Clustering with Axialities

Sergei O. Kuznetsov

HSE University, Pokrovsky bd. 11, Moscow, 109028, Russian Federation

Abstract

Formal concepts can be considered as rigid biclusters where all objects from the bicluster (formal extent) share all attributes from the intent. Relaxed versions of concept-based bicluster, e.g. OA-biclusters, are also well-known. In this note we show that *axial* (aka monotone, disjunctuve) concepts arising from axialities (adjunctions on powersets of objects and attributes) can help to perform clustering of tricky data like those where clusters are not separable by hyperplanes or present complex dynamical objects, where standard formal concepts and interval patterns would hardly help to catch the required patterns.

Keywords

Formal Concept Analysis, Clustering, Axial Concepts

1. Introduction

It is well known that Formal Concept Analysis (FCA) presents natural tools for clustering [1]. A formal concept can be considered as a rigid (bi)cluster where all objects of the (bi)cluster (formal extent) share all attributes of the intent, which embodies the similarity of the objects from the extent. Relaxed versions of concept-based bicluster, e.g. OA-biclusters [5, 6] are also well-known. Another well-studied FCA-based clustering model is the one based on interval pattern structures [7]. Clustering based on *equiconcepts* in symmetric contexts, where extents and intents coincide as studied in [12, 13]. In this note we show that axial (disjunctive [9]) concepts arizing from axialities (adjunctions aka residuated mappings or monotone Galois connections on powersets of objects and attributes) [1] can help to naturally cluster tricky data like dynamic streaming data or data of the form 1.1, 2.1, 4.1 in Figure 1, where clusters are dense sets of points with clear connectivity property, so that standard formal concepts and interval pattern concepts would hardly help to catch the required patterns.

2. Definitions and Main Idea

First, let us recall the definitions of (interval) pattern structure and pattern concept [4, 7].

A *pattern structure* [4] is a triple $(G, \mathbb{D}, \delta)$, which is a generalization of a formal context (G, M, I) so that G is a set of objects, $\mathbb{D} = (D, \sqcap)$ is a complete semilattice on descriptions from set D with meet (infimum) $\sqcap$, and $\delta : G \to \mathbb{D}$ takes an object from G to its description in D. For any pattern description $d \in \mathbb{D}$ one can define its *pattern extent* $d^{\diamond} = \{g \in G \mid d \sqsubseteq \delta(g)\}$ and for any subset of objects $A \subseteq G$ one can define its *pattern intent* $A^{\diamond} = \sqcap\{\delta(g) \mid g \in A\}$. A pair of corresponding pattern extent A and pattern intent d forms a *pattern concept*: (A, d), where $A^{\diamond} = d, d^{\diamond} = A$.

The description semilattice of an *interval pattern structure* [7] $\mathbb{D}_{\text{int}}$ consists of tuples of real-numbered intervals $\mathbb{D}_{\text{int}}$, where intervals are ordered by interval subsumption $\sqsubseteq_{\text{int}}$:

$\mathbb{D}_{\text{int}} = \{[l, r] \mid l, r \in \mathbb{R}, l \leq r\}$ and $\forall [l_1, r_1], [l_2, r_2] \in \mathbb{D}_{\text{int}}, [l_1, r_1] \sqcap_{\text{int}} [l_2, r_2] = [min\{l_1, l_2\}, max\{r_1, r_2\}]$ so that $[l_1, r_1] \sqsubseteq_{\text{int}} [l_2, r_2] \iff [l_1, r_1] \supseteq [l_2, r_2]$.

Interval pattern concepts propose a natural way of clustering numerical data as proposed in [7]. The experiments show that interval pattern concepts, whose intents make hyperrectangles with axis-aligned

FCA4AI 2024: The 12th International Workshop "What can FCA do for Artificial Intelligence?", October 19 2024, Santiago de Compostela, Spain.

skuznetsov@hse.ru (S. O. Kuznetsov)

0000-0003-3284-9001 (S. O. Kuznetsov)

edges and faces can be successfully used for clustering data like 5.1, 6.1 (Figure 1), can be used with less success in clustering data like 3.1 and perform much worse for data of the form 1.1, 2.1, 4.1.

So, in this note we propose another FCA-based tool - called axial (aka disjunctive, monotone [9]) concepts - which can help in clustering data that are hard to cluster using formal or interval pattern concepts.

Let $K = (G, M, I)$ be a formal context, then *axialities* (aka adjunctions or residuated mappings on powersets) [2] are defined for K as

$$^{\leftarrow}A = \{b \in M \mid aIb \text{ for no } a \in G \setminus A\}, \tag{1}$$

$$^{\rightarrow}B = \{a \in G \mid aIb \text{ for some } b \in B\}. \tag{2}$$

where $A \subseteq G$ is a subset of objects and $B \subseteq M$ is a subset of attributes.

An *axial* (or disjunctive [9]) *concept* based on axialities is defined in a similar way as the standard formal concept [3], i.e. as a pair (A, B), where $A \subseteq G, B \subseteq M$ and $A =^{\rightarrow} B, B =^{\leftarrow} A$. Unlike formal concepts, the extents and intents of axial concepts are isotonic, i.e. for two axial concepts (X_1, Y_1) and (X_2, Y_2) one has $X_1 \subseteq X_2$ iff $Y_1 \subseteq Y_2$

While for some clusterization tasks in the left column of Figure 1, like 3,5,6, the generalization of formal concepts to interval pattern concepts fits quite well, the clustering tasks 1,2,4 are hardly well-solvable by means of interval patterns, since they make only axis-aligned hyperrectangles and are insensitive to density and continuity properties of data.

Here we propose to apply axial concepts by first making a transformation of original data, which is well-known in Machine Learning as the “kernel trick”[10].

First, we introduce data model which will be studied further. Let G be a set of data points in a metric space with metric d. Let $A_1, \ldots, A_n$ be disjoint subsets of data points: $A_i \subseteq G$, $A_i \cap A_j = \emptyset$. We call the family of sets $A_1, \ldots, A_n$ (ε, k)-dataset if $d(a_i, a_j) > \varepsilon$ for every $a_i \in A_i$ and $a_j \in A_j$ where $i \neq j$.

Let us define the following formal context, which we call ε*-kernel context*: (G, G, I_ε), where $I_\varepsilon \subseteq G \times G$ is defined as $(g, h) \in I_\varepsilon$ iff $d(g, h) \leq \varepsilon$.

Proposition 2.1. *For each cluster A_i there is an axial concept (A_i, A_i) of the context (G, G, I_ε).*

Proof. By the construction of the context (G, G, I_ε) every subset $A \subseteq A_i$ makes the monotone concept (A, A). □

Example 1. Consider a simplified example of the “half moons” dataset of type 2.1 in Figure 2 where the set of data points is $G = \{g_1, \ldots, g_{12}\}$ as in Figure 2, with $A_1 = \{g_1, \ldots, g_6\}$, $A_2 = \{g_7, \ldots, g_{12}\}$ and $d(g_1, g_2), d(g_2, g_3), d(g_3, g_4), d(g_4, g_5), d(g_5, g_6) < \epsilon$ and $d(g_7, g_8), d(g_8, g_9), d(g_9, g_{10}), d(g_{10}, g_{11}), d(g_{11}, g_{12}) < \varepsilon$ and for any $g_i \in A_1$ and $g_j \in A_2$ one has $d(g_i, g_j) > \varepsilon$.Then the cross-table of (G, G, I_ε) is given in Table 1.

Consider now that ε takes values $\varepsilon_1 < \varepsilon_2 < \varepsilon_3$. For $\varepsilon = \varepsilon_1$ close to zero, the resulting clusters would contain only single points. Increasing ε to $\varepsilon = \varepsilon_2$ we obtain two clusters staying for sets A_1 and A_2. If we increase ε further to $\varepsilon = \varepsilon_3$, the clusters will merge in one.

Similar effects will be observed for data of the types 1.1, 4.1 in Figure 1. As for data of the types 3.1 and 5.1 where there are “bridges” between clusters, let us consider the following example.

A part of the diagram of the axial concept lattice for Example 1 is given in Figure 4.

Example 2. Consider a point dataset in Figure 5. Here two clusters A_1 and A_2 are not totally disjoint, but have a “bridge” element g_5 shared by both clusters.

Figure 6 gives the diagram of the axial concept lattice for the context in Table 2. Notation $\overline{a, b}$ with $a < b$ denotes the set of elements (both objects and attributes) $\{a, a+1, \ldots, b\}$. Every axial concept here corresponds to a cluster and every antichain of axial concepts corresponds to a clusterization where clusters may intersect.

Note that clustering in this case can also be easily performed by using formal concepts $(\overline{1,4}, \overline{1,4})$, $(\overline{4,6}, \overline{4,6})$, $(\overline{6,9}, \overline{6,9})$, $(4, \overline{1,6})$, $(6, \overline{4,9})$, with objects 4 and 6 playing the role of outliers in their clusters.

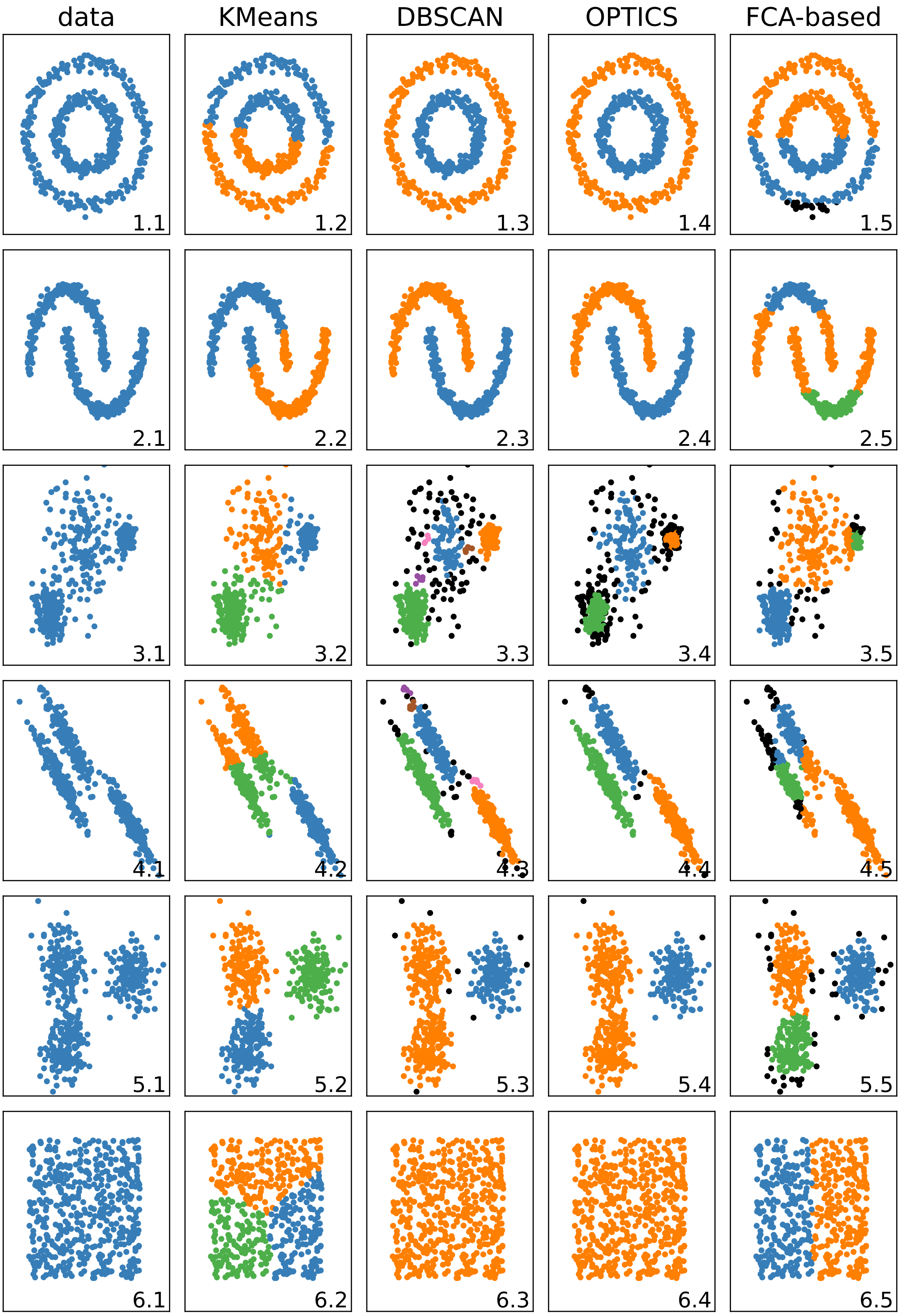

Figure 1: The left-most column presents clustering data from Sci-Kit learn https://scikit-learn.org/stable/modules/clustering.html. The other columns stay give visual comparison of clusterings based on various approaches: KMeans, DBSCAN, OPTICS and FCA-based. Dots colours correspond to clusters, black dots represent non-clustered objects (outliers).

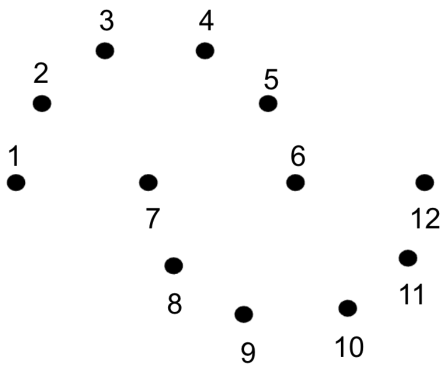

Figure 2: Simplified half moons data.

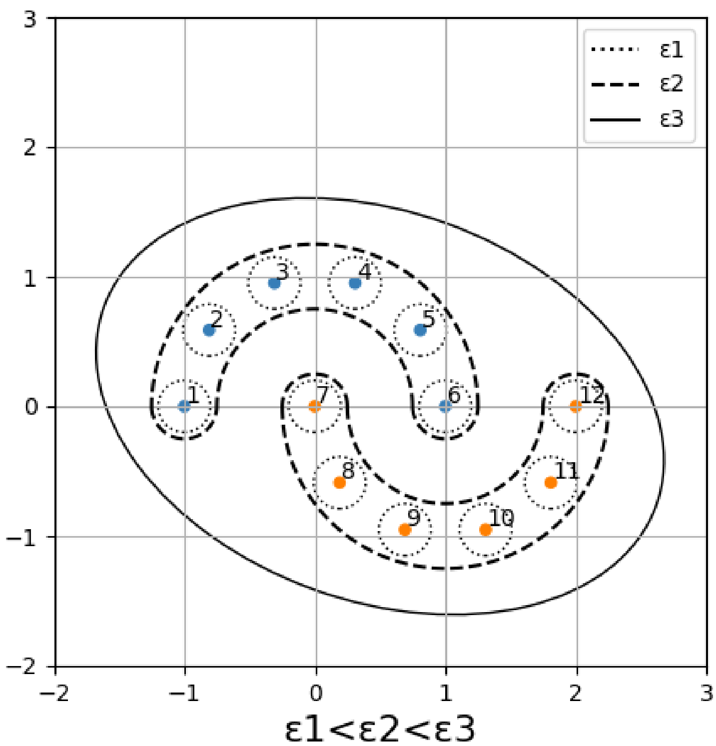

Figure 3: The dotted lines stay for ε_1, the dashed lines stay for ε_2, and the solid line stays for ε_3.

3. Computing clusters as axial concepts

It is well-known [2, 9] that (A, B) is an axial (disjunctuve) concept of context (G, M, I) iff $(G \setminus A, (G \setminus A)')$ is a formal concept of $(G, M, \bar{I})$. So, to compute axial concepts of (G, G, I_ϵ), one can use standard FCA algorithms like CbO [8]. For example, to compute maximal (both by extent and intent) axial concepts, one can compute minimal extents of $(G, G, \bar{I}_\epsilon)$, which can be done in $O(k \times |G|^2)$ time.

Although clusters correspond to axial extents of ε-kernel context, not every extent makes a "good" cluster. For Example 1 with the context in Table 1 every subset $\overline{1,k}$ for $k \in \overline{1,12}$, except for $k = 7$, makes an axial extent, however the desired cluster among them is only $\overline{1,6}$, which corresponds to

	1	2	3	4	5	6	7	8	9	10	11	12
1	x	x										
2	x	x	x									
3		x	x	x								
4			x	x	x							
5				x	x	x						
6					x	x						
7							x	x				
8							x	x	x			
9								x	x	x		
10									x	x	x	
11										x	x	x
12											x	x

Table 1
Context (G, G, I_ε) for Example 1 with ε_2.

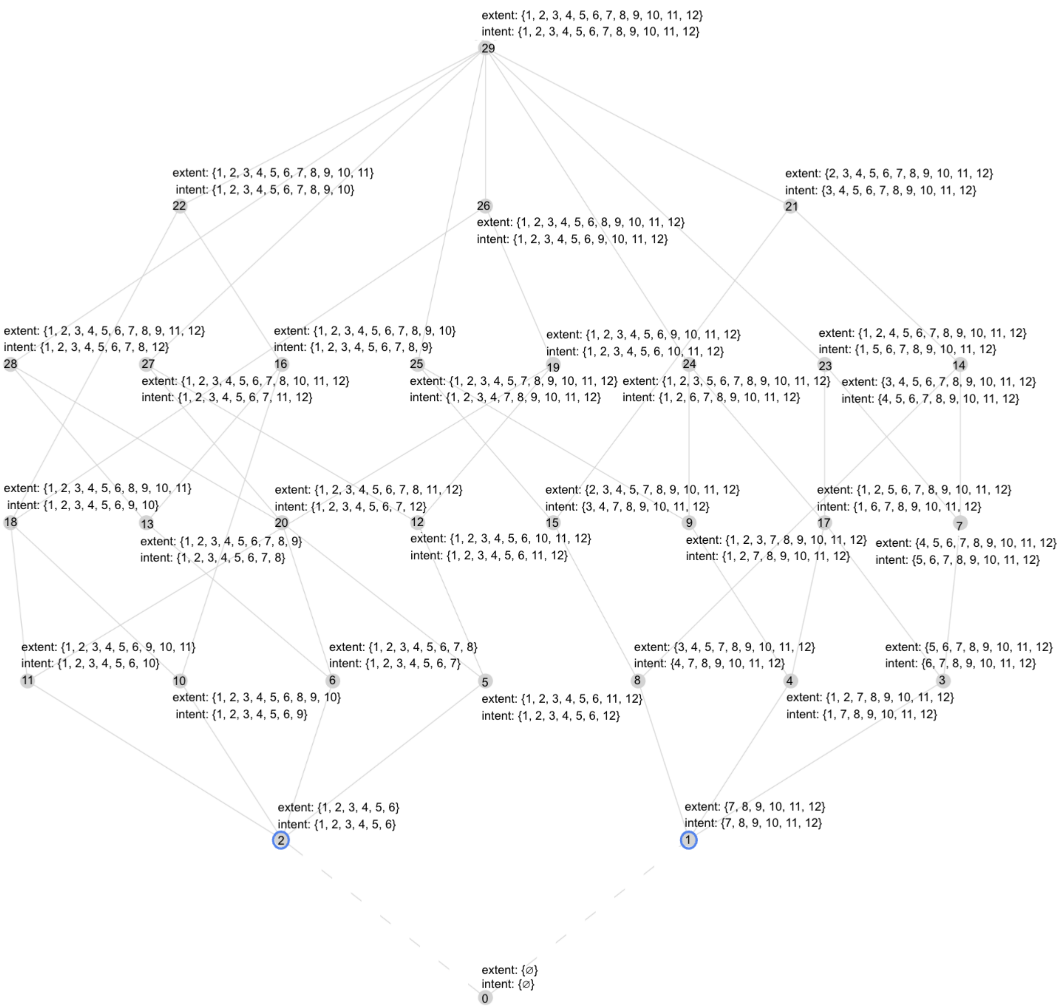

Figure 4: The desired clustering of points $1, \ldots, 12$ is seen as the antichain of two axial concepts $(\overline{1,6}, \overline{1,6})$ and $(\overline{7,12}, \overline{7,12})$.

the axial concept $(\overline{1,6}, \overline{1,6})$. Consider a CbO-like object-wise strategy of computing axial concepts by adding object $k+1$ to the current axial extent $\overline{1,k}$. Till $k = 6$ it runs in a uniform way by adding new row and new column. However, when one tries to add object 7 (or any of the objects 8,9,10,11,12) to

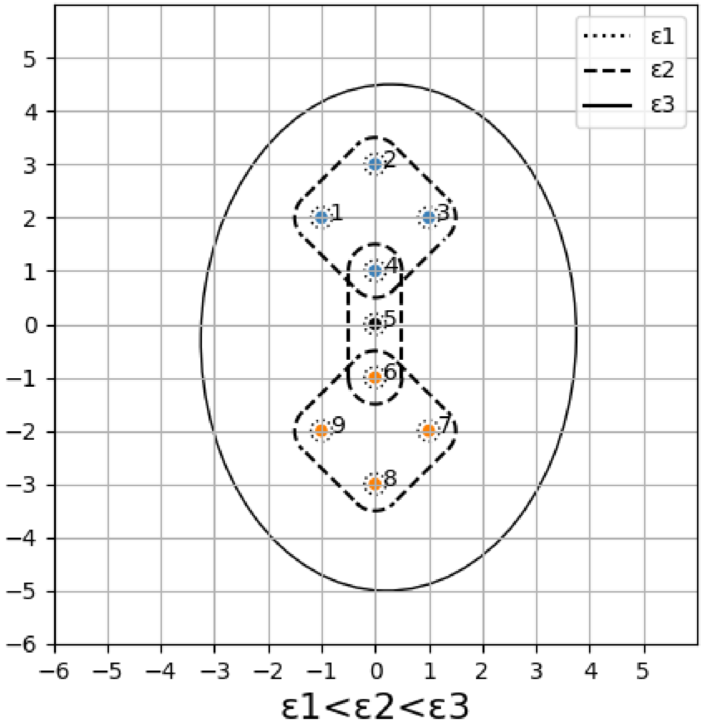

Figure 5: Clusters A_1 and A_2 share common element g_5.

	1	2	3	4	5	6	7	8	9
1	x	x	x	x					
2	x	x	x	x					
3	x	x	x	x					
4	x	x	x	x	x	x			
5				x	x	x			
6				x	x	x	x	x	x
7						x	x	x	x
8						x	x	x	x
9						x	x	x	x

Table 2
Context (G, G, I_ε) for Example 2 with ε_2.

the extent $\overline{1,6}$ of the concept $(\overline{1,6}, \overline{1,6})$, one again, performing $^\leftarrow$ and then $^\rightarrow$, obtains axial concept $(\overline{1,6}, \overline{1,6})$. This actually signifies that objects 7,8,9,10,11,12 have no similarity to objects $\overline{1,6}$ and the construction of the cluster should be terminated, making it $\overline{1,6}$. This observation can be formalized as a general rule as follows: if for a current axial extent A adding any new object and performing the composition of operations $\rightarrow \circ \leftarrow$ results in the old extent A, then one should output A as a cluster. One can design other similar rules based on a "termination condition" depending on the data and problem setting. Most desirable, such a condition should be (anti)monotonic, so that - once violated - would not hold with the addition of new objects, thus ensuring that termination without the need for further computation is justified.

For example, consider data in Figure 5 with ε_2 and respective context in Table 2. Since 5 has only two neighbors, the respective column and row have only three entries. All other elements have at least three neighbors. So, the algorithm computing axial concepts here may have a termination condition such that

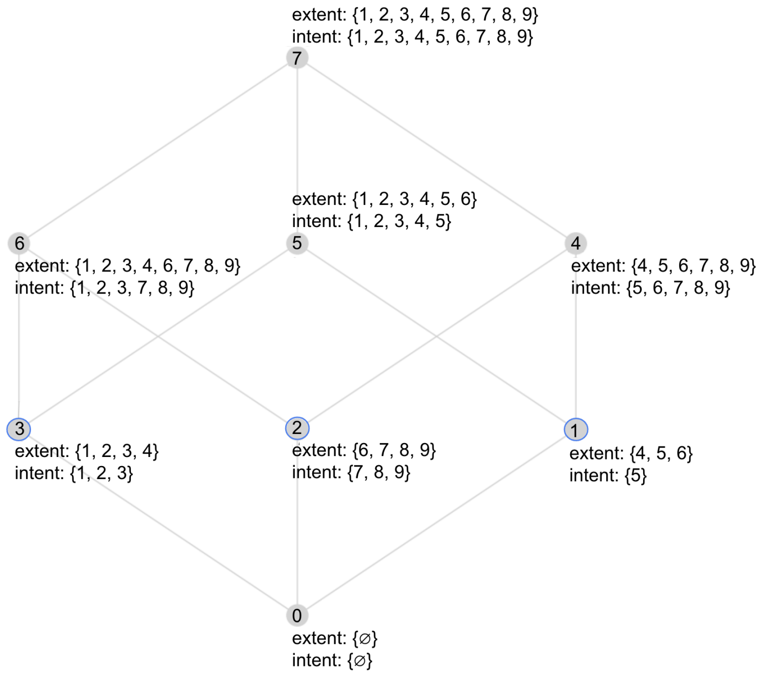

Figure 6: Diagram of the lattice of axial concepts for the context in Table 2.

if the algorithm gets a row (column) with less than 4 entries, thus outputting two clusters A_1 and A_2 as required. It is also worth noting that transforming initial data to the ε-kernel context given by a table results in quadratic increase of the data size, so the computation of axialities for clustering can be made more efficient if the algorithms are adapted to the initial data representation. Then, instead of traversing rows and columns of the kernel cross-table, one can operate with ε-neighborhoods (computed on the fly) of the points in original representation. In this form one arrives to an approach close to DBSCAN and its successors [11].

4. Conclusions

An idea of a clustering framework based on ε-kernel trick, axialities, and respective axial concepts was proposed. The clusters correspond to special types of axial concepts of the ε-kernel context related to the original dataset. Axialities propose a natural way to express continuity in clusters, where not all points of a cluster are close to each other, however, as in dynamical (e.g. streaming) data, there is a continuous "dense" path of neighboring points joining any two points of the cluster. The proposed formalization allows for a natural way of computing clusters by means of standard FCA-algorithms. However, not all axial concepts correspond to good clusters, so the main challenge for a particular clustering setting remains in finding easily computable termination conditions that would allow computing exactly those axial concepts that correspond to best clustering. The further study should also take into account the lessons learned in the development of density-based clustering approaches like DBSCAN and its successors.

5. Acknowledgements

The work by S.O. Kuznetsov in preparing this article was supported by grant 22-11-00323 of the Russian Science Foundation and carried out at the National Research University Higher School of Economics, Moscow.

6. Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] C. Carpineto and G. Romano, A lattice conceptual clustering system and its application to browsing retrieval. Machine learning (1996), 24, 95-122.

[2] M. Erné, Adjuunctions and Galois Connections. Origins, History and Development, in K. Denecke, M. Erné, S.L.Wismath (Eds.), Galois Connections and Applications, Springer, 2004.

[3] B. Ganter and R. Wille, Formal Concept Analysis, Mathematical Foundations (Second Edition), Springer (2024).

[4] B. Ganter, S.O. Kuznetsov: Pattern Structures and Their Projections. Proc. ICCS 2001: 129-142.

[5] D.I. Ignatov, S.O. Kuznetsov, J. Poelmans, L.E. Zhukov: Can triconcepts become triclusters? Int. J. Gen. Syst. 42(6): 572-593 (2013).

[6] D.I. Ignatov, D.V. Gnatyshak, S.O. Kuznetsov, B.G. Mirkin: Triadic Formal Concept Analysis and triclustering: searching for optimal patterns. Mach. Learn. 101(1-3): 271-302 (2015).

[7] M. Kaytoue, S.O. Kuznetsov, A. Napoli, S. Duplessis "Mining gene expression data with pattern structures in formal concept analysis." Information Sciences 181.10 (2011): 1989-2001.

[8] S.O. Kuznetsov, A fast algorithm for computing all intersections of objects from an arbitrary semilattice. Nauchno-Tekhnicheskaya Informatsiya Ser. 2 - Informatsionnye protsessy i sistemy, No. 1, pp.17-20, 1993.

[9] S.O. Kuznetsov, N. Makhazhanov, M. Ushakov: On Neural Network Architecture Based on Concept Lattices. Proc. ISMIS 2017: 653-663

[10] B. Schölkopf, A.J. Smola, F. Bach, Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2018.

[11] R. Campello, D. Moulavi, J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates. Advances in Knowledge Discovery and Data Mining. Vol. 7819. Springer (2013) pp. 160–172.

[12] Fei Hao, Stephen S Yau, Geyong Min, and Laurence T Yang. Detecting k-balanced trusted cliques in signed social networks. IEEE Internet Computing, 18(2):24–31, 2014.

[13] Min Tao, Fei Hao, Ling Wei, Huilai Zhi, Sergei O. Kuznetsov, Geyong Min, Fairness-aware Maximal Cliques Identification in Attributed Social Networks with Concept-cognitive Learning IEEE Transactions on Computational Social Systems (2024) doi: 10.1109/TCSS.2024.3445721