

Towards Concise Representation for Taxonomies of Epistemic Communities

Camille Roth^{1,2}, Sergei Obiedkov³, and Derrick G. Kourie³

¹ Department of Social and Cognitive Science,
University of Modena and Reggio Emilia, Italy

² Center of Research in Applied Epistemology, CNRS/Ecole Polytechnique,
Paris, France

`camille.roth@polytechnique.edu`

³ Department of Computer Science, University of Pretoria, Pretoria, South Africa
`sergei.obj@gmail.com, dkourie@cs.up.ac.za`

Abstract. We discuss an application of formal concept analysis to representing the structure of knowledge communities as a lattice-based taxonomy built upon groups of agents jointly manipulating some notions. The resulting structure is usually complex and uneasy to comprehend. We consider two approaches to build a concise representation hiding uninteresting information: a pruning strategy based on concept stability and a representational improvement based on nested line diagrams. We illustrate the method with a community of embryologists.

1 Introduction

A knowledge community is a group of agents who produce and exchange knowledge within a given knowledge field, achieving a widespread social cognition task in a rather decentralized, collectively interactive, and networked fashion. The study of such communities is usually a focal topic for social epistemology as well as scientometrics and political science [1–3], *inter alia*. A traditional concern relates to the description of the structure of knowledge communities [4] generally organized in several subcommunities. In contrast to the limited, subjective, and implicit representation that agents have of their own global community—a folk taxonomy [5]—epistemologists typically use expert-made taxonomies, more reliable but still falling short of precision, objectivity, and comprehensiveness.

We describe an application of formal concept analysis (FCA) aimed at representing the structure of a knowledge community under the form of a lattice-based taxonomy built upon groups of agents jointly manipulating some notions. This work is a development of the approach from [6, 7], which is briefly described in Sect. 2. In Sect. 3, we concentrate on how to make lattice-based taxonomies concise and intelligible. Concept lattices faithfully represent every trait in data, including those due to noise; we need tools that would allow us to abstract from insignificant traits. We propose a pruning technique based on concept stability [8] and apply it on its own and in combination with nested line diagrams [9], which allows representing the community structure at various levels of precision. These techniques admit modifications discussed in Sect. 4.

2 An FCA Approach in Applied Epistemology

2.1 Framework

In applied epistemology and scientometrics [2], taxonomies of knowledge communities are typically described with trees or two-dimensional maps of agents and topics. Various quantitative methods are being used, often based on categorization techniques and data describing links between authors, papers and/or notions—such as co-citation [4], co-authorship [10], or co-occurrence data [11]. In contrast, lattice-based taxonomies allow overlapping categories, with agents belonging to several fields at once. Our notion of community is both looser and more general than the sociological notion of structural equivalence [12]: we identify groups of agents linked jointly to various sets of notions instead of *exactly* the same notions.

A similar problem of identifying communities exists in the area of social networks. Lattices have also been used there [13–15], but in that context, groups of actors are generally considered to be disjoint and a lattice is only a first step in their construction. Besides, social aspects of the community structure are usually of prime interest, whereas we rather try to discover a structure of a scientific field without focusing on individuals. Because of this, social network lattices are typically based on data describing interactions between actors, while our data describes actors in terms of the domain for which we build a taxonomy.

Before proceeding, we briefly recall the FCA terminology [9]. Given a (*formal*) context $\mathbb{K} = (G, M, I)$, where G is called a set of *objects*, M is called a set of *attributes*, and the binary relation $I \subseteq G \times M$ specifies which objects have which attributes, the derivation operators $(\cdot)^I$ are defined for $A \subseteq G$ and $B \subseteq M$:

$$A^I = \{m \in M \mid \forall g \in A : gIm\}; \quad B^I = \{g \in G \mid \forall m \in B : gIm\}.$$

If this does not result in ambiguity, $(\cdot)'$ is used instead of $(\cdot)^I$. Sets A'' and B'' are said to be *closed*, as $(\cdot)''$ is a closure operator, i.e., it is extensive, idempotent, and monotonous.

A (*formal*) *concept* of the context (G, M, I) is a pair (A, B) , where $A \subseteq G$, $B \subseteq M$, $A = B'$, and $B = A'$. In this case, we also have $A = A''$ and $B = B''$. The set A is called the *extent* and B is called the *intent* of the concept (A, B) .

A concept (A, B) is a *subconcept* of (C, D) if $A \subseteq C$ (equivalently, $D \subseteq B$), then (C, D) is called a *superconcept* of (A, B) . We write $(A, B) \leq (C, D)$ and define the relations \geq , $<$, and $>$ as usual. If $(A, B) < (C, D)$ and there is no (E, F) such that $(A, B) < (E, F) < (C, D)$, then (A, B) is a *lower neighbor* of (C, D) and (C, D) is an *upper neighbor* of (A, B) ; notation: $(A, B) \prec (C, D)$.

The set of all concepts ordered by \leq forms a lattice, which is denoted by $\mathfrak{B}(\mathbb{K})$ and called the *concept lattice* of the context \mathbb{K} . The relation \prec defines edges in the *covering graph* of $\mathfrak{B}(\mathbb{K})$.

Epistemic Community Taxonomy. Our primary data consists of scientific papers dealing with a certain topic, from which we construct a set G of authors

and a set M of notions or terms. Thus, we have a context (G, M, I) , where gIm iff the author g uses the term m in one of his or her papers. Here, a concept intent is a subtopic and the corresponding extent is the set of all authors active in this subtopic. Thus, formal concepts provide a formalization of the notion of *epistemic community* (EC), traditionally defined as a group of agents dealing with a common set of issues and a common goal of knowledge creation [3]. By EC, we understand henceforth a field together with its authors, irrespective of their affiliation or personal interactions. The concept lattice represents the structure of a knowledge community as a taxonomy of ECs.

2.2 Empirical Example and Protocol

We focus on a bibliographical database of **MedLine** abstracts coming from the community working on the zebrafish in 1998–2003.⁴ We build a context describing which author used which notion during the whole period, where the notion set is made of a limited dictionary of about 70 lemmatized words selected by the expert [16] among the most frequent yet significant words of the community. Then, we extract a random sample context of 25 agents and 18 words, which we use to illustrate the techniques described in the paper. The lattice of this context is shown in Fig. 1 (only attribute labels are shown); it contains 69 concepts.⁵

According to the experts [16–18], three major subfields are distinguished. First, an important part of the community focuses on biochemical signaling mechanisms, involving pathways and receptors, which are crucial in understanding embryo growth processes. The second field includes comparative studies: the zebrafish may show similarities with other close vertebrate species, in particular, mice and humans. Another significant area of interest relates to the brain and the nervous system, notably in association with signaling in brain development.

3 Concise Representation

3.1 Rationale

Even if the concept lattice appears to adequately organize these communities, the number of ECs is still likely to be large. Although derived from a small context, the diagram in Fig. 1 is indeed rather complicated. This caveat affects concept lattices in general, which structure the data but do not reduce it much. To quote [9], “even carefully constructed line diagrams lose their readability from a certain size up”.

In this respect, some ECs are likely to be irrelevant to describe the taxonomy of knowledge fields. One solution is to compute only an upper part of the lattice (an order filter), e.g., concepts covering at least $n\%$ of authors (in this

⁴ Data is obtained from a query on article abstracts containing the term “*zebrafish*” at <http://www.pubmed.com>.

⁵ Diagrams are produced with ConExp (<http://sourceforge.net/projects/conexp>) and ToscanaJ (<http://sourceforge.net/projects/toscanaj>).

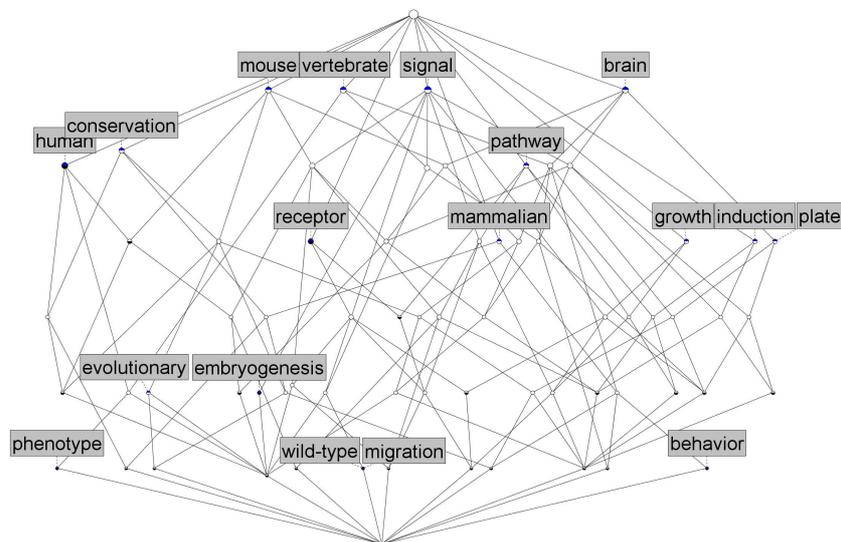


Fig. 1. The concept lattice of a sample zebrafish context.

case, we get an “iceberg lattice” [19]). However, to take into account small but interesting groups as well, typical for example of heterodox topics or new yet still minor trends, one should also compute all lower neighbors of “large” ECs (satisfying the $n\%$ threshold). Top-down lattice construction algorithms are particularly suitable for this approach [20]; alternatively, one may look at algorithms designed specifically for constructing iceberg lattices [19] and other algorithms from the frequent itemset mining community [21]. The reduction in the number of concepts can be considerable; however, though computationally feasible, this would still be unsatisfying from the standpoint of manual analysis.

Clearly, the size of the concept lattice is not only a computational problem. The lattice may contain nodes that are just too similar to each other because of noise in data or real minor differences yet irrelevant to our purposes. In this case, taking an upper part of the lattice does not solve the problem, since this part may well contain such similar nodes. Besides, it should also be interesting to distinguish major trends from minor subfields; that is, having a representation allowing for different levels of precision.

In this section, we consider two approaches to improve the readability of line diagrams: pruning and nesting. When pruning, we assume that some concepts are irrelevant: we filter out those which do not satisfy specified constraints of a certain kind. In a previous attempt to use concept lattices to represent EC taxonomies [6, 7], heuristics combining various criteria—such as extent size, shortest distance from top, number of lower neighbors, etc.—were used to score ECs and

keep only the n best ones. The resulting pictures were meaningful taxonomies but required a posteriori manual analysis, while it was unclear whether it could be possible to go further than a rough representation. Here, we focus on a particular pruning strategy based on the notion of stability of a concept [8].

Nested line diagrams [9], on the other hand, provide no reduction and, hence, do not incur any loss of information. Rather, they rearrange the concepts in such a way that the entire structure becomes more readable; they provide the user with a partial view, which can then be extended to a full view if so desired. Thus, nested line diagrams offer a useful technique for representing complex structures. Yet, because they preserve all details of the lattice, in order to get rid of (many) irrelevant details we combine nesting and pruning in Sect. 3.4. We thus try to get a representation that respects the original taxonomy while hiding at the same time uninteresting and superfluous information; our aim is a compromise between the noise level, the number of detail, and readability.

3.2 Stability-Based Pruning

Our structures are more complex than necessary, as our data is noisy: an author might use a term accidentally, or different terms denote the same thing. Many concepts do not correspond to real communities, and pruning seems unavoidable.

Our pruning technique is based on stability indices from [22, 8].

Definition 1. Let $\mathbb{K} = (G, M, I)$ be a formal context and (A, B) be a formal concept of \mathbb{K} . The stability index, σ , of (A, B) is defined as follows:

$$\sigma(A, B) = \frac{|\{C \subseteq A \mid C' = B\}|}{2^{|A|}}.$$

The stability index of a concept indicates how much the concept intent depends on particular objects of the extent. A stable intent is probably “real” even if the description of some objects is “noisy”. In application to our data, the stability index shows how likely we are to still observe a field if we ignore several authors. Apart from noise-resistance, a stable field does not disappear when a few members stop being active or switch to another topic. The following proposition describing the stability index of a concept (A, B) as a ratio between the number of subcontexts of \mathbb{K} where B is an intent and the total number of subcontexts of \mathbb{K} makes more explicit the idea behind stability:

Proposition 1. Let $\mathbb{K} = (G, M, I)$ be a formal context and (A, B) be a formal concept of \mathbb{K} . For a set $H \subseteq G$, let $I_H = I \cap (H \times M)$ and $\mathbb{K}_H = (H, M, I_H)$.

$$\sigma(A, B) = \frac{|\{\mathbb{K}_H \mid H \subseteq G \text{ and } B = B^{I_H I_H}\}|}{2^{|G|}}.$$

Proof. For $C \subseteq A$, define $\mathfrak{F}_C(\mathbb{K}) = \{\mathbb{K}_H \mid C \subseteq H \subseteq G \text{ and } A \cap H = C\}$. Sets $\mathfrak{F}_C(\mathbb{K})$ form a partition of subcontexts of \mathbb{K} (with the same M) and have the same size: $|\mathfrak{F}_C(\mathbb{K})| = 2^{|G|-|A|}$ ($C \subseteq A$). For $\mathbb{K}_H \in \mathfrak{F}_C(\mathbb{K})$, we have $B^{I_H I_H} = C^{I_H} = C'$; hence, B is closed in $\mathbb{K}_H \in \mathfrak{F}_C(\mathbb{K})$ if and only if $C' = B$. Therefore, $|\{\mathbb{K}_H \mid H \subseteq G \text{ and } B = B^{I_H I_H}\}| = \frac{2^{|G|} |\{C \subseteq A \mid C' = B\}|}{2^{|A|}}$. \square

In other words, the stability of a concept is the probability of preserving its intent after leaving out an arbitrary number of objects. This is the idea of cross-validation [23] carried to its extreme: stable intents are those generated by a large number of subsets of our data.

Computing Stability. In [8], it is shown that, given a formal context and its concept, the problem of computing the stability index of the concept is $\#P$ -complete. Below, we present a simple algorithm that takes the covering graph of a concept lattice $\mathfrak{B}(\mathbb{K})$ and computes the stability indices for every concept of the lattice. The algorithm is meant only as an illustration of a general strategy for computing the stability; therefore, we leave out any possible optimizations.

Algorithm ComputeStability

```

Concepts :=  $\mathfrak{B}(\mathbb{K})$ 
for each (A, B) in Concepts
  Count[(A, B)] := the number of lower neighbors of (A, B)
  Subsets[(A, B)] :=  $2^{|A|}$ 
end for
while Concepts is not empty
  let (C, D) be any concept from Concepts with Count[(C, D)] = 0
  Stability[(C, D)] := Subsets[(C, D)] /  $2^{|C|}$ 
  remove (C, D) from Concepts
  for each (A, B) > (C, D)
    Subsets[(A, B)] := Subsets[(A, B)] - Subsets[(C, D)]
    if (A, B) > (C, D)
      Count[(A, B)] := Count[(A, B)] - 1
    end if
  end for
end while
return Stability

```

To determine the stability index $\sigma(A, B)$, we compute the number of subsets $E \subseteq A$ that generate the intersection B (i.e., for which $E' = B$) and store it in **Subsets**. $\sigma(A, B)$ is simply the number of such subsets divided by the number of all subsets of A , that is, by $2^{|A|}$. Once computed, $\sigma(A, B)$ is stored in **Stability**, which is the output of the algorithm.

The algorithm traverses the covering graph from the bottom concept upwards. A concept is processed only after the stability indices of all its subconcepts have been computed; the **Count** variable is used to keep track of concepts that become eligible for processing. In the beginning of the algorithm, **Count**[(A, B)] is initialized to the number of lower neighbors of (A, B). When the stability index is computed for some lower neighbor of (A, B), we decrement **Count**[(A, B)]. By the time **Count**[(A, B)] reaches zero, we have computed the stability indices for all lower neighbors of (A, B) and, consequently, for all subconcepts of (A, B). Then, it is possible to determine the stability index of (A, B).

Initially, $\text{Subsets}[(A, B)]$ is set to the number of all subsets of A , that is, $2^{|A|}$. Before processing (A, B) we process all subconcepts (C, D) of (A, B) and decrement $\text{Subsets}[(A, B)]$ by the number of subsets of C generating the intersection D . By doing so, we actually subtract from $2^{|A|}$ the number of subsets of A which do not generate B : indeed, every subset of A generates either B or the intent of a subconcept of (A, B) . Thus, the value of $\text{Subsets}[(A, B)]$ eventually becomes equal to the number of subsets of A generating B .

Applying Stability. The basic stability-driven pruning method is to remove all concepts with stability below a fixed threshold. We computed the stability for concepts of our example from Fig. 1. There are 17 intents with stability above 0.5. Accidentally, they are closed under intersection and form a lattice shown in Fig. 2.

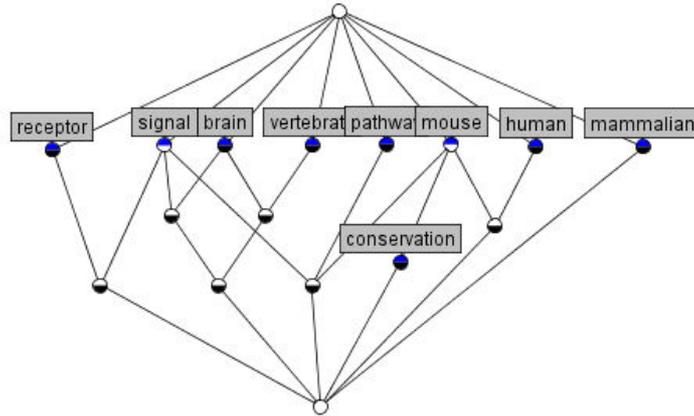


Fig. 2. The pruned lattice of Fig. 1, with stability threshold 0.52.

Stable concepts do not always form a lattice, which may or may not be a problem. If all we need is a directly observable taxonomy of scientific fields, there seem to be no reason to ask for a lattice. Sometimes we may want to have a lattice to be able to apply lattice-based analysis. This issue is beyond the scope of the present paper, but we suggest possible strategies in Sect. 4.2.

Figure 2 presents a more readable epistemic taxonomy representation, displaying the major fields of the community along with some meaningful joint communities (such as “*mouse, human*” and “*signal, receptor*”). However, some less important communities, like “*mouse, conservation*”, “*mammalian*” or “*signal, pathway, mouse*”, are also shown, and it cannot be seen from the picture that they are less important. Raising the stability threshold would eliminate these communities, but we would rather prefer having a multi-level represen-

tation with these communities rendered at a deeper level. Something similar applies, e.g., to the community “*signal, receptor, growth, pathway*”, which is missing from Fig. 2, but is interesting according to the expert-based description of the field (see Sect. 2.2). In this respect, nested line diagrams should provide a handy representation by distinguishing various levels of importance of notions.

3.3 Nested Line Diagrams

Nested line diagrams are a well-established tool in formal concept analysis that makes it possible to distribute representation details across several levels [9]. The main idea is to divide the attribute set of the context into two (or more) parts, construct the concept lattices for generated subcontexts, and draw the diagram of one lattice inside each node of the other lattice. In the case of two parts, an inner concept (A, B) enclosed within an outer concept (C, D) corresponds to a pair $(A \cap C, B \cup D)$. Not every such pair is a concept of the original context. Only inner nodes that correspond to concepts are represented by circles; such nodes are said to be “realized”. The outer diagram structures the data along one attribute subset, while the diagram inside an outer concept describes its structure in terms of the remaining attributes. For more details, see [9].

Partitioning the Attribute Set. The first step in constructing a nested line diagram is to split the attribute set into several parts. These parts do not have to be disjoint, but they will be in our case; hence, we are looking for a partition of the attribute set. As we seek to improve readability, we should display foremost the most significant attributes; therefore, we should assign major notions to higher levels, while leaving minor distinctions for lower levels. To this end, the words should be partitioned according to a “preference function”, which could range from the simplest (e.g., word frequency within the corpus) to more complicated designs.

One could consider a minimal set of notions covering all authors, i.e., find an irredundant cover set, as words from such a set could be expected to play a key role in describing the community. In practice, we use the algorithm from [24]. We apply it iteratively: the first subcontext contains notions forming an irredundant cover set for the whole author set; the second subcontext includes notions not occurring in the first subcontext, while covering the set of authors excluding those that use only notions from the first subcontext, etc. The last level contains the remaining notions. Denoting by $\mathcal{IC}(\mathbb{K})$ an irredundant cover set of a context \mathbb{K} , we start with the context $\mathbb{K}_0 = (G_0, M_0, I_0)$ and, for $k > 0$, recursively define the context $\mathbb{K}_k = (G_k, M_k, I_k)$, where $M_k = M_{k-1} \setminus \mathcal{IC}(\mathbb{K}_{k-1})$, $G_k = \bigcup_{m \in M_k} \{m\}'$, and $I_k = I \cap (G_k \times M_k)$. The sequence $(\mathcal{IC}(\mathbb{K}_0), \mathcal{IC}(\mathbb{K}_1), \dots, \mathcal{IC}(\mathbb{K}_n), M_0 \setminus M_n)$ for some $n > 0$ defines a partition of the attribute set M_0 to be used for nesting.

In our example, “*receptor*”, “*growth*”, “*signal*”, “*brain*”, “*mouse*”, and “*human*” cover the whole set of authors and constitute the outer-level subcontext notions. As we use only two levels, the inner-level subcontext is made of the remaining terms.

The resulting diagrams are shown in Fig. 3. Yet, while nesting makes it possible to distinguish between various levels of precision, both the outer and inner diagrams are still too large and recall the jumbled picture of Fig. 1. Stability-based pruning will address this problem; combining both procedures should yield a concise hierarchical representation.

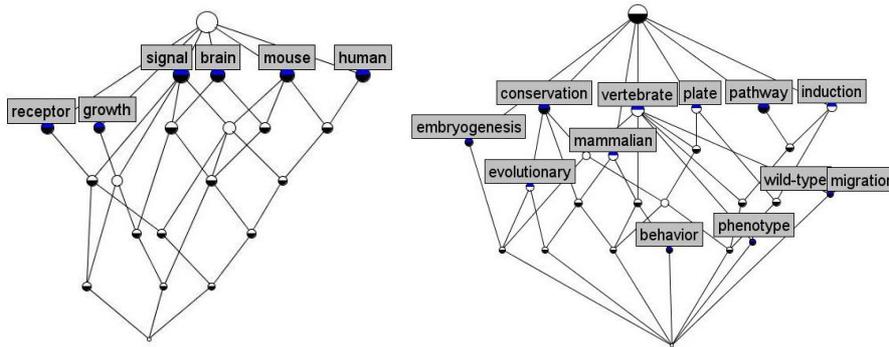


Fig. 3. Outer and inner diagrams for the nested line diagram of the zebrafish context

3.4 Combining Nesting and Stability-Based Pruning

After partitioning the set of words and building lattices for individual parts, we prune each lattice using the stability criterion. We can use different thresholds for different parts depending on the number of concepts we are comfortable to work with. For our example, we get the two diagrams shown in Fig. 4. Many attributes of the inner diagram are not shown in the picture, as they are not contained in any stable intent.

We proceed by drawing one diagram inside the other and interpret the picture as usual. Again, only inner nodes corresponding to concepts of the full context are represented by circles (and said to be “realized”). Figure 5 shows the resulting structure for our context.

This approach may also help in reducing the computational complexity. Generally, computing inner concepts is the same as computing the lattice for the whole context, but, combining nesting and pruning, we compute inner nodes only for relevant (that is, non-pruned) outer nodes.

Let us denote by $\mathfrak{B}_p(\mathbb{K})$ the set of concepts of \mathbb{K} satisfying the chosen pruning criteria and ordered in the usual way (one may regard p as an indicator of a specific pruning strategy). Assume that contexts $\mathbb{K}_1 = (G, M_1, I_1)$ and $\mathbb{K}_2 = (G, M_2, I_2)$ are subcontexts of $\mathbb{K} = (G, M, I)$ such that $M = M_1 \cup M_2$ and $I = I_1 \cup I_2$. We define the set of concepts corresponding to nodes of the nested

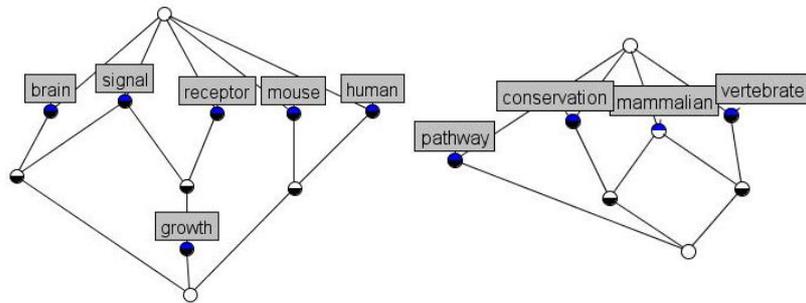


Fig. 4. The pruned outer and inner lattices from Fig. 3 (resp. thresholds 0.70 and 0.54)

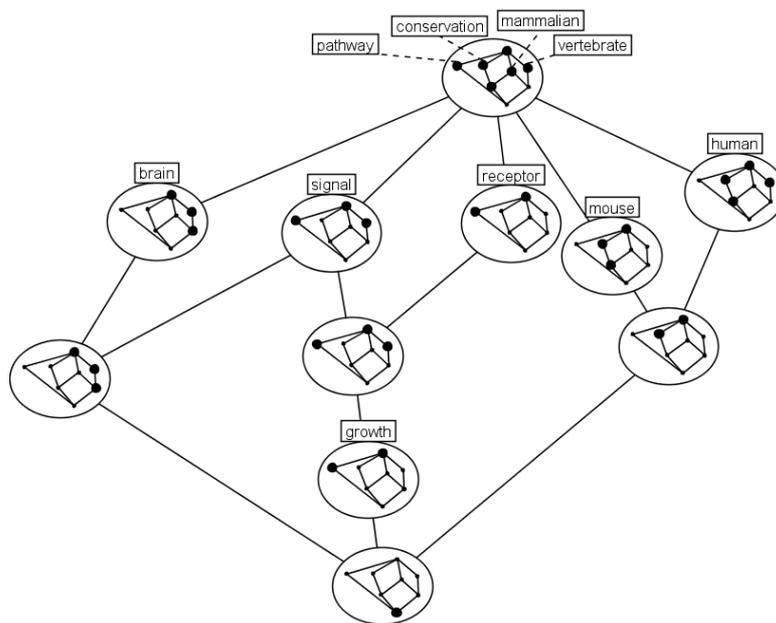


Fig. 5. Nested line diagram of pruned lattices from Fig. 4

line diagram of the pruned concept sets $\underline{\mathfrak{B}}_p(\mathbb{K}_1)$ and $\underline{\mathfrak{B}}_p(\mathbb{K}_2)$:

$$\underline{\mathfrak{B}}_p(G, M_1, M_2, I) = \{(A, B) \in \underline{\mathfrak{B}}(\mathbb{K}) \mid \forall i \in \{1, 2\} : ((B \cap M_i)', B \cap M_i) \in \underline{\mathfrak{B}}_p(\mathbb{K}_i)\}.$$

Proposition 2. *If $\underline{\mathfrak{B}}_p(\mathbb{K}_1)$ and $\underline{\mathfrak{B}}_p(\mathbb{K}_2)$ are \vee -subsemilattices of $\underline{\mathfrak{B}}(\mathbb{K}_1)$ and $\underline{\mathfrak{B}}(\mathbb{K}_2)$, respectively, then $\underline{\mathfrak{B}}_p(G, M_1, M_2, I)$ is a \vee -subsemilattice of $\underline{\mathfrak{B}}(\mathbb{K})$ and the map*

$$(A, B) \mapsto (((B \cap M_1)', B \cap M_1), ((B \cap M_2)', B \cap M_2))$$

is a \vee -preserving order embedding of $\underline{\mathfrak{B}}_p(G, M_1, M_2, I)$ in the direct product of $\underline{\mathfrak{B}}_p(\mathbb{K}_1)$ and $\underline{\mathfrak{B}}_p(\mathbb{K}_2)$.

We omit the proof due to space restrictions.

Unlike in standard nesting [9], the component maps $(A, B) \mapsto ((B \cap M_i)', B \cap M_i)$ are not necessarily surjective on $\underline{\mathfrak{B}}(\mathbb{K}_i)$. Hence, some outer nodes in our nested line diagram may be empty, i.e., contain no realized inner nodes, and some nodes of the inner diagram may never be realized.

Back to our example, the pruned outer diagram embraces most of the expert-based description outlined in Sect. 2.2 within a readable structure: it shows a joint focus on “human” and “mouse” (comparative studies); features several subfields made of “signal”, “receptor”, and “growth”; and displays brain studies, also in connection with signaling issues. The nested line diagram allows a deeper insight into the substructure of particular fields embedded within the pruned outer diagram. One may notice that outer nodes involving “human” and “mouse” show “vertebrate”, “mammalian”, and “conservation” in their inner diagrams, while outer nodes involving “signal” and “receptor” display “pathway”, which is consistent with the real state of affairs.

4 Further Work

4.1 Variants of Stability

The index in [8] refers to the stability of an intent; we may call it *intensional*. The *extensional stability index* of a concept (A, B) can be defined similarly:

$$\sigma_e(A, B) = \frac{|\{D \subseteq B \mid D' = A\}|}{2^{|B|}}.$$

The extensional stability shows if a concept extent is preserved after dropping some attributes; a proposition similar to Proposition 1 holds. Extensional stability relates to the social aspect of the network, measuring how much the community as a group of people depends on a particular topic. It allows one to fight noisy words: a community based on a noisy word will be extensionally unstable.

Proposition 1 suggests how the *general stability index* of a concept (A, B) should be defined—as the ratio between the number of subcontexts of $\mathbb{K} = (G, M, I)$ preserving the concept up to the omitted objects and attributes and the total number of subcontexts, or, more formally, for $A_H = A \cap H$ and $B_N = B \cap N$:

$$\frac{|\{(H, N, J) \mid H \subseteq G, N \subseteq M, J = I \cap (H \times N), A_H^J = B_N, B_N^J = A_H\}|}{2^{|G|+|M|}}.$$

As of now, we are not aware of any realistic method for computing the general stability; thus, it is only of theoretical interest. On the other hand, limited versions of stability (e.g., computed over subsets of a certain size only) and combinations of extensional and intensional stability are worth trying.

4.2 Strategies for Pruning

Other techniques aiming at reducing the number of concepts should be tested and, perhaps, some of them can be combined with stability for better results—notably pruning based on monotonous criteria like extent/intent size. Another method is given by attribute-dependency formulas [25], involving an expert-specified hierarchy on the attribute set (e.g., “*human*” and “*mouse*” are subtypes of “*vertebrate*”).

As noticed in Sect. 3.2, pruning may not necessarily yield a lattice. We can handle this situation in several ways: for example, enlarging the resulting structure by including all intersections of stable intents or reducing it by eliminating some stable intents. Regarding the latter option, we may prefer to merge an unstable concept (A, B) with its subconcept (C, D) rather than simply drop (A, B) . It is not immediately clear how to choose (C, D) —only that it should be somehow close to (A, B) . Merging can be done by assuming that all objects from A have all attributes from D and replacing the context relation I by $I \cup A \times D$ (cf. [26]). However, the modified context may have intents that are absent from the initial context, which is probably undesirable. Alternatively, one could add $B \rightarrow D$ to the implication system of the context. The lattice of attribute subsets generated by this augmented implication system will be different from the original lattice only in that B and possibly some of its previously closed supersets are not in the new lattice.

A different approach would involve merging based on partial implications (or association rules): compute all partial implications for the given confidence threshold and add them to the implication system of the context. It is a matter of further experiments to see which strategies are suitable for our goals.

4.3 Nesting

Nested line diagrams are not limited to two levels, although it still has to be investigated whether multi-level diagrams remain readable and interpretable. Various techniques for partitioning attribute sets should be explored. One strategy specific to our application is to partition words according to their type: as a verb, noun, adjective; or as a method, object, property, etc. Nesting seems to have more potential if implemented in interactive software tools that allow the user to zoom in and out on particular communities. The fact that one need not compute everything at once provides an additional computational advantage.

4.4 Dynamic Monitoring

Modeling changes of the community structure is important to describe the evolution of fields. One approach is to establish a relation between community

structures corresponding to different time points, e.g., identifying cases when several communities have merged into one or a community has divided into sub-communities. FCA offers methods for comparing two lattices built from identical objects and/or attributes [27]. Yet their relevance is likely to be application-dependent, and they should be adapted for reduced lattice-based structures we work with. A more ambitious approach assumes that any elementary change in the database (in G , M , or I) should correspond to a change in the representation of communities, and it should always be possible to trace a change in the community structure to a sequence of elementary changes in the database.

5 Conclusion

In previous work [6, 7], it was shown how concept lattices can provide an alternative to single-mode characterizations of the community structure in knowledge-based social networks. However, lattice-based taxonomies tend to be huge and, therefore, hard to compute and analyze. The computational complexity is partly addressed by considering a random sample of authors, since our taxonomies are centered on knowledge fields rather than individuals. The interpretability of results requires a more serious effort. We have proposed a pruning method based on the concept stability [8]. We think that it does not simply reduce the concept lattice to a somewhat rougher structure, but also helps to fight noise in data, so that the resulting structure might even be more accurate in describing the knowledge community than the original lattice is.

This method can be applied to constituent parts of a nested line diagram to achieve an optimal relationship between the readability and the level of detail. This is beneficial from the viewpoint of computational complexity, too: it is easier to compute the lattices of subcontexts and then prune each of them individually than to deal with the entire context. This approach admits “lazy” computation: within an interactive software tool the user can choose which outer nodes to explore; inner diagrams corresponding to other outer nodes will not be computed.

We have illustrated the proposed techniques with a small example; wider experiments are certainly needed. Some possible research directions are summarized in Sect. 4. Thus, this paper is only a step towards a consistent methodology for creating concise knowledge taxonomies based on concept lattices.

References

1. Schmitt, F., ed.: *Socializing Epistemology: The Social Dimensions of Knowledge*. Lanham, MD: Rowman & Littlefield (1995)
2. Leydesdorff, L.: In search of epistemic networks. *Social Studies of Science* **21** (1991) 75–110
3. Haas, P.: Introduction: epistemic communities and international policy coordination. *International Organization* **46**(1) (1992) 1–35
4. McCain, K.W.: Cocited author mapping as a valid representation of intellectual structure. *Journal of the American Society for Information Science* **37**(3) (1986) 111–122

5. Atran, S.: Folk biology and the anthropology of science: Cognitive universals and cognitive particulars. *Behavioral and Brain Sciences* **21** (1998) 547–609
6. Roth, C., Bourguine, P.: Epistemic communities: Description and hierarchic categorization. *Mathematical Population Studies* **12**(2) (2005) 107–130
7. Roth, C., Bourguine, P.: Lattice-based dynamic and overlapping taxonomies: the case of epistemic communities. *Scientometrics* **69**(2) (2006)
8. Kuznetsov, S.O.: On stability of a formal concept. In SanJuan, E., ed.: *JIM*, Metz, France (2003)
9. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin (1999)
10. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *PNAS* **99** (2002) 7821–7826
11. Noyons, E.C.M., van Raan, A.F.J.: Monitoring scientific developments from a dynamic perspective: self-organized structuring to map neural network research. *Journal of the American Society for Information Science* **49**(1) (1998) 68–81
12. Lorrain, F., White, H.C.: Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology* **1**(49–80) (1971)
13. White, D.R., Duquenne, V.: Social network & discrete structure analysis: Introduction to a special issue. *Social Networks* **18** (1996) 169–172
14. Freeman, L.: Cliques, Galois lattices, and the structure of human social groups. *Social Networks* **18** (1996) 173–187
15. Falzon, L.: Determining groups from the clique structure in large social networks. *Social Networks* **22** (2000) 159–172
16. Peyri ras, N. Personal communication. (2005)
17. Grunwald, D.J., Eisen, J.S.: Headwaters of the zebrafish – emergence of a new model vertebrate. *Nature Rev. Genetics* **3**(9) (2002) 717–724
18. Bradbury, J.: Small fish, big science. *PLoS Biology* **2**(5) (2004) 568–572
19. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with TITANIC. *Data and Knowledge Engineering* **42** (2002) 189–222
20. Kuznetsov, S.O., Obiedkov, S.: Comparing performance of algorithms for generating concept lattices. *J. Expt. Theor. Artif. Intell.* **14**(2/3) (2002) 189–216
21. Bayardo, Jr., R., Goethals, B., Zaki, M., eds.: *Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2004)*, CEUR-WS.org (2004)
22. Kuznetsov, S.O.: Stability as an estimate of the degree of substantiation of hypotheses derived on the basis of operational similarity. *Nauchn. Tekh. Inf., Ser.2 (Automat. Document. Math. Linguist.)* (12) (1990) 21–29
23. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *IJCAI*. (1995) 1137–1145
24. Batni, R.P., Russell, J.D., Kime, C.R.: An efficient algorithm for finding an irredundant set cover. *Journal of the Association for Computing Machinery* **21**(3) (1974) 351–355
25. Belohl vek, R., Sklenar, V.: Formal concept analysis constrained by attribute-dependency formulas. In Ganter, B., Godin, R., eds.: *ICFCA 2005*. Volume 3403 of *LNAI*. (2005) 176–191
26. Rome, J.E., Haralick, R.M.: Towards a formal concept analysis approach to exploring communities on the world wide web. In Ganter, B., Godin, R., eds.: *ICFCA 2005*. Volume 3403 of *LNAI*. (2005) 33–48
27. Wille, R.: Conceptual structures of multicontexts. In Eklund, P., Ellis, G., Mann, G., eds.: *Conceptual Structures: Knowledge Representation as Interlingua*. Volume 1115 of *LNAI*, Heidelberg-Berlin-New York, Springer (1996) 23–29