Data conversion and interoperability for FCA

ANDREWS, Simon J <http://orcid.org/0000-0003-2094-7456>

Published version

ANDREWS, Simon J (2009). Data conversion and interoperability for FCA. In: Conceptual Structures Tools Interoperability Workshop at the 17th International Conference on Conceptual Structures, Moscow, 2009.

Data Conversion and Interoperability for FCA

Simon Andrews

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

Abstract. This paper proposes a tool that converts non-FCA format data files into an FCA format, thereby making a wide range of public data sets and data produced by non-FCA tools interoperable with FCA tools. This will also offer the power of FCA to a wider community of data analysts. A repository of converted data is also proposed, as a consistent resource of public data for analysis and for the testing, evaluation and comparison of FCA tools and algorithms.

1 Introduction

Most tools for Formal Concept Analysis require data to be in a particular format, usually representing a formal context and/or concept lattice. Unfortunately, most publicly available data sets, and most of the data produced by non-FCA applications, is not in this format. To make them interoperable with FCA tools, they need to be converted. Furthermore, there is a variety of data set formats and data types, each requiring different treatment. Converting data sets for FCA can be a time consuming and awkward task. A further problem arises in that a particular data set may be interpreted in different ways, resulting in inconsistent conversions. This can lead to different analyses of the same data and can make the comparison of FCA tools and algorithms more difficult. These problems have been pointed out by this author [2] and by Kuznetsov and Ob"edkov [5]:

> "We would like to propose the community to reach a consensus w.r.t. databases to be used as testbeds. Our idea is to consider two types of testbeds. On the one hand, some "classical" (well-recognised in data analysis community) databases should be used, with clearly defined scalings if they are many-valued. On the other hand, we propose to use "randomly generated contexts"... The community should specify particular type(s) of random context generator(s) that can be tuned by the choice of ... parameters."

This paper, therefore, has two proposals to improve the interoperability and consistency of use of non-FCA format data sets:

1. A 'To-FCA Format' Data Converter, that will provide FCA practitioners with an efficient and consistent means of converting a range of non-FCA data set formats into a format suitable for FCA.
2. An FCA Data Repository, that will provide FCA practitioners with a resource of public data sets in FCA format.

2 Data Conversion

Public data set repositories, such as the UCI Machine Learning Repository [3] and Amazon Public Data Sets [1], provide a useful resource for FCA. Several data sets from the UCI Repository have become familiar to FCA. Four of these are listed in Table 1 and are useful to illustrate issues in data conversion. The table lists the name of the data set, its format, number of objects, number of attributes, the data type/s of the attributes and the number of attributes once converted into a formal context. Some of these have a question mark indicating that there are different interpretations possible. For example, the Mushroom data set has been quoted variously as having 125 [2], 119 [4] and 120 [9] attributes.

Table 1. Some UCI Repository Data Sets

Name	Format	Objs	Atts	Att Type	FCA Atts
Mushroom	data only	8124	22	Categorical	125?
Adult	data only	48842	14	Categorical, Integer	96?
MS Web	DST Sparse	32711	294	N\A	294
Internet Ads	data only	3279	1558	Real, Relational	1555?

There are several issues to consider when converting data sets into a formal context:

Data Set Format Different 'standard' data set formats require different treatment. Some contain data values only, others include additional information, such as the names of the attributes or the classes of objects. Many contain data in tabular form, with rows representing objects and columns representing attributes. The type of the attribute (categorical, integer, real or relational) will determine the conversion technique required. Many data sets are multivariate, having a mixture of attribute types. Some data sets do not have a tabular format, such as the DST Sparse Data Format and Sparse ARFF, where relational data is represented by attribute/object pairs; clearly suitable for FCA, but requiring a different method of conversion. Similarly, RDF files [6] would require a different method of conversion, where an RDF subject becomes an FCA object. Databases are another important consideration; RDBMS data files would require a significantly different approach to the treatment of flat-file data.

Categorical Attributes Categorical (many valued) attributes are the most common type and can be converted by creating a formal context attribute for each of the values. The attribute *cap-surface*, for example, in the Mushroom data set, has four values: *fibrous*, *grooves*, *scaly* and *smooth.* In a formal context, this becomes four attributes: *cap-surface_fibrous*, *cap-surface_grooves*, *cap-*

surface_scaly and *cap-surface_smooth*. However, different interpretations are possible if an attribute has only two values; it may be said that not having one value implies having the other, thus leading to a single attribute in the formal context. This is particularly so if the values are opposites. For example, in the Mushroom data set, there is an attribute called *bruises?* that has values *bruises* and *no*. Should this be interpreted as a single attribute, *bruises*, or two: *bruises?_bruises* and *bruises_no*? If, for a particular attribute, there is no object with a particular value, or all objects have the same value, how should this be interpreted? In the Mushroom data set, for example, none of the mushrooms has a universal veil; all have a partial veil. Should the veil attribute be ignored or interpreted as one or two attributes?

Other Attribute Types A table of Boolean values is used by some data sets to indicate the relationship between attributes and objects, such as the majority of the data in the Internet Ads data set. Such data can be translated, one-to-one, into a corresponding context. Attributes with integer or real types are less easily dealt with. Should they be ignored or should some form of scaling be used to create context attributes with ranges of values? Attributes that have free text values (a person's address, for example) are the least convertible and will almost certainly be omitted in a conversion.

Missing Values Many data sets contain missing values. Should they be treated as a 'has not' relationship or should objects with missing values be removed?

Classes Some data sets contain classes of data. Should these be ignored in the conversion, treated as attributes, or should a separate context be created for each class?

Clearly, a useful tool for data conversion must deal with all of these issues.

2.1 A Conversion Example

A simple example will help illustrate some of the issues outlined above, and visualise possible input and output files of an FCA data converter. Figure 1 is a miniature version of the UCI Mushroom data file, `mushroom.data`; a data only flat-file of comma separated values, and a format very suitable for input to an FCA data converter. The first column gives the mushroom class, the other four are mushroom attributes. Each is nominally valued in the following way:

column 1 class: edible = e, poisonous = p
column 2 bruises?: bruises = t, no = f
column 3 gill-size: broad = b, narrow = n (missing value = ?)
column 4 veil-type: partial = p, universal = u
column 5 ring-number: none = n, one = o, two = t

```
e,t,b,p,n
e,t,n,p,t
p,f,n,p,n
e,t,?,p,o
p,f,n,p,n
```

Fig. 1. Miniature version of `mushroom.data`

Figure 2 is an interpretation of the mushroom data as a formal context. The following decisions have been made in the interpretation:

- The mushroom class is not used.
- The attribute ‘bruises?’ is a single formal attribute.
- All other attributes have a formal attribute for each of their possible values.

Mushroom	bruises	gill-size-broad	gill-size-narrow	veil-type-partial	veil-type-universal	ring-number-none	ring-number-one	ring-number-two
mushroom1	×	×		×		×		
mushroom2	×		×	×				×
mushroom3			×	×		×		
mushroom4	×			×			×	
mushroom5			×	×		×		

Fig. 2. Mushroom context

Figure 3 is the Mushroom context in the Burmeister cxt file format, a common FCA context format used by a number of FCA tools and one that could be output from an FCA data converter. The ‘B’ at the start of the file possibly stands for ‘Burmeister’; it appears to be a tradition of the format! The numbers that follow are the number of objects and the number of attributes, respectively. The names of the objects and attributes are then listed, followed by the incidence vectors of the objects, consisting of dots and ‘X’s.

How the object and attribute names are obtained will be a factor in the design of an FCA data converter, and how far this process can be automated will need to be considered.

```
B

5
8

mushroom1
mushroom2
mushroom3
mushroom4
mushroom5
bruises
gill-size-broad
gill-size-narrow
veil-type-partial
veil-type-universal
ring-number-none
ring-number-one
ring-number-two
XX.X.X..
X.XX...X
..XX.X..
X..X..X.
..XX.X..
```

Fig. 3. `mushroom.cxt`

3 Proposal 1: A ‘To-FCA Format’ Data Converter

Figure 4 is a simple, high-level view of the proposed ‘To-FCA Format’ data converter. The proposed output of the converter is a file in the Burmeister cxt format.

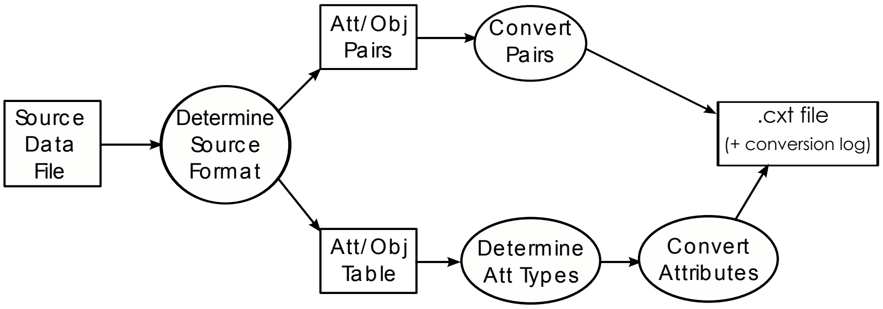

Fig. 4. Proposed ‘To-FCA’ Data Converter

After selection of the source data file format, the main process is determined by whether the data is in the form of attribute/object pairs or in the form of a table. In the latter case, the tool must carry out the appropriate conversion depending on the type of each attribute. Information will be required by the tool, concerning the number and names of objects and attributes, the type of each attribute and its categories, if appropriate. This information may be obtained by the tool from the source data file, or from the user, depending on the source format, and will be output in a conversion log, along with information regarding decisions made in the conversion process, such as any original attributes not converted.

There are a number of FCA context formats, other than cxt, used by a variety of FCA tools and applications. The data converter could be expanded to output these formats, too, but FcaStone[1] [7, 8], already exists that easily converts one commonly used FCA file type into another. The proposed data converter could integrate with FcaStone, making non-FCA format data interoperable with a much wider range of FCA tools and applications (Figure 5). In conjunction with Graphviz[2], an open-source graph visualisation tool, FcaStone can produce a range of graph formats for the production of concept lattices, thus expanding further the range of FCA tools interoperable with the original data.

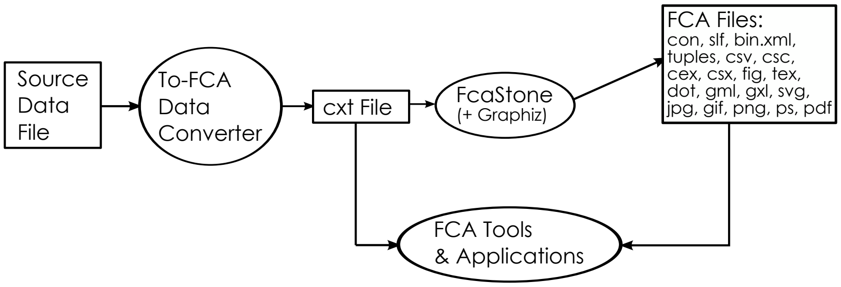
Source Data File
To-FCA Data Converter
cxt File
FcaStone
(+ Graphiz)
FCA Files:
con, slf, bin.xml, tuples, csv, csc, cex, csx, fig, tex, dot, gml, gxl, svg, jpg, gif, png, ps, pdf
FCA Tools & Applications

Fig. 5. Possible Integration of 'To-FCA' Data Converter and FcaStone

4 Proposal 2: An FCA Data Repository

Figure 6 is a diagram of the proposed, web-based, FCA data repository. It is proposed that the data converter tool and FcaStone are incorporated into the repository allowing users to convert data sets as well as to access the stored converted data sets. Users will also be able to donate data sets. A converted data set will be stored along with, where possible, its original data file, information about the original data (probably from the original data source), a link back to the original data source, the conversion log, and FCA information, such as context density and number of concepts.

To address needs outlined in the introduction of this paper, the repository will also provide access to stored random data and incorporate a random data

[1] FcaStone: http://fcastone.sourceforge.net

[2] Graphviz: http://www.graphviz.org

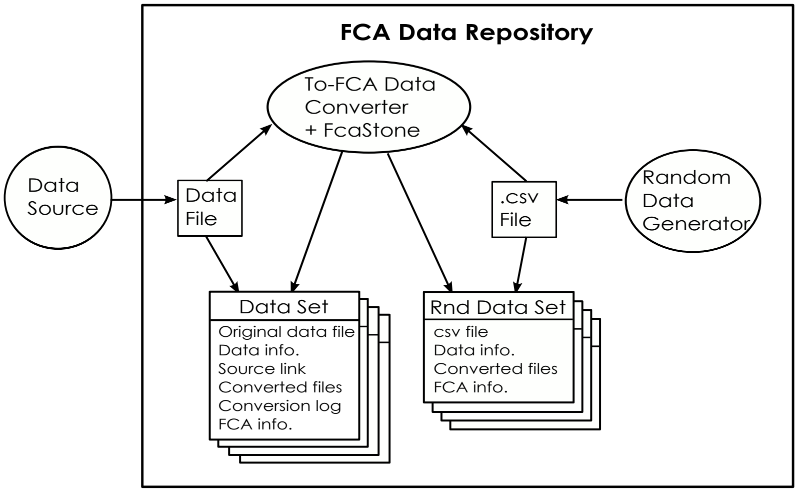

Fig. 6. Proposed FCA Data Repository

generator, the initial output of which will be a file of comma-separated object number, attribute number pairs. The csv file can then be converted into the required format. The user will be able to determine the number of attributes, number of objects and the density. It is proposed that the user will also select one of a small range of random number generator seeds; thus the same data set will be generated by any given seed/data parameters.

In this way, the repository can act as a bench-marker for the comparison of tools and algorithms by providing citeable random data as well as converted 'real' data sets.

5 Development of the Proposals and Conclusion

The To-FCA data converter will be developed as an open-source software. An initial prototype is under development, converting 'vanilla' data sets (such as the UCI Mushroom data set) and should be ready as a demonstrator tool in June/July 2009. A Link has been formed with the JISC Information Environment Demonstrator Project[3] to provide possible dissemination vehicles for the development. It is also hoped that participation in an ICCS 2009 workshop comparing the performance of FCA algorithms[4] will provide useful steering regarding data formats.

The UCI Machine Learning Repository has kindly given the author permission to reformat and make their data sets available online. The FCA data

[3] JISC IE Demonstrator Project: http://www.jisc.ac.uk/whatwedo/programmes/reppres/iedemonstrator.aspx

[4] Comparing performance of FCA algorithms: http://iccs09.org/forum

repository is likely to initially take the form of a web service offering a small selection of converted UCI data sets. The addition of random data sets and the incorporation of tools will be an incremental development. If data sets are to be encouraged from donors, a repository librarian will be required to validate them, ensuring that they are in required format and that the necessary (and verifiable) supporting information is provided.

The final vision is of an interactive set of web-services, offering FCA tools interoperability with a wide range of data, providing a resource of useful collections of 'real' and random data sets in a wide variety of FCA context and lattice formats, and offering users the facility to create, convert and donate data sets of their own. Conversion between non-FCA and FCA formats will also open a way to the wider use of FCA, offering its power to those currently outside the FCA community.

References

1. Amazon Web Services: Public Data Sets on AWS [http://aws.amazon.com/publicdatasets/] (2009)
2. Andrews, S.: In-Close, a Fast Algorithm for Computing Formal Concepts. To be presented at the Seventeenth International Conference on Conceptual Structures (2009).
3. Asuncion, A., Newman, D. J.: UCI Machine Learning Repository [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science (2007).
4. Krajca, P., Outrata, J., Vychodil, V.: Parallel Recursive Algorithm for FCA. In: Belohlavek, R., Kuznetsov, S.O. (eds.), *Proceeding of the Sixth International Conference on Concept Lattices and their Applications*, pp. 71-82, Palacky University, Olomouc (2008).
5. Kuznetsov, S.O., Ob"edkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. In: *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 14, pp. 189-216 (2002).
6. Passin, T. B.: Explorer's Guide to the Semantic Web. Manning Publications Co., Greenwich, CT 06830, USA (2004).
7. Priss, U.: FcaStone - FCA File Format and Interoperability Software. In: Croitoru, M., Jaschkë, R., Rudolph, S. (eds.), Conceptual Structures and the Web, *Proceedings of the Third Conceptual Structures and Tool Interoperability Workshop*, pp. 33-43 (2008).
8. Priss, U: FCA Software Interoperability, In: Belohlavek, R., Kuznetsov, S. O. (eds.) *Proceeding of the Sixth International Conference on Concept Lattices and Their Applications*, pp. 133-144 (2008).
9. Wang, J., Han, J., and Pei, J.: CLOSET+: searching for the best strategies for mining frequent closed itemsets. *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pp. 236-245, ACM, New York (2003).