k-Partite Graphs as Contexts

Alexandre Bazin and Aurélie Bertaux

Le2i, Université Bourgogne Franche-Comté, Dijon, France

Abstract In formal concept analysis, 2-dimensional formal contexts are bipartite graphs. In this work, we generalise the notions of context and concept to graphs that are not bipartite. We then study the complexity of the enumeration and identify the structure of the set of such concepts.

1 Introduction

Formal concept analysis (FCA) is a mathematical framework centered on the notions of formal context (data) and formal concept (significant patterns). Most of the simpler real-life data sets take the form of formal contexts and the interesting patterns are often variations on the theme of formal concepts, making FCA well-suited for applications in any field that deals with data [3,10,6,12]. However, it has its limitations. With the increasing complexity of data, FCA requires extensions and generalisations such as fuzzy or multi-dimensional approaches [2,1,7,13].

Formal contexts in their basic form are binary tables – i.e. bipartite graphs for which a bipartition into independent sets is given. One of the most important generalizations of FCA, Polyadic Concept Analysis (PCA) [13], deals with the same notions of context and concept when said context is an n-uniform[1] n-partite[2] hypergraph – modeling the majority of multidimensional data sets. In PCA, again, an n-partition of the hypergraph is given. This trend can be found in all variants of FCA : the number of dimensions is the size of the data tuples.

We believe that it would be interesting, ultimately, to generalise FCA to n-partite hypergraphs that are not n-uniform in order to create new opportunities of applications involving exotic data. In this work, as a first step toward this goal, we focus on the case of n-partitioned graphs (2-uniform hypergraphs) with $n > 2$. We define the corresponding "concepts", briefly study the complexity of their enumeration and show that they form a complete n-lattice, implying that known algorithms can be used to compute them.

[1] i.e. hypergraph such that all its hyperedges have size n

[2] i.e. the set of graph vertices is decomposed into n disjoint sets such that no two graph vertices within the same set are adjacent

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 59–67,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University Olomouc, 2018.

2 Basics

This section briefly presents the basic notions in formal concept analysis and polyadic concept analysis. For a deeper look into the 2-dimensional case, we refer the reader to [5].

2.1 Binary Formal Concept Analysis

Definition 1 *A* (formal) context *is a triple* (S_1, S_2, R) *in which* S_1 *and* S_2 *are sets of what is commonly referred to as* objects *and* attributes *and* R *is a binary relation between objects and attributes representing the fact that an object is* described *by an attribute.*

A formal context is usually represented by a crosstable.

R	a	b	c	d	e
1	×	×			
2		×	×	×	
3		×		×	×
4			×		×
5				×	×

Figure 1. A formal context $(\{1, 2, 3, 4, 5\}, \{a, b, c, d, e\}, R)$

Definition 2 *Let* $\mathcal{C} = (S_1, S_2, R)$ *be a context. A* (formal) concept *of* $\mathcal{C}$ *is a pair* $(E \subseteq S_1, I \subseteq S_2)$ *such that* $E \times I \subseteq R$ *and both* E *and* I *are maximal for this property.*

In other words, a concept is a maximal rectangle full of crosses up to permutation of objects or attributes, also called in graph theory: a full bipartite subgraph or a biclique.

In our Fig. 1 example, $(1, ab)$ and $(23, bd)$ are concepts.

The set of concepts can be ordered by the inclusion relation on both objects and attributes and then forms a complete lattice (i.e. graph of concepts). Every complete lattice is isomorphic to the concept lattice of some context [5].

2.2 Multidimensional Formal Concept Analysis

The notions of formal contexts and concepts have been extensively studied and are successfully used in various fields such as data mining, data analysis, information retrieval, source code error correction, machine learning and for building taxonomies and ontologies [9]. The multidimensional generalization of FCA, polyadic concept analysis [13], has received comparatively less attention but is a promising theoretical as well as applicative field. Let us present here the basics.

Definition 3 *An* n-context *is a tuple* $(S_1, \dots, S_n, R)$ *in which* S_i, $i \in \{1, \dots, n\}$, *is a set called a* dimension *and* $R \subseteq \prod_{i \in \{1,\dots,n\}} S_i$ *is an* n*-ary relation.*

An n-context can be represented by an n-dimensional crosstable.

	a	b	c	a	b	c	a	b	c
1	×	×		×			×		
2	×			×			×	×	
3	×			×		×			×
		α			β			γ	

Figure 2. A 3-context $(\{1,2,3\}, \{a,b,c\}, \{\alpha,\beta,\gamma\}, R)$

Definition 4 *Let* $\mathcal{C} = (S_1, \dots, S_n, R)$ *be an* n*-context. An* n*-concept of* $\mathcal{C}$ *is an* n*-tuple* $(T_1, \dots, T_n)$ *such that* $T_i \subseteq S_i$, $\prod_{i \in \{1,\dots,n\}} T_i \subseteq R$ *and there is no* $d \in \{1, \dots, n\}$ *and* $k \in S_d \setminus T_d$ *such that* $(T_1, \dots, T_d \cup \{k\}, \dots, T_n)$ *respects this property.*

In other words, an n-concept is a maximal n-dimensional box full of crosses in $\mathcal{C}$ up to permutations inside dimensions.

In our Fig. 2 example, $(\{1,2,3\}, \{a\}, \{\alpha,\beta\})$ and $(\{2\}, \{a,b\}, \{\gamma\})$ are 3-concepts.

The set of all the n-concepts in an n-context, together with the n quasi-orders induced by the inclusion relation on the subsets of each dimension, forms an n-lattice and each complete n-lattice is isomorphic to the concept lattice of an n-context, as stated in the basic theorem of polyadic concept analysis [13].

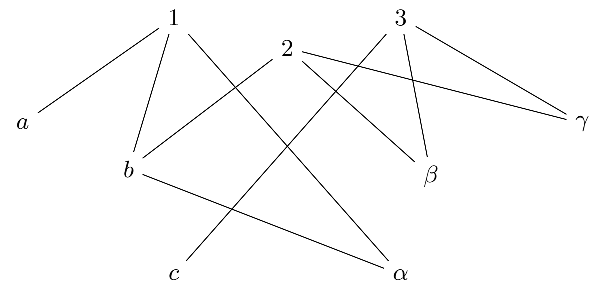

Figure 3. Graph that will be used as running example.

2.3 Graphs

A graph is a pair $G = (V, E)$ in which V is a set of elements called *vertices* and $E \subseteq V^2$ a set of *edges*.

A set $X \subseteq V$ of vertices is a *clique* if there is an edge between any two of its elements. A clique is *maximal* if it is not contained in another clique. An *independent set* is a set of vertices that does not contain any edge. An independent set is *maximal* if it is not contained in any independent set. A *vertex cover* is a set of vertices that contains at least one vertex from every edge. A vertex cover is *minimal* if it does not contain any vertex cover. A (maximal) independent set in a graph G is a (maximal) clique in the complementary graph $\overline{G}$ and reciprocally. The complement of a (maximal) independent set is a (minimal) vertex cover and reciprocally.

We will use $\mathcal{M}(G)$ to denote the set of maximal cliques in a graph G.

A graph $G = (V, E)$ is *k-partite* iff V can be partitioned into k independent sets.

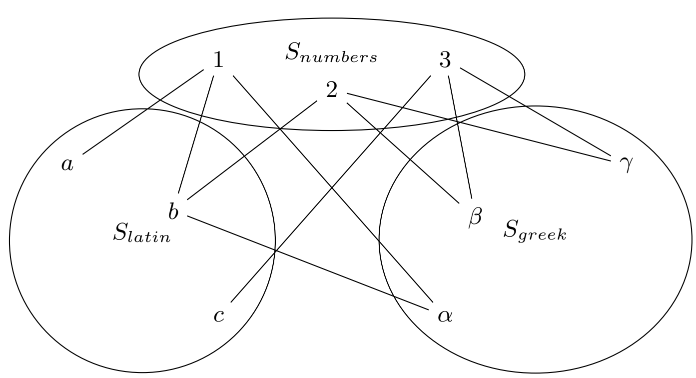

Figure 4. Partition of our example graph into three independent sets $S_{numbers}$, S_{latin} and S_{greek}.

A *complete k-partite graph* is a k-partite graph such that there is an edge between every pair of vertices that do not belong to the same independent set.

In our running example, the subgraphs induced by the vertices sets $\{1, b, \alpha\}$ and $\{1, a, b\}$ are, respectively, complete tripartite and bipartite graphs.

Bidimensional formal contexts (S_1, S_2, R) are bipartite graphs $(S_1 \cup S_2, R)$ for which a bipartition is given. In graph terminology, 2-concepts are thus maximal complete bipartite subgraphs of the context.

3 k-Partite Graphs as Contexts

FCA offers tools to find and manipulate patterns in bipartite graphs. What happens to these patterns and tools when the input graph is not bipartite ?

3.1 Defining the Concepts

Let us start by defining the objects we are looking for. The central patterns in FCA are concepts : maximal complete bipartite subgraphs of the context. When the context is k-partite, a natural generalisation can then be expressed as follows.

Definition 5 *Let $G = (V, E)$ be a graph and $S = (S_1, \ldots, S_k)$ a partition of V into k independent sets. Let $\{j_1, \ldots, j_m\} \subseteq \{1, \ldots, k\}$. An m-$_2$*concept *of (S, E) is a tuple $C = (C_{j_1}, \ldots, C_{j_m})$, $C_{j_x} \neq \emptyset$, $C_{j_x} \subseteq S_{j_x}$, such that $\bigcup_{x \in \{1,\ldots,m\}} C_{j_x}$ induces a maximal complete m-partite subgraph of G and there is no $(C_{j_1}, \ldots, C_{j_m}, C_{j_{m+1}})$ with this property.*

In "m-$_2$concept", the m means that we consider an m-partite graph as "concept" (m dimensions are involved in the pattern) and the $_2$ means the pattern is found in a 2-uniform graph. We have chosen to define them as m-tuples instead of k-tuples with $m \leq k$ in order to avoid having to consider the $m - k$ empty components and confusion with k-concepts from PCA.

We will now suppose, for the remainder of this paper, that our running example is partitioned as in Fig. 4. In this case, $(1, b, \alpha)$ is a 3-$_2$concept and $(1, ab)$ and $(23, \beta\gamma)$ are 2-$_2$concepts. The tuple $(3, c, \beta\gamma)$ is not a 3-$_2$concept because the induced subgraph is complete bipartite, not complete tripartite[3]. The tuple $(1, \alpha)$ is not a 2-$_2$concept because $(1, b, \alpha)$ is a 3-$_2$concept.

When the graph is bipartite **and the partition provided is binary**, the 2-$_2$concepts are the formal concepts with non-empty intents and extents. It is important to note that S_i, $i \in \{1, \ldots, k\}$, is a complete 1-partite subgraph – though (S_i) is not necessarily a 1-$_2$concept.

We will use $\mathcal{T}((S, E))$ to denote the set of m-$_2$concepts, $1 < m \leq |S|$, of a k-partite graph (V, E) together with a partition S of V into k independent sets.

[3] Two sets are considered {3} and {c,$\beta\gamma$} without relations between {c} and {$\beta\gamma$}

Proposition 1 *Let (V, E) be a graph and $S = (S_1, \ldots, S_k)$ a partition of V into k independent sets.*

$$\mathcal{T}((S, E)) = \mathcal{M}((V, E \cup X))$$

with $X = \bigcup_{i \in \{1,\ldots,k\}} \binom{S_i}{2}$

Proof. In $G = (V, E \bigcup_{i\in\{1,\ldots,k\}} \binom{S_i}{2})$, we have that $\forall i \in \{1, \ldots, k\}$, S_i is a clique. Let $C = (C_{j_1}, \ldots, C_{j_m})$ with $C_{j_i} \subseteq S_{j_i}$ be such that $\bigcup_{i\in\{1,\ldots,m\}} C_{j_i}$ is a maximal clique in G. By definition, any two vertices $x \in C_{j_a}$ and $y \in C_{j_b}$, $a \neq b$ are neighbours in G. As such, they are neighbours in (V, E) too. Clearly, that makes C an m-partite complete subgraph of (V, E). The maximality property holds from one graph to the other so C is an m-$_2$concept of (V, E).

Let $C = (C_{j_1}, \ldots, C_{j_m})$ be an m-$_2$concept of (V, E). By definition, any two vertices $x \in C_{j_a}$ and $y \in C_{j_b}$, $a \neq b$ are neighbours in (V, E). As such, they are neighbours in G. As, $\forall i \in \{1, \ldots, k\}$, S_i is a clique, $\bigcup_{i\in\{1,\ldots,m\}} C_{j_i}$ is a clique in G. The maximality property once again holds from one graph to the other so $\bigcup_{i\in\{1,\ldots,m\}} C_{j_i}$ is a maximal clique in G. □

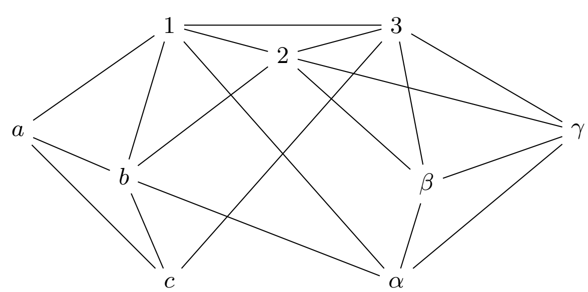

Figure 5. Our example graph with its partitions made into cliques.

This proposition states that m-$_2$concepts are maximal cliques in a graph that can be constructed in polynomial time from the context. This implies that $\mathcal{T}((S, E))$ can be computed from (S, E) in output-polynomial time [11].

3.2 Structuring the Concepts

We now have to characterise the structure of the set $\mathcal{T}((S, E))$. We will show that it forms a k-lattice when put together with the appropriate quasi-orders. The best way to do this is to show that $\mathcal{T}((S, E))$ is isomorphic to the concept k-lattice of a k-context.

Let $\mathcal{K}((S, E)) = (S_1 \cup \{s_1\}, \ldots, S_k \cup \{s_k\}, R)$ be a k-context such that $s_i \notin S_i$ and

$$(x_1, \ldots, x_k) \in R \iff \forall x_i \neq s_i, x_j \neq s_j, \exists e \in E \text{ such that } x_i, x_j \in e$$

Note that, potentially, $x_i = x_j$. In the context $\mathcal{K}((S, E))$ each cross corresponds to a clique of the graph (V,E), including 1-element ones, with the elements s_i representing the fact that a clique does not intersect the set S_i. Figure 6 illustrates the 3-context corresponding to our running example..

Clearly, if $(X_1, \ldots, X_k)$ is a k-concept of $\mathcal{K}((S, E))$, then $\forall i \in \{1, \ldots, m\}$, $s_i \in X_i$.

	a	b	c	s_2	a	b	c	s_2	a	b	c	s_2	a	b	c	s_2
1		×		×									×	×		×
2								×				×		×		×
3								×				×			×	×
s_1		×		×				×				×	×	×	×	×
	α				β				γ				s_3			

Figure 6. The 3-context $(\{1, 2, 3, s_1\}, \{a, b, c, s_2\}, \{\alpha, \beta, \gamma, s_3\}, R)$ corresponding to our running example.

Theorem 1. *Let (V, E) be a graph and S a k-partition of (V, E) into k independent sets. The set of m-$_2$concepts of (S, E), together with the k quasi-orders induced by the inclusion relation on each independent set, forms a k-lattice.*

Proof. Let $(X_1, \ldots, X_k)$ be a k-concept of $\mathcal{K}((S, E)) = (S_1 \cup \{s_1\}, \ldots, S_k \cup \{s_k\}, R)$. By definition, $\prod_{i \in \{1,\ldots,k\}} (X_i \setminus \{s_i\}) \subseteq R$. From the construction of $\mathcal{K}((S, E))$, we get that $\forall x_i \in X_i \setminus \{s_i\}, x_j \in X_j \setminus \{s_j\}, \exists e \in E$ such that $x_i, x_j \in e$. This means that the tuple $(X_{j_1} \setminus \{s_{j_1}\}, \ldots, X_{j_m} \setminus \{s_{j_m}\})$, such that the different $X_{j_i} \setminus \{s_{j_i}\}$ are the non-empty components of $(X_1 \setminus \{s_1\}, \ldots, X_k \setminus \{s_k\})$, is an m-$_2$concept of (S, E).

Let $(C_{j_1}, \ldots, C_{j_m})$ be an m-$_2$concept of (S, E). By definition, $\forall A \in \prod_{i \in \{1,\ldots,m\}} C_{j_i}$, $\forall x, y \in A$, $\exists e \in E$ such that $x, y \in e$. As such, the tuple $(X_1, \ldots, X_k)$ such that

$$X_i = \begin{cases} C_i \cup \{s_i\} & \text{if } i \in \{j_1, \ldots, j_m\} \\ \{s_i\} & \text{otherwise} \end{cases}$$

is a k-concept of $\mathcal{K}((S, E))$. $\square$

This implies that algorithms [4,8] for computing n-concepts can be used to compute m-$_2$concepts.

In Fig. 6, the 3-concepts are

$$\begin{array}{cc} (1s_1, bs_2, \alpha s_3) & (23s_1, s_2, \beta\gamma s_3) \\ (1s_1, abs_2, s_3) & (12s_1, bs_2, s_3) \\ (3s_1, cs_2, s_3) & (123s_1, s_2, s_3) \\ (s_1, abcs_2, s_3) & (s_1, s_2, \alpha\beta\gamma s_3) \end{array}$$

which yield the m-$_2$concepts of our running example once the s_i and empty sets are removed.

4 Conclusion

In this paper, we have extended the notions of formal context and concept to graphs that are not bipartitioned in order to allow the handling of a different kind of data. We have shown that, given a k-partition of the graph into independent sets, the set of such m-$_2$concepts forms a k-lattice. This allows the use of any k-lattice algorithm to compute m-$_2$concepts.

The next step would be to generalise the notion of n-concept to hypergraphs that are not n-partite n-uniform. This, however, is not as straightforward as m-$_2$concepts. Indeed, the k-lattice structure of m-$_2$concepts comes from the fact that a clique with n vertices can freely be converted into 2^n hyperedges (the subsets of vertices). Converting an edge (a, b) into two singletons (a) and (b) does not add complexity. However, converting an hyperedge (a, b, c) into a triangle (a, b), (b, c), (a, c) can potentially create new triangles that do not correspond to existing hyperedges of size 3.

References

1. Belohlavek, R.: What is a fuzzy concept lattice? II. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing pp. 19–26 (2011)
2. Burusco Juandeaburre, A., Fuentes-González, R.: The study of the L-fuzzy concept lattice. Mathware & soft computing. 1994 Vol. 1 Núm. 3 p. 209-218 (1994)
3. Carbonnel, J., Huchard, M., Miralles, A., Nebut, C.: Feature model composition assisted by formal concept analysis. In: ENASE: Evaluation of Novel Approaches to Software Engineering. pp. 27–37. SciTePress (2017)
4. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Data-peeler: Constraint-based closed pattern mining in n-ary relations. In: proceedings of the 2008 SIAM International conference on Data Mining. pp. 37–48. SIAM (2008)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edn. (1997)
6. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Information Sciences **181**(10), 1989–2001 (2011)
7. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. Conceptual structures: applications, implementation and theory pp. 32–43 (1995)
8. Makhalova, T., Nourine, L.: An incremental algorithm for computing n-concepts. In: Formal Concept Analysis for Knowledge Discovery. Proceedings of International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017), Moscow, Russia, June 1, 2017. CEUR-WS. org (2017)
9. Škopljanac Mačina, F., Blašković, B.: Formal concept analysis – overview and applications. Procedia Engineering **69**, 1258 – 1267 (2014). https://doi.org/https://doi.org/10.1016/j.proeng.2014.03.117, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

10. Snelting, G.: Software reengineering based on concept lattices. In: Software Maintenance and Reengineering, 2000. Proceedings of the Fourth European. pp. 3–10. IEEE (2000)
11. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing **6**(3), 505–517 (1977)
12. Valtchev, P., Missaoui, R., Godin, R.: Formal concept analysis for knowledge discovery and data mining: The new challenges. In: ICFCA. vol. 2961, pp. 352–371. Springer (2004)
13. Voutsadakis, G.: Polyadic concept analysis. Order **19**(3), 295–304 (2002)