Formal Concept Analysis as a Framework for Business Intelligence Technologies II

Juraj Macko

Division of Applied computer science
Dept. Computer Science
Palacky University, Olomouc
17. listopadu 12, CZ-77146 Olomouc
Czech Republic
email: {juraj.macko}@upol.cz

Abstract. Formal concept analysis (FCA) with measures can be seen as a framework for Business Intelligence technologies. In this paper we introduce new ideas about an OLAP cube. We take a focus on a high-dimensional OLAP cube reduction and on a hierarchy of attributes in an OLAP cube.

1 Introduction

This paper continues with results proposed in [9] and for more details we will refer on it. The paper is structured as follows: In "Preliminaries" the fundamentals of FCA with measures are described, the formal definition of the *OLAP cube* is shown. "Compressing High-Dimensional OLAP Cube Using FCA With Measures" shows an efficient reduction of an OLAP space using FCA with measures. In "Attribute Hierarchy In OLAP And In FCA With Measures" we discuss different types of hierarchies in OLAP. This paper is supplemented with comprehensive examples. The final part summarizes the results.

2 Preliminaries

An input dataset for FCA is a formal context, which is a relation between the set of objects X and the set of attributes Y, is denoted by $\langle X, Y, I\rangle$ where $I \subseteq X \times Y$. The concept forming operators $()^{\uparrow}$ and $()^{\downarrow}$ are defined as $A^{\uparrow} = \{y \in Y \mid \text{ for each } x \in X : \langle x, y\rangle \in I\}$ and $B^{\downarrow} = \{x \in X \mid \text{for each } y \in Y : \langle x, y\rangle \in I\}$. A formal concept of the formal context $\langle X, Y, I\rangle$ is denoted by $\langle A, B\rangle$, where $A \subseteq X$ and $B \subseteq Y$. $\langle A, B\rangle$ is a formal concept iff $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The set A is called an extent and the set B an intent. A set of all formal concepts of $\langle X, Y, I\rangle$ is denoted by $\mathcal{B}(X, Y, I)$ and equipped with a partial order $\leq$ forms a concept lattice of $\langle X, Y, I\rangle$.

Definition 1 (Measure of Object and Attribute [9]). *A Measure of the object is mapping* $\Phi : X \to \mathbb{R}^+$ *and a Measure of the attribute is mapping* $\Psi : Y \to \mathbb{R}^+$.

Definition 2 (Value of Extent and Intent [9]). *The Value of extent is mapping* $v : A_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $v(A) = \underset{x \in A}{\odot} \Phi(x)$, *where* $\odot$ *is either the symbol for the sum* Σ *(the "sum" operation) or the symbol for cardinality* $|A|$ *or the arbitrary aggregation function* Θ. A *is an extent of the formal concept* $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$. *Similarly, the value of the intent is mapping* $w : B_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $w(B) = \underset{y \in B}{\odot} \Psi(y)$, *where* B *is an intent of the formal concept* $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$

The Database table is a relation r on the relation scheme $R = \{A_1, A_2 \ldots, A_n\}$ defined as a set of mappings $\{t_1, t_2 \ldots, t_m\}$ from R to $\mathcal{D}$ where $\mathcal{D}$ is a set of all D - domains of attributes A, n is the number of the columns and m the number of rows in a database table (see [4]). Domains in $\mathcal{D}$ are divided into the two groups: $H_k \in \mathcal{H}$- dimensions and $M_s \in \mathcal{M}$ - measures, where $k \in [1; |\mathcal{H}|]$, $s \in [1; |\mathcal{M}|]$ and $M_s \subseteq \mathbb{R}^+$).

Definition 3 (OLAP Cube space, OLAP Cube [9]). *The space for the OLAP cube is a cartesian product* $C = L^{H_1} \times \cdots \times L^{H_k} \times \cdots \times L^{H_{|\mathcal{H}|}}$, *where* $L = \{0, 1\}$. *The OLAP cube is a mapping* $\sigma : C \to \mathbb{R}^+$ *and is defined as* $\sigma(h_1, \ldots, h_n) = \overset{m}{\underset{i=1}{\odot}} t_i(M_s)$ *such that* $\{t_i(A_j)\} \supseteq h_j$ *for all* $j \in [1; |\mathcal{H}|]$, *where the symbol* $\odot$ *stands for the sum operator* Σ, *the cardinality operator* $||$ *or the arbitrary aggregation operator* Θ *and* $|\mathcal{H}|$ *is the number of OLAP cube dimensions.*

3 Compressing High-Dimensional OLAP Cube Using FCA With Measures

In the previous paper [9] we have shown, that FCA with measures can be seen as a generalized OLAP. OLAP uses data which are organized in dimensions. As a direct consequence is, that the scaled attributes (using a nominal scale [1]) from one domain are mutually exclusive. FCA with values enables to analyze the data which are not organized in dimensions, thus those which are independent (see the example with cars and components taken from [9] shown in Table 4). This fact means, that we can work with a relational (binary) data as well. When the attributes with a binary domain are used, usually there is a relatively big amount of such attributes. It implies a high-dimensional OLAP cube. Recall from [9], that the size of an OLAP cube is $(|H_1| + 1) \times \cdots \times (|H_{|\mathcal{H}|}| + 1)$. The expression "+1" means, that using the domain $H_1 = \{BMW, SKODA, FIAT\}$ we consider such situation, when no attribute is selected. In a binary case we have two possibilities only (an attribute is selected or not), so a space of such cube will be $2^{|\mathcal{H}|}$, where $|\mathcal{H}|$ is a number of domains (all domains are binary in this case). Hence, the space of the OLAP cube is exponential wrt. number

of attributes. FCA with measures enables to compress such exponential space. Consider Y as a set of attributes in FCA. Number of formal concepts (which contains intents, closed sets of attributes) is usually significantly lower than a powerset 2^Y, because a real dataset is usually sparse. Using FCA with measures we can replace OLAP cube with a concept lattice with values and we do not loose any information comparing to OLAP. This compression can be used also for the attributes with a many-valued domain. In Figure 1 the example of such compression is show. From the database table (i), OLAP cube is computed with the space $(3+1) \times (2+1) = 12$ cells (ii). In (iii) the formal context (using the nominal scaling) is shown and finally in (iv) the concept lattice is depicted. The concept lattice has only 10 concepts with the values (1 trivial concept is just technical, with no value). Two OLAP cube cells (in (ii) are highlighted using gray color) are missing in the compressed concept lattice. Consider the well known dataset "Mushroom", which contains 23 original attributes ($22 + 1$ class considered as an attribute) where a cardinality of the domains is between 2 and 12. Using a formula for the OLAP space we get $7,36 \times 10^{16}$ of cells in the OLAP cube. Comparing to the amount of the formal concepts, which is $2,39 \times 10^{05}$ we get the space reduced approximately by 10^{12}. In the Table 1 we can see the original OLAP spaces comparing to the reduced ones using five well-known datasets (see the highlighted items with a significant space compression).

TradeMark	Country	Price in 000 EUR
BMW	Germany	30
BMW	France	35
SKODA	Germany	20
SKODA	France	25
FIAT	France	13

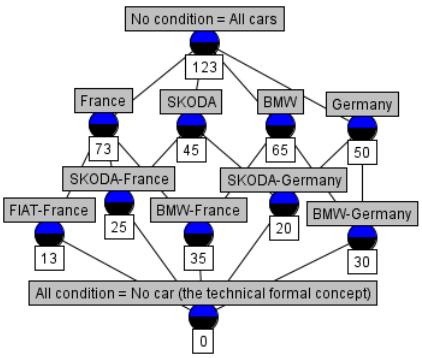
(i) Database

	All countries	France	Germany
All trademarks	123	73	50
FIAT	**13**	13	**0**
SKODA	45	25	20
BMW	65	35	30

(ii) OLAP Cube

Car nr.	BMW	SKODA	FIAT	Germany	France	Price in 000 EUR
1	×			×		30
2	×				×	35
3		×		×		20
4		×			×	25
5			×		×	13

(iii) Formal context with measures



(iv) Concept lattice with extent values

Fig. 1. OLAP space compression - example

	Dataset				
	Mushrooms	**Adults**	Cars	**Wine**	Tic-tac-toe
Nr. of original attributes (before scaling)	23	14	6	13	29
Nr. of formal attributes (after scaling)	119	124	25	68	29
Nr. of objects	8 124	32 561	1 728	178	958
original OLAP space (nr. of cells in OLAP cube)	7,36E+16	7,64E+10	4,00E+04	5,22E+10	7,86E+05
compressed OLAP space (nr. of formal concepts)	2,39E+05	1,06E+06	1,26E+04	2,54E+04	5,95E+04
compression ratio (compressed / original)	**3,25E-12**	**1,39E-05**	3,16E-01	**4,86E-07**	7,57E-02

Table 1. OLAP space compression, source of datasets: http://fcarepository.com

Remark 1: Not the whole OLAP cube is presented to a user and also not the whole lattice with values is presented to user. The data are stored using the different way, but the presentation to user can be the same (e.g. using pivot tables, pivot charts or other repost).

Remark 2: Not the whole OLAP cube is materialized in a real application. However a compression ratio is calculated from the whole OLAP cube comparing to compressed one (e.g. see [13]).

In [13] there were presented another solution how to compress OLAP cube, thus using a Dwarf Cube. The authors of [13] claim, that a Petabyte 25-dimensional cube was shrunk this way to a 2.3GB Dwarf Cube. For a detailed description of Dwarf cube we refer to [13]. Here we only compare our approach using a toy example from [13]. In Table 2 data for OLAP are shown and in Table 3 a comparison of tuples is shown for two OLAP representations (Dwarf and FCA with measures). In Table 3 we can see, that Dwarf Cube contains more tuples than

Store	Customer	Product	Price
S1	C2	P2	70
S1	C3	P1	40
S2	C1	P1	90
S2	C1	P2	50

Table 2. Data for OLAP

concept lattice. But this is only a toy example. Our hypothesis is, that FCA with measures contains a minimal possible amount of all non-redundant tuples for the OLAP cube compression. A formal proof as well as an experimental study will be part of our future research.

	Tuple in the Dwarf Cube	Tuple derived from an intent
1	$\langle S1, C2, P2\rangle$	$\langle S1, C2, P2\rangle$
2	$\langle S1, C2, ALL\rangle$	
3	$\langle S1, C3, P1\rangle$	$\langle S1, C3, P1\rangle$
4	$\langle S1, C3, ALL\rangle$	
5	$\langle S1, ALL, P1\rangle$	
6	$\langle S1, ALL, P2\rangle$	
7	$\langle S1, ALL, ALL\rangle$	$\langle S1, ALL, ALL\rangle$
8	$\langle S2, C1, P1\rangle$	$\langle S2, C1, P1\rangle$
9	$\langle S2, C1, P2\rangle$	$\langle S2, C1, P2\rangle$
10	$\langle S2, C1, ALL\rangle$	$\langle S2, C1, ALL\rangle$
11	$\langle ALL, ALL, P1\rangle$	$\langle ALL, ALL, P1\rangle$
12	$\langle ALL, ALL, P2\rangle$	$\langle ALL, ALL, P2\rangle$
13	$\langle ALL, ALL, ALL\rangle$	$\langle ALL, ALL, ALL\rangle$

Table 3. Dwarf Cube vs. FCA with measures

4 Attribute Hierarchy In OLAP And In FCA With Measures

In the paper [9] we claim, that FCA with measures is a generalization of OLAP cube. This claim however excludes the case, when a hierarchy of attributes in OLAP is defined. Attributes in a dimension can be split into smaller parts, e.g. in the dimension $Date$ we can consider the hierarchy $Year > Month$. In FCA with measures we can consider the dimension "Date" and it can be nominally scaled into attributes $Year$ and $Month$ as well. Consider the following Figure 2. When the original table (i) is scaled (ii) and formal concepts are computed, we get the concepts with the intent $\{Jan\}$ and $\{Feb\}$. Such intents can generally be used e.g. for analyzing the seasonality, however using the hierarchy $Year > Month$ such intent is not interesting (in this case it is a total amount of all cars sold in January regardless of the year). All other formal concepts are reasonable (i.e. total amount in one year or total amount in one month of the particular year). There are two possibilities how to deal with such problem. The first approach

Obj.	Date
1	Jan, 2011
2	Feb, 2011
3	Jan, 2012
4	Feb, 2012

(i) Original data

Obj.	2011	2012	Jan	Feb
1	×		×	
2	×			×
3		×	×	
4		×		×

(ii) Nominal scaling

Obj.	2011	2012	2011 Jan	2011 Feb	2012 Jan	2012 Feb
1	×		×			
2	×			×		
3		×			×	
4		×				×

(iii) Hierarchical scaling

Fig. 2. Hirerarchy of attributes

is just to scale the original data from (i) using a hierarchy (iii). Such scaling directly enables to avoid undesired formal concepts with intents such as $\{Jan\}$ and $\{Feb\}$ (Note: The undesired formal concept with the intent *all attributes* technically remains just to form a lattice).

Another option is to use AD formulas proposed in [11, 12]. An *AD formula* over a set Y of attributes is an expression $A \sqsubseteq B$, where $A, B \subseteq Y$. $A \sqsubseteq B$ is true in $K \subseteq Y$ if whenever $A \cap K \neq \emptyset$, then $B \cap K \neq \emptyset$. For a given set T of AD formulas over Y and a formal context $\langle X, Y, I \rangle$ we get the concept lattice constrained by T, which is denoted by $\mathcal{B}_T(X, Y, I)$. Such lattice consists of formal concepts of $\langle X, Y, I \rangle$ in which all AD formulas from T are true. For more details we refer to [11, 12]. In our example we can use AD formula $\{Jan, Feb\} \sqsubseteq \{2011, 2012\}$, which means: whenever we have a month in an intent of a formal concept (here Jan or Feb), we need also to have a year in intent (here 2011 and 2012). In other words, a year is hierarchically higher than a month. Constraining the original concept lattice by AD formula, undesired formal concepts will be avoided. A formal concept analysis with measures using AD formula can be seen as a generalization of OLAP with hierarchies.

In [14] there were presented some types of a hierarchy used in OLAP cube, but not all types of a hierarchy can be defined using AD formula. In this paper a preliminary results are presented (all examples of hierarchies are taken from [14]) :

1. simple hierarchies (represented by a tree)
   (a) symmetric hierarchy:
   $\{Department\ A\} \sqsupseteq \{Category\ 1\}$, $\{Department\ A\} \sqsupseteq \{Category\ 2\}$, $\{Category\ 1\} \sqsupseteq \{Product\ 1\}$, $\{Category\ 1\} \sqsupseteq \{Product\ 2\}$, $\{Category\ 2\} \sqsupseteq \{Product\ 3\}$, $\{Category\ 2\} \sqsupseteq \{Product\ 4\}$
   (b) asymmetric hierarchy:
   $\{bank\ X\} \sqsupseteq \{branch\ 1\}$, $\{bank\ X\} \sqsupseteq \{branch\ 2\}$, $\{bank\ X\} \sqsupseteq \{branch\ 3\}$, $\{branch\ 1\} \sqsupseteq \{agency\ 11\}$, $\{branch\ 1\} \sqsupseteq \{agency\ 12\}$, $\{branch\ 3\} \sqsupseteq \{agency\ 31\}$, $\{branch\ 3\} \sqsupseteq \{agency\ 32\}$, $\{agency\ 11\} \sqsupseteq \{ATM\ 111\}$, $\{agency\ 11\} \sqsupseteq \{ATM\ 112\}$
   (c) generalized hierarchy:
   $\{area\ A\} \sqsupseteq \{branch\ 1\}$, $\{area\ A\} \sqsupseteq \{branch\ 2\}$, $\{branch\ 1\} \sqsupseteq \{class\ 1\}$, $\{class\ 1\} \sqsupseteq \{profession\ A\}$, $\{class\ 1\} \sqsupseteq \{profession\ B\}$,$\{profession\ B\} \sqsupseteq \{customer\ X\}$, $\{profession\ B\} \sqsupseteq \{customer\ Y\}$, $\{branch\ 1\} \sqsupseteq \{sector\ 1\}$, $\{sector\ 1\} \sqsupseteq \{type\ A\}$, $\{sector\ 1\} \sqsupseteq \{type\ B\}$, $\{type\ B\} \sqsupseteq \{customer\ Z\}$, $\{type\ B\} \sqsupseteq \{customer\ K\}$
2. non-strict hierarchy:
   $\{division\ A\} \sqsupseteq \{Section\ 1, Section\ 2, Section\ 3\}$, $\{Section\ 1, Section\ 2, Section\ 3\} \sqsupseteq \{employee\ X\}$

This approach we can use also on for attributes which are not organized in dimensions by telling which group of independent attributes is more important than other group. In the example with cars (see the Tables 4 and 5) there are attributes Air Conditioning (AC), Airbag (AB), Antilock Braking System (ABS), Tempomat (TMP), Extra Guarantee (EG) and Automatic Transmission (AT).We can say, that $\{AB, ABS\}$ are more important (because of security) than $\{AC, TMP, EG, AT\}$ (which are used just for a higher comfort). AD formula in this case is $\{AB, ABS\} \sqsubseteq \{AC, TMP, EG, AT\}$, which means, that we

will care about values of formal concepts (e.g. the $Total\ Price$) only for such cars, which posses at least one of the security attributes AB or ABS (in the Table 5 labeled by $*$).).

	1. AC	2. AB	3. ABS	4. TMP	5. EG	6. AT	$\Phi(X)$ = Price in EUR
Car1	×	×					16 000
Car2		×	×	×			12 000
Car3		×	×	×	×		14 000
Car4	×			×	×		16 000
Car5	×				×		12 000
Car6	×	×	×				12 000
Car7		×	×	×			12 000
Car8			×				14 000
Car9							16 000
Car10		×					12 000
Car11		×		×			12 000
Car12	×	×	×	×	×	×	14 000
Car13		×		×			16 000
Car14	×	×		×	×	×	16 000
Car15			×		×		14 000
Car16	×	×					12 000
Car17	×	×					12 000
Car18		×	×	×			16 000
Car19			×				16 000
Car20	×	×	×		×		14 000
$\Psi(Y)$ = Price in EUR	1 000	500	800	600	250	100	

Table 4. The formal context of the cars, the additional components, the price of the car and the price of the component [9]

5 Conclusion

FCA with measures as a new area is just on the beginning. Based on our preliminary research it appears, that FCA with measures can significantly reduce a space of OLAP cube. FCA with measures can also be used as a generalization of OLAP even different hierarchy of attributes is included. In the future research we will focus on the detailed experimental research, where we will compare other reducing techniques of an OLAP cube space with our approach.

References

1. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.

	Extent Cars	Intent Components	Extent value Total Price	secure cars
1	X - all cars	$\emptyset$	278 000	
2	$\{2, 3, 4, 7, 11, 12, 13, 14, 18\}$	$\{TMP\}$	128 000	
3	$\{3, 4, 5, 12, 14, 15, 20\}$	$\{EG\}$	100 000	
4	$\{1, 4, 5, 6, 12, 14, 16, 17, 20\}$	$\{AC\}$	124 000	
5	$\{1, 2, 3, 6, 7, 10, 11, 12, 13, 14, 16, 17, 18, 20\}$	$\{AB\}$	190 000	$*$
6	$\{2, 3, 6, 7, 8, 12, 15, 18, 19, 20\}$	$\{ABS\}$	138 000	$*$
7	$\{3, 4, 12, 14\}$	$\{EG, TMP\}$	60 000	
8	$\{4, 5, 12, 14, 20\}$	$\{EG, AC\}$	72 000	
9	$\{2, 3, 7, 11, 12, 13, 14, 18\}$	$\{AB, TMP\}$	112 000	$*$
10	$\{3, 12, 14, 20\}$	$\{AB, EG\}$	58 000	$*$
11	$\{3, 12, 15, 20\}$	$\{ABS, EG\}$	56 000	$*$
12	$\{1, 6, 12, 14, 16, 17, 20\}$	$\{AC, AB\}$	96 000	$*$
13	$\{2, 3, 6, 7, 12, 18, 20\}$	$\{AB, ABS\}$	94 000	$*$
14	$\{4, 12, 14\}$	$\{AC, TMP, EG\}$	46 000	
15	$\{3, 12, 14\}$	$\{TMP, EG, AB\}$	44 000	$*$
16	$\{12, 14, 20\}$	$\{EG, AB, AC\}$	44 000	$*$
17	$\{2, 3, 7, 12, 18\}$	$\{AB, ABS, TMP\}$	68 000	$*$
18	$\{3, 12, 20\}$	$\{ABS, EG, AB\}$	42 000	$*$
19	$\{6, 12, 20\}$	$\{ABS, AC, AB\}$	40 000	$*$
20	$\{12, 14\}$	$\{AB, EG, AC, TMP, AT\}$	30 000	$*$
21	$\{3, 12\}$	$\{AB, EG, TMP, ABS\}$	28 000	$*$
22	$\{12, 20\}$	$\{AB, AC, EG, ABS\}$	28 000	$*$
23	$\{12\}$	Y - all components	14 000	$*$

Table 5. The formal concepts with the extent value [9]

2. Codd E.F., Codd S.B., and Salley C.T.: Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate *Codd & Date* (1993)
3. Wang Z., Klir G.: *Generalized measure theory*, Springer, New York, 2009
4. Maier D.: *The theory of relational databases*, Computer Science Press, Rockville, 1983
5. Kuznetsov S. D., Kudryavtsev A.: A mathematical model of the OLAP cubes, Programming and Computer Software, 2009, Vol. 35, No. 5, pp. 257–265. Pleiades Publishing, Ltd., 2009.
6. Calvo T., Kolesárová A., Komorníková M., Mesiar R. *Aggregation operators: Properties, classes and construction methods* Aggregation Operators: New Trend and Applications, p. 3-106 , Eds: Calvo T., Mayor G., Mesiar R., Physica Verlag, (Heidelberg 2002)
7. Belohlavek R., Vychodil V.: *Background Knowledge in Formal Concept Analysis: Constraints via Closure Operators.* ACM SAC 2010, 1113–1114.
8. Belohlavek R., Vychodil V.: *Formal concept analysis with constraints by closure operators.* In: H. Scharfe, P. Hitzler, and P. Ohrstrom (Eds.): Proc. ICCS 2006, Lecture Notes in Artificial Intelligence 4068, pp. 131-143, Springer-Verlag, Berlin Heidelberg, 2006.
9. Macko J.: *Formal Concept Analysis as a Framework for Business Intelligence Technologies.* In: F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, Springer, Heidelberg, 2012, pp. 195-210.
10. Kanovsky J., Macko J.: *ConSeQueL - SQL Preprocessor Using Formal Concept Analysis with Measures* CUBIST 2012 workshop

11. Belohlavek R., Sklenář V.: Formal concept analysis constrained by attribute-dependency formulas. In: B. Ganter and R. Godin (Eds.): ICFCA 2005, *Lect. Notes Comp. Sci.* **3403**, pp. 176–191, Springer-Verlag, Berlin/Heidelberg, 2005.
12. Belohlavek R., Vychodil V.: Formal concept analysis with background knowledge: attribute priorities. *IEEE Trans. Systems, Man, and Cybernetics, Part C* Volume 39 Issue 4, July 2009, pp. 399–409. DOI: 10.1109/TSMCC.2008.2012168.
13. Sismanis Y., Deligiannakis A., Roussopoulos N., Kotidis Y.: Dwarf: Shrinking the petacube *Proceedings of the 2002 ACM SIGMOD international conference on Management of data, p.464-475*
14. Malinowski E., Zimányi E. : Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering 59.2 (2006): 348-377)*