Chapter XVIII

Relational Data, Formal Concept Analysis, and Graded Attributes

Radim Belohlavek
Binghamton University – SUNY, USA and Palacky University, Czech Republic

ABSTRACT

Formal concept analysis is a particular method of analysis of relational data. Also, formal concept analysis provides elaborate mathematical foundations for relational data. In the course of the last decade, several attempts appeared to extend formal concept analysis to data with graded (fuzzy) attributes. Among these attempts, an approach based on residuated implications plays an important role. This chapter presents an overview of foundations of formal concept analysis of data with graded attributes, with focus on the approach based on residuated implications and on its extensions and particular cases. Presented is an overview of both of the main parts of formal concept analysis, namely, concept lattices and attribute implications, and an overview of the underlying foundations and related methods. In addition to that, the chapter contains an overview of topics for future research.

INTRODUCTION

Tabular Data, Formal Concept Analysis, and Related Methods

Tables, that is, two-dimensional arrays, represent perhaps the most popular way to describe data. Tables, rows correspond to objects of our interest, table columns correspond to some of their attributes, and table entries contain values of attributes on the respective objects. As an example, consider patients as objects and the patients’ names, weight, gender, and so forth as attributes. Table rows and columns are usually labeled by objects’ and attributes’ names. A particular case arises when all the attributes are logical attributes (presence or absence attributes) like male, headache, left-handed, and so forth. A patient either is a male or not, and, in general, either has a logical attribute or not. In this case, a table entry corresponding to object \( x \) and attribute \( y \) contains \( \times \) or is blank depending on whether object \( x \) has or does not have attribute \( y \).
Many methods of various kinds have been and are being developed for representation, processing, and analysis of tabular data. This chapter is concerned with formal concept analysis (FCA), which is a particular method for knowledge extraction from tabular data. Although some previous attempts exist (see Barbut, 1965), FCA was initiated by Wille’s (1982) seminal paper. Since then, significant progress has been made in theoretical foundations, algorithms, and methods. Applications of FCA can be found in many areas of human affairs, including engineering, sciences, economics, information processing, mathematics, psychology, and education; see, for example, Carpineto and Romano (2004b) and Koster (2006) for applications in information retrieval; Snelting and Tip (2000) for applications in object-oriented design; Ganapathy, King, Jaeger, and Jha (2007) for applications in security; Pfaltz (2006) for applications in software engineering; Zaki (2004) for how concept lattices can be used to mine nonredundant association rules; and Ganter and Wille (1999) and Carpineto and Romano (2004a) for further applications. Two monographs on FCA are available: Ganter and Wille (1999, mainly mathematical foundations) and Carpineto and Romano (2004a; mainly algorithms and applications). There are three international conferences devoted to FCA, namely, ICFCA (International Conference on Formal Concept Analysis), CLA (Concept Lattices and their Applications), and ICCS (International Conference on Conceptual Structures). In addition, further papers on FCA can be found in journals and proceedings of other conferences.

A table with logical attributes can be represented by a triplet \( \langle X, Y, I \rangle \) where \( I \) is a binary relation between \( X \) and \( Y \). Elements of \( X \) are called objects and correspond to table rows, elements of \( Y \) are called attributes and correspond to table columns, and for \( x \in X \) and \( y \in Y \), \( \langle x, y \rangle \in I \) indicates that object \( x \) has attribute \( y \) while \( \langle x, y \rangle \notin I \) indicates that \( x \) does not have \( y \). For instance, Figure 1 (left) depicts a table with logical attributes.

The corresponding triplet \( \langle X, Y, I \rangle \) is given by \( X = \{x_1, x_2, x_3, \ldots \} \), \( Y = \{y_1, y_2, y_3, \ldots \} \), and we have \( \langle x_1, y_1 \rangle \in I \), \( \langle x_2, y_1 \rangle \notin I \), and so forth. Since representing tables with logical attributes by triplets is common in FCA, we say “table \( \langle X, Y, I \rangle \)” instead of “triplet \( \langle X, Y, I \rangle \) representing a given table.” FCA aims at obtaining two outputs out of a given table. The first one, called a concept lattice, is a partially ordered collection of particular clusters of objects and attributes. The second one consists of formulas, called attribute implications (AIs), describing particular attribute dependencies that are true in the table. The clusters, called formal concepts, are pairs \( \langle A, B \rangle \) where \( A \subseteq X \) is a set of objects and \( B \subseteq Y \) is a set of attributes such that \( A \) is a set of all objects that have all attributes from \( B \), and \( B \) is the set of all attributes that are common to all objects from \( A \). For instance, \( \langle \{x_1, x_2\}, \{y_1, y_2\} \rangle \) and \( \langle \{x_1, x_2, x_3\}, \{y_1\} \rangle \) are examples of formal concepts of the (visible part of) the left table in Figure 1. An attribute implication is an expression \( A \Rightarrow B \) with \( A \) and \( B \) being sets of attributes. \( A \Rightarrow B \) is true in table \( \langle X, Y, I \rangle \) if each object having all attributes from \( A \) has all attributes from \( B \) as well. For instance, \( \{y_1\} \Rightarrow \{y_2\} \) is true in the (visible part of) the left table in Figure 1, while \( \{y_1\} \Rightarrow \{y_3\} \) is not (\( x_2 \) serves as a counterexample).

**Graded Attributes and Extensions of Formal Concept Analysis**

Contrary to classical (two-valued) logic, fuzzy logic uses intermediate truth degrees in addition to 0 (false) and 1 (true). Fuzzy logic thus allows us to assign truth degrees like 0.8 to propositions like “Customer C is satisfied with service s.” In this example, assigning 0.8 to the above proposi-
Related Content

Process-Embedded Data Integrity
[www.igi-global.com/article/process-embedded-data-integrity/3307?camid=4v1a](www.igi-global.com/article/process-embedded-data-integrity/3307?camid=4v1a)

Object-Oriented System Design
[www.igi-global.com/article/object-oriented-system-design/51111?camid=4v1a](www.igi-global.com/article/object-oriented-system-design/51111?camid=4v1a)

Interactive Indexing of Documents with a Multilingual Thesaurus
[www.igi-global.com/chapter/interactive-indexing-documents-multilingual-thesaurus/9203?camid=4v1a](www.igi-global.com/chapter/interactive-indexing-documents-multilingual-thesaurus/9203?camid=4v1a)

Aiding Maintenance of Database Applications Through Extracting Attribute Dependency Graph
[www.igi-global.com/article/aiding-maintenance-of-database-applications-through-extracting-attribute-dependency-graph/84067?camid=4v1a](www.igi-global.com/article/aiding-maintenance-of-database-applications-through-extracting-attribute-dependency-graph/84067?camid=4v1a)