushroom habitat/population lattice in ConExp

cation and employment type. The formal context produced when all the (suitable) attributes were converted contained over 100,000 concepts. However, let us say that in this analysis we are interested in comparing how pay is effected by gender in adults who have had a higher education. To carry out this analysis, FcaBedrock was used (Figure 4) to convert only the sex (*male*, *female*), class (pay <=*\$50k*, pay >*\$50k*) and education attributes. Furthermore, FcaBedrock's attribute value restriction feature was used to convert only those objects (adults) with the education attribute value *Bachelors*, *Masters* or *Doctorate* (three out of the 16 possible categories of education in the data set). This resulted in a sub-context with 7 formal attributes and 7,491 objects.

The resulting concept lattice in ConExp is shown in Figure 5, containing 37 concepts. The number of objects (and the percentage of the whole) is being displayed for the concepts of interest. Because only objects with Bachelors, Masters or Doctorate education categories were included in the conversion, there were 13 education categories (such as 10th grade and high school graduate) left with zero objects associated with them. Such unsupported formal attributes are normally labeled at the infimum of the concept lattice, but these labels can be hidden in ConExp.

Although a little cluttered with lines, the lattice is still readable, aided by a ConExp feature whereby information regarding a concept is displayed when a node is pointed to with the mouse. An examination of the concepts allows us to compare the percentage of males and females who earn more than \$50k, for each type of higher education. With a Bachelor's degree, 21% of females and 50% of males earned more than \$50k. This 'gender gap' was maintained at Master's level, with 33% of females and 65% of males earning more than \$50k,



No.	Attributes: 15	Convert	Type	Categories: 20	No.	Restrict Values (File): 3	No.
0	age	n	c	18,30,40,50,60,70,>	7		0
1	workclass	n	c	Private,Self-emp-not-	8		0
2	fnlwgt	n	c		0		0
3	education	y	c	Bachelors,Some-colleg	16	Bachelors, Masters,	3
4	education-num	n	c		0		0
5	marital-status	n	c	Married-civ-spouse,Di	7		0
6	occupation	n	c	Tech-support,Craft-re	14		0
7	relationship	n	c	Wife,Own-child,Husban	6		0
8	race	n	c	White,Asian-Pac-Islan	5		0
9	sex	y	c	Female,Male	2		0
10	capital-gain	n	c		0		0
11	capital-loss	n	c		0		0
12	hours-per-week	n	c		0		0
13	native-country	n	c	United-States,Cambodi	41		0
14	class	y	c	>50K,<=50K	2		0

☑ Restrict Mode

Input Data File
File Name: adult.data
Type: CSV
Objects: 32561
Attributes: 15

Bedrock File
File Name: adult.bed
Created: 27/01/2010 14:00:42
Modified: 18/01/2010 11:55:34

Output Context File
File Name: adultDegreeSexPay.cxt
Type: Burmeister
Extension: .cxt

Fig. 4. Adult Degree/sex/pay sub-context being created by FcaBedrock

but narrowed slightly at Doctorate level, with 58% of females and 78% of males earning more than $50k.

3 Reducing 'Noise' in a Context

The approaches described so far rely on reducing the size of the context. The next approach is to focus on the size of the concepts, using the well-known idea of minimum support [15] to filter out relatively small concepts (noise) from the data. This is achieved by specifying a minimum number of objects and/or attributes for a concept. Noise is therefore simply the concepts containing numbers of attributes or objects smaller than the user-defined minimums. This approach has been shown to be useful in gene-expression analysis [8] although the process described appeared to involve a degree of manual manipulation of the data and bespoke programming, and the analysis stopped short of visualising the results in a concept lattice.

In contrast, the approach described here is a semi-automated form of lattice 'iceberging' [13] to reduce noise using minimum support to such an extent that a manageable and meaningful concept lattice can be produced from the remaining concepts, thus giving a broad conceptual overview of the data. The reduction of noise is achieved by mining a context for concepts that satisfy a minimum support and then *re-writing* the context using only those concepts.

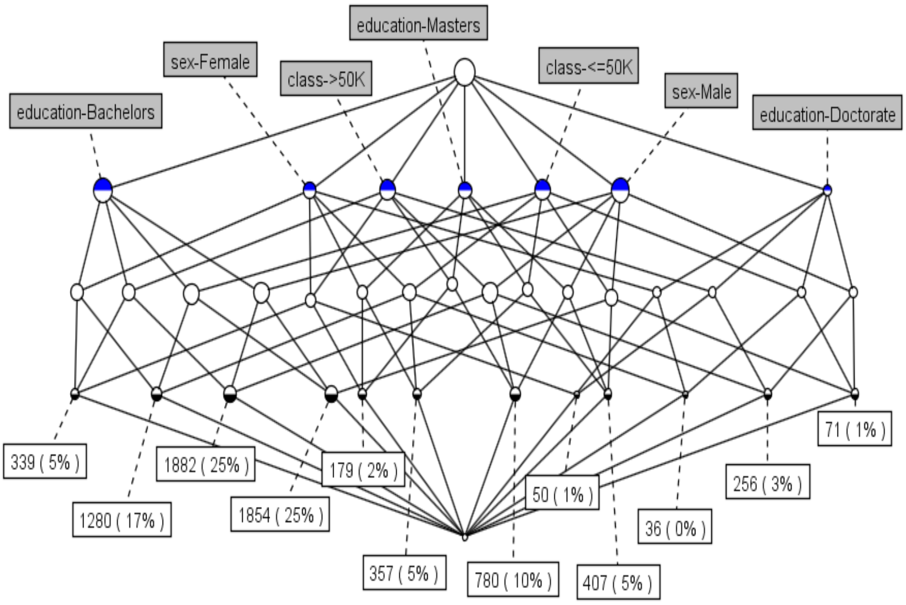


Fig. 5. Adult Degree/sex/pay lattice in ConExp

In-Close does this automatically. After mining concepts that satisfy a minimum support, In-Close uses them to output a ‘quiet’ (or ‘clean’) version of the original cxt file. This can then be used to produce a readable concept lattice. Figures 6 and 7 show a small example of applying a minimum support of two attributes and two objects. The two concepts large enough to satisfy the minimum support are ({a0,a1,a2},{o0,o1}) and ({a2,a3},{o1,o2}). These are mined and used to create the ‘quiet’ version. The lattices of the small ‘noisy’ and ‘quiet’ examples are shown in Figures 8 and 9 respectively.

In contrast to traditional iceberging, where a lattice is truncated by removing concepts that do not have a defined minimum number of objects, a complete hierarchy is maintained here in the resulting lattice, because where the large concepts ‘overlap’, other concepts are found during a *second pass* of concept mining (with no minimum support) when producing the concept lattice. In this way, possibly significant concepts that would not have satisfied the original minimum support are retained. In this simple example, the connecting concepts are ({a2},{o0,o1,o2}) and ({a0,a1,a2,a3},{o1}). These will be generated, along with the ones that satisfied the original minimum support. It should be noted that this means that it is the second, usually larger number of concepts that has to be noted when deciding on the level at which a manageable lattice is produced.

	a0	a1	a2	a3	a4	a5
o0	×	×	×			
o1	×	×	×	×		×
o2			×	×		
o3	×				×	

Fig. 6. A small 'noisy' context

	a0	a1	a2	a3	a4	a5
o0	×	×	×			
o1	×	×	×	×		
o2			×	×		
o2						

Fig. 7. A 'quiet' version of the 'noisy' context

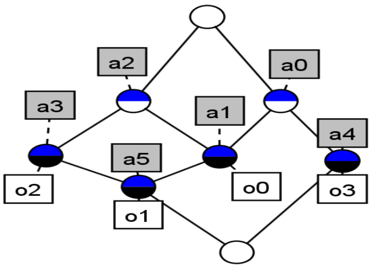


Fig. 8. Lattice in ConExp of the small 'noisy' context

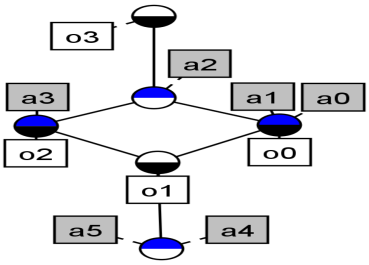


Fig. 9. Lattice in ConExp of the small 'quiet' context

3.1 A Student Survey Example

To illustrate this method, an analysis is carried out here on a set of student survey data [6] consisting of demographic and 'problem' data from 587 university undergraduates. The 'problem' data consists of 'yes/no' responses to 36 problems (or reasons for problems) that a student may have experienced during their studies; such as missing too many lectures, performing below their expectations, finding it difficult to settle or having high outside commitments. This is a good example of a noisy data set: there were 145 formal attributes when all the original attributes were converted by FcaBedrock, and the rather subjective 'yes/no' data gave rise to a context that was quite dense (20%) and noisy. Using In-Close, there were found to be 22,760,243 concepts.

However, let us say that we were only interested in analysing the 'problem' data. By only converting these attributes and excluding the demographic ones, a context containing 339,672 concepts was produced; a significant reduction but still far too many for a readable lattice. To reduce this number of concepts further, In-Close was used with minimum support specified for the intent and extent. The minimum size of intent was set to four. Although this may seem an arbitrary choice, it was decided that this would represent a sensible number if we were interested in combinations of problems that are experienced by students. The minimum support for the extent (number of students) was then varied to obtain large concepts that were small enough in number to obtain a readable lattice. Experimenting with In-Close, it appeared that a minimum support of between 80 and 70 students would produce a manageable lattice with between 32 and 160 nodes.

The lattice shown in Figure 10 was produced using the context generated by In-Close with a minimum support of 80 students (minimum of four attributes). In this case, the nodes in ConExp had a fixed radius to make them more prominent. The attributes related to the following problems asked in the student survey:

- course: Have sometimes found course stressful
- stress: Stress problems
- leave: Considered changing/leaving at some stage
- different: Course different from expectation
- examinations: Examination performance below expectations
- commitments: Outside commitments high
- distractions: Too many distractions that affect ability to study

The lattice appeared to give some useful insight into commonly connected problems experienced by undergraduates. It seemed that a stressful course was a common problem (217 students) in combination with general stress problems, considering leaving a course, finding a course different to expectations and not performing as expected in examinations. Problems that were less common, but still related, seemed to be having high outside commitments and a large number of distractions; all students who reported these problems found their course stressful.

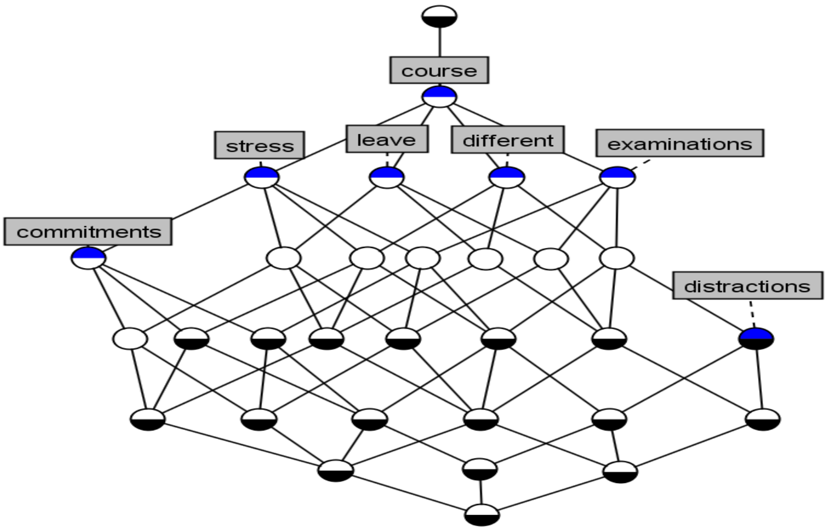


Fig. 10. Student problem lattice in ConExp

3.2 Comparing Quiet Sub-Contexts

To further illustrate the usefulness of this method, a second analysis of the Mushroom data set was carried out, this time to investigate differences between poisonous and edible mushrooms. The attribute value restriction of FcaBedrock was used to divide the mushroom context into an edible mushroom sub-context and a poisonous mushroom sub-context. There were 4,208 objects (mushrooms) and 92,543 concepts in the edible mushroom sub-context and 3,916 objects and 86,198 concepts in the poisonous mushroom sub-context. To provide a significant number of attributes to compare, 10 was chosen as the minimum support for intent. The process of reducing noise using minimum support by In-Close was carried out, resulting in an edible mushroom concept lattice containing 2,848 objects and 17 concepts, and a poisonous mushroom concept lattice containing 3,344 objects and 14 concepts (Figures 11 and 12).

Similarities were identified as attributes expressed in both lattices and were moved to the right of each lattice. Differences were identified as attributes expressed in only one lattice and were moved to the left, thus giving a clear visualisation for comparison. Examining the lattices suggest combinations of features that indicate if a mushroom is safe to eat. For example, it seems that smooth-stalked mushrooms are likely to be safer to eat than silky-stalked ones. The lattices also suggest that mushrooms with pendant rings are more wholesome than those with evanescent rings. The combination of a foul smell, bulbous root and a chocolate coloured spore print would seem to be a strong indication of danger. The fact that bruised mushrooms are likely to be edible could be be-

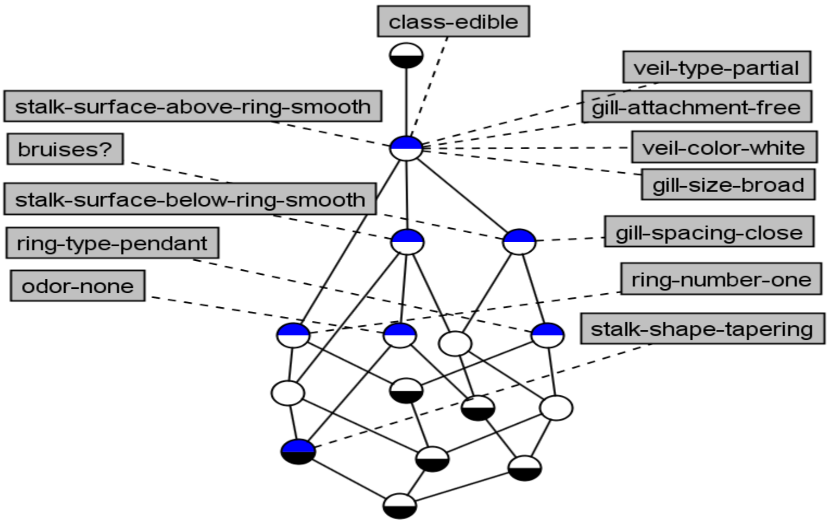


Fig. 11. Edible mushroom lattice in ConExp

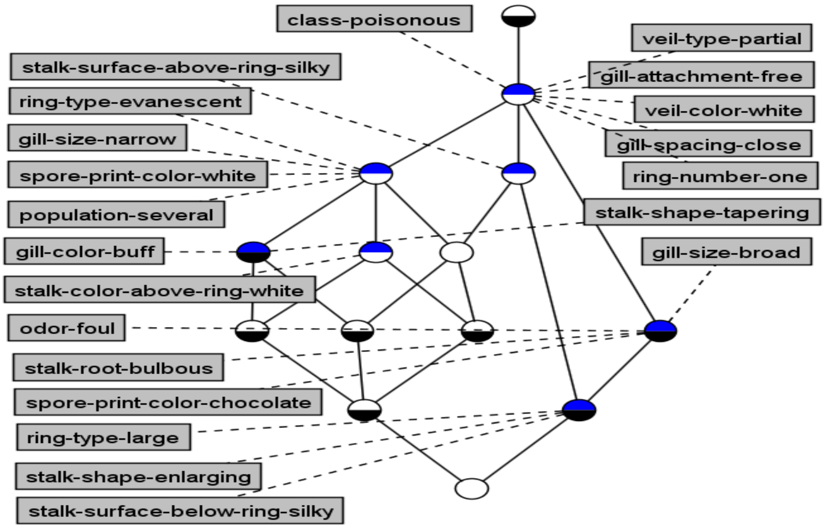


Fig. 12. Poisonous mushroom lattice in ConExp

cause foraging animals may damage mushrooms that are near to ones they are eating.

4 Conclusion

Although large data sets may be difficult to deal with computationally, it is the number of formal concepts derived from a data set that is the key factor in determining if a concept lattice will be useful as a visualisation. Typical data sets contain too many concepts. It has been shown here that readable lattices can be produced from real data sets with a straightforward process of creating subcontexts and reducing noise, using freely available software. Whilst the analyses presented here have not been rigorously corroborated with domain expertise or statistical analysis of the data, we have nonetheless demonstrated that understandable results are obtainable from existing data sources where this would not normally have been the case. Formal contexts that would normally be intractable for visualisation have been processed in a disciplined manner to provide meaningful results. This has been achieved by 1) focusing on information of interest and by 2) reducing 'noise' in the context, thus revealing readable lattices that lucidly portray conceptual meaning in large data sets.

The cxt format, commonly used in FCA, was used successfully here to interoperate between the tools. Further integration of the tools would make the analysis easier to perform. This would also give scope for improvements such as discarding the attributes that are left with no objects after the second pass of concept mining, rather than manually hiding these labels in ConExp. Integration will also open up the possibility of more dynamic analysis, where a user can quickly see changes in the lattice when altering the parameters of analysis. Some of the operations, such as determining the level of minimum support, would particularly benefit from this greater degree of automation.

There may be limitations in the technique of noise reduction presented here; for example, if two otherwise identical attributes that have high support differ only in one object (possibly even due to error of data entry), concepts for the first and second attribute will be included in the resulting lattice. Such anomalies might be removed by applying the notion of concept stability, where the stability of a concept is defined by the extent to which its attributes are dependent on its objects (a stable concept is not affected by attribute 'noise' in object descriptions) [10] or by applying so-called fault-tolerant FCA, in allowing a bound number of exceptions (false values) in a concept [11]. A comparison of techniques of noise reduction and concept clustering would be useful.

Acknowledgement This work is part of the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"), funded by the European Commission's 7th Framework Programme of ICT, under topic 4.3: Intelligent Information Management.

References

1. Andrews, S.: In-Close, A Fast Algorithm for Computing Formal Concepts. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS'09, `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-483/` (2009)
2. Andrews, S.: Data Conversion and Interoperability for FCA. In: CS-TIW 2009, pp. 42-49, `http://www.kde.cs.uni-kassel.de/ws/cs-tiw2009/proceedings_final_15July.pdf` (2009)
3. Andrews, S., Orphanides, C.: *FcaBedrock, a Formal Context Creator*. In: Croitoru, M., Ferre, S. and Lukose, D. (eds.) ICCS 2010, LNAI 6208. Springer-Verlag, Berlin/Heidelberg (2010)
4. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html`. Irvine, CA: University of California, School of Information and Computer Science (2007)
5. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., et al. (eds.) Formal Concept Analysis, LNCS (LNAI), vol. 3626, pp. 324-348. Springer-Verlag, Berlin-Heidelberg (2005)
6. Burley, K.: Data Mining Techniques in Higher Education Research: The Example of Student Retention. A thesis submitted in partial fulfillment of the requirements of Sheffield Hallam University for the degree of Doctor of Education. Sheffield Hallam University (2006)
7. Ganter, B., Wille, R.: Conceptual Scaling. In: Roberts, F. (ed.) Applications of Combinatorics and Graph Theory to the Biological and Social Sciences. IMA, vol. 17, pp. 139-168, Springer, Berlin-Heidelberg-New York (1989)
8. Kaytoue-Uberall, M., Duplesssis, S., Napoli, A.: Using Formal Concept Analysis for the Extraction of Groups of Co-expressed Genes. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) MCO 2008. CCIS vol. 14, pp. 439-449. Springer-Verlag, Berlin/Heidelberg (2008)
9. Krajca, P., Outrata, J., Vychodil, V.: Parallel Recursive Algorithm for FCA. In: Belohlavek, R., Kuznetsov, S.O. (eds.) CLA 2008, pp. 71-82. Palacky University, Olomouc (2008)
10. Kuznetsov, S., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007, LNAI 4604, pp. 241-254. Springer-Verlag, Berlin/Heidelberg (2007)
11. Pensa, R. G., Boulicaut, J-F.: Towards Fault-Tolerant Formal Concept Analysis. In: Banidini, S., Manzoni, S. (eds.) AI*IA 2005, LNAI 3673, pp. 212-223, Springer-Verlag, Berlin/Heidelberg (2005)
12. Priss, U.: FcaStone - FCA File Format and Interoperability Software. In: Croitoru, M., Jaschkë, R., Rudolph, S. (eds.) CS-TIW 2008, pp. 33-43 (2008)
13. Stumme, G., Taouil, R., Bastide, Y., Lakhal, L., Conceptual clustering with iceberg concept lattices. In: Proceedings of GIFachgruppentreffen Maschinelles Lernen01, Universitat Dortmund, vol. 763. (2001)
14. Yevtushenko, S.A.: System of data analysis "Concept Explorer". (In Russian). Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia, 2000.
15. Zaki, M. J., Hsiao, C-J.: Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. In: IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pp. 462-478. IEE Computer Society (2005)