

# Quantitative Concept Analysis

Dusko Pavlovic

Royal Holloway, University of London, and University of Twente

Email: dusko.pavlovic@rhul.ac.uk

## Abstract

Formal Concept Analysis (FCA) begins from a context, given as a binary relation between some objects and some attributes, and derives a lattice of concepts, where each concept is given as a set of objects and a set of attributes, such that the first set consists of all objects that satisfy all attributes in the second, and vice versa. Many applications, though, provide contexts with quantitative information, telling not just whether an object satisfies an attribute, but also quantifying this satisfaction. Contexts in this form arise as rating matrices in recommender systems, as occurrence matrices in text analysis, as pixel intensity matrices in digital image processing, etc. Such applications have attracted a lot of attention, and several numeric extensions of FCA have been proposed. We propose the framework of *proximity sets* (*proxets*), which subsume partially ordered sets (posets) as well as metric spaces. One feature of this approach is that it extracts from quantified contexts quantified concepts, and thus allows full use of the available information. Another feature is that the categorical approach allows analyzing any universal properties that the classical FCA and the new versions may have, and thus provides structural guidance for aligning and combining the approaches.

## 1 Introduction

Suppose that the users  $U = \{\text{Abby, Dusko, Stef, Temra, Luka}\}$  provide the following star ratings for the items  $J = \{\text{"Nemo", "Crash", "Ikiru", "Bladerunner"}\}$

	"Nemo"	"Crash"	"Ikiru"	"Bladerunner"
Abby	★ ★ ★ ★	★ ★ ★ ★ ★	★ ★	★ ★ ★ ★
Dusko	★ ★	★ ★	★ ★ ★ ★	★ ★ ★ ★ ★
Stef	★ ★	★ ★ ★ ★ ★	★ ★ ★	★ ★
Temra	★	★ ★ ★	★ ★ ★	★ ★ ★ ★
Luka	★ ★ ★ ★ ★	★	★	★ ★

This matrix  $\Phi = (\Phi_{ju})_{J \times U}$  contains some information about the relations between these users' tastes, and about the relations between the styles of the items (in this case movies) that they rated. The task of data analysis is to extract that information. In particular, given a *context* matrix  $\Phi : J \times U \rightarrow R$  like in the above table, the task of *concept* analysis is to detect, on one hand, the latent concepts of *taste*, shared by some of the users in  $U$ , and on the other hand the latent concepts of *style*, shared by some of the items in  $J$ . In Formal Concept Analysis (FCA) [35, 15, 9, 14, 33], the latent concepts are expressed as sets: a taste  $t$  is a set of users, i.e. a map  $U \xrightarrow{t} \{0, 1\}$ , whereas a style  $s$  is a set of items, i.e. a map  $J \xrightarrow{s} \{0, 1\}$ . We explore a slightly refined notion of concept, which tells not just whether two users (resp. two items) share the same taste (resp. style) or not, but it also quantifies the degree of *proximity* of their tastes (resp. styles). This is formalized by expressing a taste as a map  $U \xrightarrow{\tau} [0, 1]$ , and a style as a map  $J \xrightarrow{\sigma} [0, 1]$ . The value  $\tau_u$  is thus a number from the interval  $[0, 1]$ , telling how close is the taste  $\tau$  to the user  $u$ ; whereas the value  $\sigma_i$  tells how close is the item  $i$  to the style  $\sigma$ . These concepts are *latent*, in the sense that they are not given in advance, but mined from the context matrix, just like in FCA, and similarly like in Latent Semantic Analysis (LSA) [10]. Although the extracted concepts are interpreted differently for the users in  $U$  and for the items in  $J$  (i.e. as the tastes and the styles, respectively) it turns out that the two obtained concept structures are isomorphic, just like in FCA and LSA. However, our approach allows initializing a concept analysis session by some prior concept structures, which allow

building upon the results of previous analyses, from other data sets, or specified by the analyst. This allows introducing different conceptual backgrounds for the users in  $U$  and for the items in  $J$ .

**Related work and background.** The task of capturing quantitative data in FCA was recognized early on. The simplest approach is to preprocess any given numeric data into relational contexts by introducing thresholds, and then apply the standard FCA method [13, 15]. This basic approach has been extended in several directions, e.g. Triadic Concept Analysis [17, 18, 26] and Pattern Structures [12, 19, 20], and refined for many application domains. A different way to introduce numeric data into FCA is to allow *fuzzy* contexts, as binary relations evaluated in an abstract lattice of truth values  $L$ . The different ways to lift the FCA constructions along the inclusion  $\{0, 1\} \hookrightarrow L$  have led to an entire gamut of different versions of fuzzy FCA [3, 4, 7, 8, 23], surveyed in [5]. With one notable exception, all versions of fuzzy FCA input quantitative data in the form as fuzzy relations, and output qualitative concept lattices in the standard form. The fact that numeric input data are reduced to the usual lattice outputs can be viewed as an advantage, since the outputs can then be presented, and interpreted, using the available FCA visualization tools and methods. On the other hand, only a limited amount of information contained in a numeric data set can be effectively captured in lattice displays. The practices of spectral methods of concept analysis [1, 10, 22], pervasive in web commerce, show that the quantitative information received in the input contexts can often be preserved in the output concepts, and effectively used in ongoing analyses. Our work has been motivated by the idea that suitably refined FCA constructions could output concept structures with useful quantitative information, akin to the concept eigenspaces of LSA. It turns out that the steps towards quantitative concepts on the FCA side have previously been made by Bělohlávek in [4], where fuzzy concept lattices derived from fuzzy contexts were proposed and analyzed. This is the mentioned notable exception from the other fuzzy and quantitative approaches to FCA, which all derive just qualitative concept lattices from quantitative contexts. Bělohlávek’s basic definitions turn out to be remarkably close to the definitions we start from in the present paper, in spite of the fact that his goal is to generalize FCA using carefully chosen fuzzy structures, whereas we use enriched categories with the ultimate goal to align FCA with the spectral methods for concept analysis, such as LSA. Does this confirm that the structures obtained in both cases naturally arise from the shared FCA foundations, rather than from either the fuzzy or the categorical approach? The ensuing analyses, however, shed light on these structures from essentially different angles, and open up complementary views: while Bělohlávek provides a detailed analysis of the internal structure of fuzzy concept lattices, we provide a high level view of their universal properties, from which some internal features follow, and which offers guidance through the maze of the available structural choices. Combining the two methods seems to open interesting alleys for future work.

Our motivating example suggests that our goals might be related to those of [11], where an FCA approach to recommender systems was proposed. However, the authors of [11] use FCA to tackle the problem of *partial* information (the missing ratings) in recommender systems, and they abstract away the *quantitative* information (contained in the available ratings); whereas our goal is to capture this quantitative information, and we leave the problem of partial information aside for the moment.

**Outline of the paper.** In Sec. 2 we introduce *proximity sets* (*proxets*), the mathematical formalism supporting the proposed generalization of FCA. Some constructions and notations used throughout the paper are introduced in Sec. 2.2. Since proxets generalize posets, in Sec. 3 we introduce the corresponding generalizations of infimum and supremum, and spell out the basic completion constructions, and the main properties of the infimum (resp. supremum) preserving morphisms. In Sec. 4, we study context matrices over proximity sets, and describe their decomposition, with a universal property analogous to the Singular Value Decomposition of matrices in linear algebra. Restricting this decomposition from proxets to discrete posets (i.e. sets) yields FCA. The drawback of this quantitative version of FCA is that in it a finite context generally allows an infinite proxet of concepts, whereas in the standard version of FCA, of course, finite contexts lead to finite concept lattices. This problem is tackled in Sec. 5, where we show how the users and the items, as related in the context, induce a finite generating set of concepts. Sec. 6 provides a discussion of the obtained results and ideas for the future work.

## 2 Proxets

### 2.1 Definition, intuition, examples

**Notation.** Throughout the paper, the order and lattice structure of the interval  $[0, 1]$  are denoted by  $\leq$ ,  $\wedge$  and  $\vee$ , whereas  $\cdot$  denotes the multiplication in it.

**Definition 2.1** A proximity over a set  $A$  is a map  $(\vdash) : A \times A \rightarrow [0, 1]$  which is

- *reflexive*:  $(x \vdash x) = 1$ ,
- *transitive*:  $(x \vdash y) \cdot (y \vdash z) \leq (x \vdash z)$ , and
- *antisymmetric*:  $(x \vdash y) = 1 = (y \vdash x) \implies x = y$

If only reflexivity and transitivity are satisfied, and not antisymmetry, then we have an *intensional proximity map*. The antisymmetry condition is sometimes called *extensionality*. A(n *intensional*) proximity set, or *proxet*, is a set equipped with a(n *intensional*) proximity map. A proximity (or monotone) morphism between the proxets  $A$  and  $B$  is a function  $f : A \rightarrow B$  such that all  $x, y \in A$  satisfy  $(x \vdash y)_A \leq (fx \vdash fy)_B$ . We denote by **Prox** the category of proxets and their morphisms.

**Categorical view.** A categorically minded reader can understand intensional proxets as categories enriched [21] over the poset  $[0, 1]$  viewed as a category, with the monoidal structure induced by the multiplication. In the presence of reflexivity and transitivity,  $(x \vdash y) = 1$  is equivalent with  $\forall z. (z \vdash x) \leq (z \vdash y)$ , and with  $\forall z. (x \vdash z) \geq (y \vdash z)$ . A proximity map is thus asymmetric if and only if  $(\forall z. (z \vdash x) = (z \vdash y)) \implies x = y$ , and if and only if  $(\forall z. (z \vdash x) = (z \vdash y)) \implies x = y$ . This means that extensional proxets correspond to *skeletal*  $[0, 1]$ -enriched categories.

#### 2.1.1 Examples.

The first example of a proxet is the interval  $[0, 1]$  itself, with the proximity

$$(x \vdash y)_{[0,1]} = \begin{cases} \frac{y}{x} & \text{if } y < x \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Note that  $(\vdash) : [0, 1] \times [0, 1] \rightarrow [0, 1]$  is now an *operation* on  $[0, 1]$ , satisfying

$$(x \cdot y) \leq z \iff x \leq (y \vdash z) \quad (2)$$

A wide family of examples follows from the fact that proximity sets (proxets) generalize partially ordered sets (posets), in the sense that any poset  $S$  can be viewed as a proxet  $WS$ , with the proximity induced by the partial ordering  $\sqsubseteq_S$  as follows:

$$(x \vdash y)_{WS} = \begin{cases} 1 & \text{if } x \sqsubseteq_S y \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The proxet  $WS$  is intensional if and only if  $S$  is just a *preorder*, in the sense that the relation  $\sqsubseteq_S$  is just transitive and reflexive. The other way around, any (intensional) proxet  $A$  induces two posets (resp. preorders),  $\Upsilon A$  and  $\Lambda A$ , with the same underlying set and

$$x \sqsubseteq_{\Upsilon A} y \iff (x \vdash y)_A = 1 \qquad x \sqsubseteq_{\Lambda A} y \iff (x \vdash y)_A > 0$$

Since the constructions  $W$ ,  $\Upsilon$  and  $\Lambda$ , extended on maps, preserve monotonicity, a categorically minded reader can easily confirm that we have three functors, which happen to form two adjunctions  $\Lambda \dashv W \dashv \Upsilon : \mathbf{Prox} \rightarrow \mathbf{Pos}$ . Since

$W : \text{Pos} \hookrightarrow \text{Prox}$  is an embedding,  $\text{Pos}$  is thus a reflective and coreflective subcategory of  $\text{Prox}$ . This means that  $\Lambda WS = S = \Upsilon WS$  holds for every poset  $S$ , so that posets are exactly the proxets where the proximities are evaluated only in 0 or 1; and that  $\Lambda A$  and  $\Upsilon A$  are respectively the initial and the final poset induced by the proxet  $A$ , as witnessed by the obvious morphisms  $W\Upsilon A \rightarrow A \rightarrow W\Lambda A$ . The same universal properties extend to a correspondence between intensional proxets and preorders.

A different family of examples is induced by metric spaces: any metric space  $X$  with a distance map  $d : X \times X \rightarrow [0, \infty]$  can be viewed as a proxet with the proximity map

$$(x \vdash y) = 2^{-d(x,y)} \quad (4)$$

Proxets are thus a common generalization of posets and metric spaces. But the usual metric distances are symmetric, i.e. satisfy  $d(x,y) = d(y,x)$ , whereas the proximities need not be. The inverse of (4) maps any proximity to a *quasi-metric*  $d(x,y) = -\log(x \vdash y)$  [36], whereas intensional proximities induce *pseudo-quasi-metrics* [37]. For a concrete family of examples of quasi-metrics, take any family of sets  $X \subseteq \wp X$ , and define

$$d(x,y) = |y \setminus x|$$

The distance of  $x$  and  $y$  is thus the number of elements of  $y$  that are not in  $x$ . This induces the proximity  $(x \vdash y) = 2^{-|y \setminus x|}$ . If  $X$  is a set of documents, viewed as bags (multisets) of terms, then both constructions can be generalized to count the difference in the numbers of the occurrences of terms in documents, and the set difference becomes multiset subtraction.

**Proximity or distance?** The isomorphism  $-\log x : [0, 1] \rightleftarrows [0, \infty] : 2^{-x}$  is easily seen to lift to an isomorphism between the category of proxets, as categories enriched over the multiplicative monoid  $[0, 1]$  and the category of generalized metric spaces, as categories enriched over the additive monoid  $[0, \infty]$ . Categorical studies of generalized metric spaces were initiated in [25], continued in denotational semantics of programming languages [34, 6, 24], and have recently turned out to be useful for quantitative distinctions in ecology [27]. The technical results of this paper could equivalently be stated in the framework of generalized metric spaces. While this would have an advantage of familiarity to certain communities, the geometric intuitions that come with metrics turn out to be misleading when imposed on the applications that are of interest here. The lifting of infima and suprema is fairly easy from posets to proxets, but leads to mysterious looking operations over metrics. In any case, the universal properties of matrix decompositions do not seem to have been studied in either framework so far.

## 2.2 Derived proxets and notations

Any proxets  $A, B$  give rise to other proxets by following standard constructions:

- the *dual* (or *opposite*) proxet  $\overline{A}$ , with the same underlying set and the proximity  $(x \vdash y)_{\overline{A}} = (y \vdash x)_A$ ;
- the *product* proxet  $A \times B$  over the cartesian product of the underlying sets, and the proximity  $(x, u \vdash y, v)_{A \times B} = (x \vdash y)_A \wedge (u \vdash v)_B$
- the *power* proxet  $B^A$  over the monotone maps, i.e.  $\text{Prox}(A, B)$  as the underlying set, with the proximity  $(f \vdash g)_{B^A} = \bigwedge_{x \in A} (fx \vdash gx)_B$ .

There are natural correspondences of proxet morphisms

$$\text{Prox}(A, B) \times \text{Prox}(A, C) \cong \text{Prox}(A, B \times C) \quad \text{and} \quad \text{Prox}(A \times B, C) \cong \text{Prox}(A, C^B)$$

**Notations.** In any proxet  $A$ , it is often convenient to abbreviate  $(x \vdash y)_A = 1$  to  $x \leq_A y$ . For  $f, g : A \rightarrow B$ , it is easy to see that  $f \leq_{B^A} g$  if and only if  $fx \leq_B gx$  for all  $x \in A$ .

### 3 Vectors, limits, adjunctions

#### 3.1 Upper and lower vectors

Having generalized posets to proxets, we proceed to lift the concepts of the least upper bound and the greatest lower bound. Let  $(S, \sqsubseteq)$  be a poset and let  $L, U \subseteq S$  be a lower set and an upper set, respectively, in the sense that

$$(x \sqsubseteq y \text{ and } y \in L) \Rightarrow x \in L \quad (x \in U \text{ and } x \sqsubseteq y) \Rightarrow y \in U$$

Then an element denoted  $\sqcup L$  is supremum of  $L$ , and  $\sqcap U$  is the infimum of  $U$ , if all  $x, y \in A$  satisfy

$$\sqcup L \leq y \iff \forall x. (x \in L \Rightarrow x \sqsubseteq y) \quad (5)$$

$$x \leq \sqcap U \iff \forall y. (y \in U \Rightarrow x \sqsubseteq y) \quad (6)$$

We generalize these definitions to proxet limits in (7-8). To generalize the lower sets, over which the suprema are taken, and the upper sets for infima, observe that any upper set  $U \subseteq S$  corresponds to a monotone map  $\vec{U} : S \rightarrow \{0, 1\}$ , whereas every lower set  $L$  corresponds to an antitone map  $\overleftarrow{L} : \overleftarrow{S} \rightarrow \{0, 1\}$ , where  $\overleftarrow{S}$  is the dual proxet defined in Sec. 2.2.

**Definition 3.1** An upper and a lower vector in a proxet  $A$  are the monotone maps  $\vec{v} : A \rightarrow [0, 1]$  and  $\overleftarrow{\lambda} : \overleftarrow{A} \rightarrow [0, 1]$ . The sets of vectors  $\uparrow A = [0, 1]^A$  and  $\downarrow A = [0, 1]^{\overleftarrow{A}}$  form proxets, with the proximities computed in terms of the infima in  $[0, 1]$ , as

$$(\vec{v} \vdash \vec{\tau})_{\uparrow A} = \bigwedge_{x \in A} (\vec{\tau}_x \vdash \vec{v}_x)_A \quad \text{and} \quad (\overleftarrow{\lambda} \vdash \overleftarrow{\mu})_{\downarrow A} = \bigwedge_{x \in A} (\overleftarrow{\lambda}_x \vdash \overleftarrow{\mu}_x)_A$$

**Remark.** Note that the defining condition for upper vectors  $(x \vdash y) \leq (\vec{v}_x \vdash \vec{v}_y)$ , and the defining condition for lower vectors  $(x \vdash y) \leq (\overleftarrow{\lambda}_x \vdash \overleftarrow{\lambda}_y)$  are respectively equivalent with

$$\vec{v}_x \cdot (x \vdash y) \leq \vec{v}_y \quad \text{and} \quad (x \vdash y) \cdot \overleftarrow{\lambda}_y \leq \overleftarrow{\lambda}_x$$

#### 3.2 Limits

**Definition 3.2** The upper limit or supremum  $\sqcup \overleftarrow{\lambda}$  of the lower vector  $\overleftarrow{\lambda}$  and the lower limit or infimum  $\sqcap \vec{v}$  of the upper vector  $\vec{v}$  are the elements of  $A$  that satisfy for every  $x, y \in A$

$$(\sqcup \overleftarrow{\lambda} \vdash y)_A = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \vdash y)_A \quad (7)$$

$$(x \vdash \sqcap \vec{v})_A = \bigwedge_{y \in A} \vec{v}_y \vdash (x \vdash y)_A \quad (8)$$

The proxet  $A$  is complete under infima (resp. suprema) if every upper (resp. lower) vector has an infimum (resp. supremum), which thus yield the operations  $\sqcap : \uparrow A \rightarrow A$  and  $\sqcup : \downarrow A \rightarrow A$

**Remarks.** Condition (7) generalizes (5), whereas (8) generalizes (6). Note how proximity operation  $\vdash$  over  $[0, 1]$ , defined in (1), plays in (7–8) the role that the implication  $\Rightarrow$  over  $\{0, 1\}$  played in (5–6). This is justified by the fact that  $\vdash$  is adjoint to the multiplication in  $[0, 1]$ , in the sense of (2), in the same sense in which  $\Rightarrow$  is adjoint to the meet in  $\{0, 1\}$ , or in any Heyting algebra, in the sense of  $(x \wedge y) \leq z \iff x \leq (y \Rightarrow z)$ .

An element  $w$  of a poset  $S$  is an upper bound of  $L \subseteq S$  if it satisfies just one direction of (5), i.e.

$$(w \sqsubseteq y) \implies \forall x. (x \in L \Rightarrow x \sqsubseteq y)$$

Ditto for the lower bounds. In a proxet  $A$ ,  $u$  is an upper bound of  $\overleftarrow{\lambda}$  and  $\ell$  is a lower bound of  $\overrightarrow{v}$  if all  $x, y \in A$  satisfy

$$(u \vdash y)_A \leq \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \vdash y)_A \quad \text{and} \quad (x \vdash \ell)_A \leq \bigwedge_{y \in A} \overrightarrow{v}_y \vdash (x \vdash y)_A$$

Using (2) and instantiating  $y$  to  $u$  in the first inequality, and  $x$  to  $\ell$  in the second one, these conditions can be shown to be equivalent with  $\overleftarrow{\lambda}_x \leq (x \vdash u)_A$  and  $\overrightarrow{v}_y \leq (\ell \vdash y)_A$ , which characterize the upper and the lower bounds in proxets.

### 3.3 Completions

Each element  $a$  of a proxet  $A$  induces two *representable* vectors

$$\begin{array}{ll} \Delta a : A \rightarrow [0, 1] & \nabla a : \overline{A} \rightarrow [0, 1] \\ x \mapsto (a \vdash x)_A & x \mapsto (x \vdash a)_A \end{array}$$

It is easy to see that these maps induce proximity morphisms  $\Delta : A \rightarrow \uparrow A$  and  $\nabla : A \rightarrow \downarrow A$ , which correspond to the categorical *Yoneda embeddings* [29, Sec. III.2]. They make  $\uparrow A$  into the lower completion, and  $\downarrow A$  into the upper completion of the proxet  $A$ .

**Proposition 3.3**  $\uparrow A$  is upper complete and  $\downarrow A$  is lower complete. Moreover, they are universal, in the sense that

- any monotone  $f : A \rightarrow C$  into a complete proxet  $C$  induces a unique  $\prod$ -preserving morphism  $f_{\#} : \uparrow A \rightarrow C$  such that  $f = f_{\#} \circ \Delta$ ;
- any monotone  $g : A \rightarrow D$  into a cocomplete proxet  $D$  induces a unique  $\coprod$ -preserving morphism  $g^{\#} : \downarrow A \rightarrow D$  such that  $g = g^{\#} \circ \nabla$ .

$$\begin{array}{ccc} & \uparrow A & \\ \Delta \nearrow & & \searrow \nabla \\ A & & \\ \forall f \searrow & & \nearrow \forall g \\ & \downarrow C & \\ & \exists! f_{\#} & \\ & \downarrow D & \\ & \exists! g^{\#} & \end{array}$$

### 3.4 Adjunctions

**Proposition 3.4** For any proximity morphism  $f : A \rightarrow B$  holds (a)  $\iff$  (b)  $\iff$  (c) and (d)  $\iff$  (e)  $\iff$  (f), where

- (a)  $f \left( \coprod \overleftarrow{\lambda} \right) = \coprod f \left( \overleftarrow{\lambda} \right)$
- (b)  $\exists f_* : B \rightarrow A \forall x \in A \forall y \in B. (fx \vdash y)_B = (x \vdash f_* y)_A$
- (c)  $\exists f_* : B \rightarrow A. \text{id}_A \leq f_* f \wedge f f_* \leq \text{id}_B$
- (d)  $f \left( \prod \overrightarrow{v} \right) = \prod f \left( \overrightarrow{v} \right)$
- (e)  $\exists f^* : B \rightarrow A \forall x \in A \forall y \in B. (f^* y \vdash x)_B = (y \vdash f x)_A$
- (f)  $\exists f^* : B \rightarrow A. f^* f \leq \text{id}_A \wedge \text{id}_B \leq f f^*$

The morphisms  $f^*$  and  $f_*$  are unique, whenever they exist.

**Definition 3.5** An upper adjoint is a proximity morphism satisfying (a-c) of Prop. 3.4; a lower adjoint satisfies (d-f). A (proximity) adjunction between proxets  $A$  and  $B$  is a pair of proximity morphisms  $f^* : A \rightleftarrows B : f_*$  related as in (b-c) and (e-f).

### 3.5 Projectors and nuclei

**Proposition 3.6** For any adjunction  $f^* : A \rightleftarrows B : f_*$  holds (a)  $\iff$  (b) and (c)  $\iff$  (d), where

$$(a) \quad \forall xy \in B. (f_*x \vdash f_*y)_A = (x \vdash y)_B$$

$$(b) \quad f^*f_* = \text{id}_B$$

$$(c) \quad \forall xy \in A. (f^*x \vdash f^*y)_B = (x \vdash y)_A$$

$$(d) \quad f_*f^* = \text{id}_A$$

**Definition 3.7** An adjunction satisfying (a-b) of Prop. 3.6 is an upper projector; an adjunction satisfying (c-d) is a lower projector. The upper (resp. lower) component of an upper (resp. lower) projector is called the upper (lower) projection. The other component (i.e. the one in (a), resp. (c)) is called the upper (lower) embedding.

**Proposition 3.8** Any upper (lower) adjoint factors, uniquely up to isomorphism, through an upper (lower) projection followed by an upper (lower) embedding through the proxet

$$\wr f \wr = \{ \langle x, y \rangle \in A \times B \mid f^*x = y \wedge x = f_*y \}$$

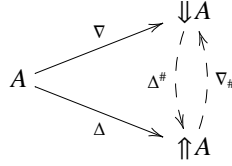
**Definition 3.9** A nucleus of the adjunction  $f^* : A \rightleftarrows B : f_*$  consists of a proxet  $\wr f \wr$  together with

- embeddings  $A \xleftarrow{e_*} \wr f \wr \xleftarrow{e^*} B$
- projections  $A \xrightarrow{p^*} \wr f \wr \xrightarrow{p_*} B$

such that  $f^* = e^*p^*$  and  $f_* = e_*p_*$ .

### 3.6 Cones and cuts

The cone operations are the proximity morphisms  $\Delta^\#$  and  $\nabla_\#$



These morphisms are induced by the universal properties of the Yoneda embeddings  $\nabla$  and  $\Delta$  as completions, stated in Prop. 3.3. Since by definition  $\Delta^\#$  preserves suprema, and  $\nabla_\#$  preserves infima, Prop. 3.4 implied that each of them is an adjoint, and it is not hard to see that they form the adjunction  $\Delta^\# : \Downarrow A \rightleftarrows \Uparrow A : \nabla_\#$ . Spelling them out yields

$$\left( \Delta^\# \overleftarrow{\lambda} \right)_a = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \vdash a) \qquad \left( \nabla_\# \overrightarrow{v} \right)_a = \bigwedge_{x \in A} \overrightarrow{v}_x \vdash (a \vdash x)$$

Intuitively,  $\left( \Delta^\# \overleftarrow{\lambda} \right)_a$  is the proximity of  $\overleftarrow{\lambda}$  to  $a$  as its upper bound, as discussed in Sec. 3.2. Visually,  $\left( \Delta^\# \overleftarrow{\lambda} \right)_a$  thus measures the cone from  $\overleftarrow{\lambda}$  to  $a$ , whereas  $\left( \nabla_\# \overrightarrow{v} \right)_a$  measures the cone from  $a$  to  $\overrightarrow{v}$ .

**Proposition 3.10** For every  $\overleftarrow{\lambda} \in \Downarrow A$  every  $\overrightarrow{v} \in \Uparrow A$  holds

$$\begin{aligned} \overleftarrow{\lambda} &\leq \nabla_\# \Delta^\# \overleftarrow{\lambda} & \text{and} & & \overleftarrow{\lambda} &\geq \nabla_\# \Delta^\# \overleftarrow{\lambda} \iff \exists \overrightarrow{v}. \overleftarrow{\lambda} = \Delta^\# \overrightarrow{v} \\ \overrightarrow{v} &\leq \Delta^\# \nabla_\# \overrightarrow{v} & \text{and} & & \overrightarrow{v} &\geq \Delta^\# \nabla_\# \overrightarrow{v} \iff \exists \overleftarrow{\lambda}. \overrightarrow{v} = \nabla_\# \overleftarrow{\lambda} \end{aligned}$$

The transpositions make the following subproxets isomorphic

$$\begin{aligned} (\Downarrow A)_{\nabla_{\#}\Delta^{\#}} &= \left\{ \overleftarrow{\lambda} \in \Downarrow A \mid \overleftarrow{\lambda} = \nabla_{\#}\Delta^{\#}\overleftarrow{\lambda} \right\} \\ (\Uparrow A)_{\Delta^{\#}\nabla_{\#}} &= \left\{ \overrightarrow{v} \in \Uparrow A \mid \overrightarrow{v} = \Delta^{\#}\nabla_{\#}\overrightarrow{v} \right\} \end{aligned}$$

**Definition 3.11** The vectors in  $(\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$  and  $(\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$  are called cones. The associated cones  $\overleftarrow{\gamma} \in (\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$  and  $\overrightarrow{\gamma} \in (\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$  such that  $\overleftarrow{\gamma} = \nabla_{\#}\overrightarrow{\gamma}$  and  $\overrightarrow{\gamma} = \Delta^{\#}\overleftarrow{\gamma}$  a cut  $\gamma = \langle \overleftarrow{\gamma}, \overrightarrow{\gamma} \rangle$  in proxet  $A$ . Cuts form a proxet  $\Downarrow A$ , isomorphic with  $(\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$  and  $(\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$ , with the proximity

$$(\gamma \vdash \varphi)_{\Downarrow A} = (\overleftarrow{\gamma} \vdash \overleftarrow{\varphi})_{\Downarrow A} = (\overrightarrow{\gamma} \vdash \overrightarrow{\varphi})_{\Uparrow A}$$

**Lemma 3.12** The  $\Downarrow A$ -infima are constructed in  $\Downarrow A$ , and  $\Uparrow A$  suprema are constructed in  $\Uparrow A$ .

**Corollary 3.13** A proxet  $A$  has all suprema if and only if it has all infima.

**Dedekind-MacNeille completion is a special case.** If  $A$  is a poset, viewed by (3) as the proxet  $WA$ , then  $\Downarrow WA$  is the Dedekind-MacNeille completion of  $A$  [28]. The above construction extends the Dedekind-MacNeille completion to the more general framework of proxets, in the sense that it satisfies in the universal property of the Dedekind-MacNeille completion [2]. The construction seems to be novel in the familiar frameworks of metric and quasi-metric spaces. However, Quantitative Concept Analysis requires that we lift this construction to matrices.

## 4 Proximity matrices and their decomposition

### 4.1 Definitions, connections

**Definition 4.1** A proximity matrix  $\Phi$  from proxet  $A$  to proxet  $B$  is a vector  $\Phi : \overline{A} \times B \rightarrow [0, 1]$ . We write it as  $\Phi : A \multimap B$ , and write its value  $\Phi(x, y)$  at  $x \in A$  and  $y \in B$  in the form  $(x \models y)_{\Phi}$ . The matrix composition of  $\Phi : A \multimap B$  and  $\Psi : B \multimap C$  is defined

$$(x \models z)_{(\Phi;\Psi)} = \bigvee_{y \in B} (x \models y)_{\Phi} \cdot (y \models z)_{\Psi}$$

With this composition and the identity matrices  $\text{Id}_A : A \times A \rightarrow [0, 1]$  where  $\text{Id}_A(x, x') = (x \vdash x')_A$ , proxets and proxet matrices form the category **Matr**.

**Remark.** Note that the defining condition  $(u \vdash x) \cdot (y \vdash v) \leq ((x \models y)_{\Phi} \vdash (u \models v)_{\Phi})$ , which says that  $\Phi$  is a proximity morphism  $\overline{A} \times B \rightarrow [0, 1]$ , can be equivalently written

$$(u \vdash x) \cdot (x \models y)_{\Phi} \cdot (y \vdash v) \leq (u \models v)_{\Phi} \tag{9}$$

**Definition 4.2** The dual  $\Phi^{\ddagger} : B \multimap A$  of a matrix  $\Phi : A \multimap B$  has the entries

$$(y \models x)_{\Phi^{\ddagger}} = \bigwedge_{\substack{u \in A \\ v \in B}} (u \models v)_{\Phi} \vdash ((u \vdash x)_A \cdot (y \vdash v)_B)$$

A matrix  $\Phi : A \multimap B$  where  $\Phi^{\ddagger\ddagger} = \Phi$  is called a suspension.



**Remarks.** It is easy to see by Prop. 3.10 that  $(x \vdash y)_\Phi \leq (x \vdash y)_{\Phi^{\ddagger\ddagger}}$  holds for all  $x \in A$  and  $y \in B$ , and that  $\Phi$  is a suspension if and only if there is some  $\Psi : B \multimap A$  such that  $\Phi = \Psi^{\ddagger}$ . It is easy to see that  $\Phi \leq \Psi \Rightarrow \Phi^{\ddagger} \geq \Psi^{\ddagger}$ , and thus  $\Phi \leq \Phi^{\ddagger\ddagger}$  implies  $\Phi^{\ddagger} = \Phi^{\ddagger\ddagger\ddagger}$ .

**Definition 4.3** The matrices  $\Phi : A \multimap B$  and  $\Psi : B \multimap A$  form a connection if  $\Phi ; \Psi \leq \text{Id}_A$  and  $\Psi ; \Phi \leq \text{Id}_B$ .

**Proposition 4.4**  $\Phi : A \multimap B$  and  $\Phi^{\ddagger} : B \multimap A$  always form a connection.

**Definition 4.5** A matrix  $\Phi : A \multimap B$  is embedding if  $\Phi ; \Phi^{\ddagger} = \text{Id}_A$ ; and a projection if  $\Phi^{\ddagger} ; \Phi = \text{Id}_B$ .

**Definition 4.6** A decomposition of a matrix  $\Phi : A \multimap B$  consists of a proxet  $D$ , with

- projection matrix  $P : A \multimap D$ , i.e.  $(d \vdash d')_D = \bigvee_{x \in A} (d \models x)_{P^*} \cdot (x \models d')_P$ ,
- embedding matrix  $E : D \multimap B$ , i.e.  $(d \vdash d')_D = \bigvee_{y \in B} (d \models y)_E \cdot (y \models d')_{E^*}$ ,

such that  $\Phi = P ; E$ , i.e.  $(x \models y)_\Phi = \bigvee_{d \in D} (x \models d)_P \cdot (d \models y)_E$ .

**Matrices as adjunctions.** A matrix  $\Phi : A \multimap B$  can be equivalently presented as either of the proximity morphisms  $\Phi_\bullet$  and  $\Phi^\bullet$ , which extend to  $\Phi_*$  and  $\Phi^*$  using Thm. 3.3

$$\frac{\overline{A} \times B \xrightarrow{\Phi} [0, 1]}{\frac{A \xrightarrow{\Phi_\bullet} \uparrow\uparrow B \quad B \xrightarrow{\Phi^\bullet} \downarrow\downarrow A}{\downarrow\downarrow A \xrightarrow{\Phi^*} \uparrow\uparrow B \quad \uparrow\uparrow B \xrightarrow{\Phi_*} \downarrow\downarrow A}} \quad \left( \Phi^* \overleftarrow{\lambda} \right)_b = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \models b)_\Phi \quad \left( \Phi_* \overrightarrow{v} \right)_a = \bigwedge_{y \in B} \overrightarrow{v}_y \vdash (a \models y)_\Phi \quad (10)$$

Both extensions, and their nucleus, are summarized in diagram (11).

The adjunction  $\Phi^* : \downarrow\downarrow A \rightleftarrows \uparrow\uparrow B : \Phi_*$  means that

$$\left( \Phi^* \overleftarrow{\lambda} \vdash \overrightarrow{v} \right)_{\uparrow\uparrow B} = \bigwedge_{y \in B} \overrightarrow{v}_y \vdash (\Phi^* \overleftarrow{\lambda})_y = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (\Phi_* \overrightarrow{v})_x = \left( \overleftarrow{\lambda} \vdash \Phi_* \overrightarrow{v} \right)_{\downarrow\downarrow A}$$

holds. The other way around, it can be shown that any adjunction between  $\downarrow\downarrow A$  and  $\uparrow\uparrow B$  is completely determined by the induced matrix from  $A$  to  $B$ .

**Proposition 4.7** The matrices  $\Phi \in \text{Matr}(A, B)$  are in a bijective correspondence with the adjunctions  $\Phi^* : \downarrow\downarrow A \rightleftarrows \uparrow\uparrow B : \Phi_*$ .

## 4.2 Matrix decomposition through nucleus

Prop. 3.10 readily lifts to matrices.

**Proposition 4.8** *For every  $\overleftarrow{\alpha} \in \Downarrow A$  every  $\overrightarrow{\beta} \in \Uparrow B$  holds*

$$\begin{aligned} \overleftarrow{\alpha} &\leq \Phi_* \Phi^* \overleftarrow{\alpha} & \text{and} & & \overleftarrow{\alpha} &\geq \Phi_* \Phi^* \overleftarrow{\alpha} \iff \exists \overrightarrow{\beta} \in \Uparrow B. \overleftarrow{\alpha} = \Phi^* \overrightarrow{\beta} \\ \overrightarrow{\beta} &\leq \Phi^* \Phi_* \overrightarrow{\beta} & \text{and} & & \overrightarrow{\beta} &\geq \Phi^* \Phi_* \overrightarrow{\beta} \iff \exists \overleftarrow{\alpha} \in \Downarrow A. \overrightarrow{\beta} = \Phi_* \overleftarrow{\alpha} \end{aligned}$$

The adjunction  $\Phi^* : A \rightleftarrows B : \Phi_*$  induces the isomorphisms between the following proxets

$$\begin{aligned} \wr \Phi \wr_A &= \{ \overleftarrow{\alpha} \in \Downarrow A \mid \overleftarrow{\alpha} = \Phi_* \Phi^* \overleftarrow{\alpha} \} \\ \wr \Phi \wr_B &= \{ \overrightarrow{\beta} \in \Uparrow B \mid \overrightarrow{\beta} = \Phi^* \Phi_* \overrightarrow{\beta} \} \\ \wr \Phi \wr &= \{ \gamma = \langle \overleftarrow{\gamma}, \overrightarrow{\gamma} \rangle \in \Downarrow A \times \Uparrow B \mid \overleftarrow{\gamma} = \Phi_* \overrightarrow{\gamma} \wedge \Phi^* \overleftarrow{\gamma} = \overrightarrow{\gamma} \} \end{aligned}$$

with the proximity

$$(\gamma \vdash \varphi)_{\wr \Phi \wr} = (\overleftarrow{\gamma} \vdash \overleftarrow{\varphi})_{\Downarrow A} = (\overrightarrow{\gamma} \vdash \overrightarrow{\varphi})_{\Uparrow B}$$

**Definition 4.9**  $\wr \Phi \wr$  is called the nucleus of the matrix  $\Phi$ . Its elements are the  $\Phi$ -cuts.

**Theorem 4.10** The matrix  $\Phi : A \multimap B$  decomposes through  $\wr \Phi \wr$  into

- the projection  $P^* : A \multimap \wr \Phi \wr$  with  $(x \models \langle \overleftarrow{\alpha}, \overrightarrow{\beta} \rangle)_{P^*} = \overleftarrow{\alpha}_x$ , and
- the embedding  $E^* : \wr \Phi \wr \multimap B$  with  $(\langle \overleftarrow{\alpha}, \overrightarrow{\beta} \rangle \models y)_{E^*} = \overrightarrow{\beta}_y$

## 4.3 Universal properties

Any proxet morphism  $f : A \rightarrow B$  induces two matrices,  $\Omega f : A \multimap B$  and  $\Upsilon f : B \multimap A$  with

$$(x \models y)_{\Omega f} = (f x \vdash y)_B \quad (\gamma \models x)_{\Upsilon f} = (y \vdash f x)_B$$

**Definition 4.11** A proximity matrix morphism from a matrix  $\Phi : F_0 \multimap F_1$  to  $\Gamma : G_0 \multimap G_1$  consists of pair of monotone maps  $h_0 : F_0 \rightarrow G_0$  and  $h_1 : F_1 \rightarrow G_1$  such that

- $\Omega h_0 ; \Gamma = \Phi ; \Omega h_1$ ,
- $h_0$  preserves any  $\sqcup$  that may exist in  $F_0$ ,
- $h_1$  preserves any  $\sqcap$  that may exist in  $F_1$ .

Let  $\mathbf{MMat}$  denote the category of proxet matrices and matrix morphisms. Let  $\mathbf{CMat}$  denote the full subcategory spanned by proximity matrices between complete proxets.

**Proposition 4.12**  $\mathbf{CMat}$  is reflective in  $\mathbf{MMat}$  along  $\wr - \wr : \mathbf{MMat} \rightleftarrows \mathbf{CMat} : U$

**Posets and FCA.** If  $A$  and  $B$  are posets, a  $\{0, 1\}$ -valued proxet matrix  $\Phi : A \leftrightarrow B$  can be viewed as a subposet  $\Phi \subseteq A \times B$ , lower closed in  $A$  and upper closed in  $B$ . The adjunction  $\Phi^* : A \rightleftarrows B : \Phi_*$  is the Galois connection induced by  $\Phi$ , and the posetal nucleus  $\wr \Phi \wr$  is now the complete lattice such that

- $A \xrightarrow{\nabla} \Downarrow A \rightarrow \wr \Phi \wr$  is  $\vee$ -generating and  $\wedge$ -preserving,
- $B \xrightarrow{\nabla} \Uparrow B \rightarrow \wr \Phi \wr$  is  $\wedge$ -generating and  $\vee$ -preserving.

When  $A$  and  $B$  are discrete posets, i.e. with all elements incomparable, then any binary relation  $R \subseteq A \times B$  can be viewed as a proxet matrix between them. Restricting to the vectors that take their values in 0 and 1 yields  $\Downarrow A \cong (\wp A, \subseteq)$  and  $\Uparrow B \cong (\wp B, \supseteq)$ . The concept lattice of FCA then arises from the Galois connection  $R^* : \Downarrow A \rightleftarrows \Uparrow B : R_*$  as the concept lattice  $\wr R \wr$ . Restricted to  $\{0, 1\}$ -valued matrices between discrete sets  $A$  and  $B$ , Prop. 4.12 thus yields a universal construction of a lattice  $\vee$ -generated by  $A$  and  $\wedge$ -generated by  $B$ . The FCA concept lattice derived from a context  $\Phi$  is thus its posetal nucleus  $\wr \Phi \wr$ . This universal property is closely related with the methods and results of [2, 16].

**Lifting The Basic Theorem of FCA.** The Basic Theorem of FCA says that every complete lattice can be realized as a concept lattice, namely the one induced by the context of its own partial order. For quantitative concept analysis, this is an immediate consequence of Prop. 4.12, which implies a proxet  $A$  is complete if and only if  $\text{Id}_A = \wr \text{Id}_A \wr$ . Intuitively, this just says that nucleus, as a completion, preserves the structure that it completes, and must therefore be idempotent, as familiar from the Dedekind-MacNeille construction. It should be noted that this property does not generalize beyond proxets.

## 5 Representable concepts and their proximities

### 5.1 Decomposition without completion

The problem with factoring matrices  $\Phi : A \leftrightarrow B$  through  $\wr \Phi \wr$  in practice is that  $\wr \Phi \wr$  is a large, always infinite structure. The proxet  $\wr \Phi \wr$  is the completion of the matrix  $\Phi : A \leftrightarrow B$  in the sense that it is

- the subproxet of the  $\coprod$ -completion  $\Downarrow A$  of  $A$ , spanned by the vectors  $\overleftarrow{\alpha} = \Phi_* \Phi^* \overleftarrow{\alpha}$ ,
- the subproxet of the  $\prod$ -completion  $\Uparrow B$  of  $B$ , spanned by the vectors  $\overrightarrow{\beta} = \Phi^* \Phi_* \overrightarrow{\beta}$ .

Since there are always uncountably many lower and upper vectors, and the completions  $\Downarrow A$  and  $\Uparrow B$  are infinite,  $\wr \Phi \wr$  follows suit. But can we extract a small set of generators of  $\wr \Phi \wr$ , still supporting a decomposition of the matrix  $\Phi$ .

**Definition 5.1** *The representable concepts induced  $\Phi$  are the elements of the completion  $\wr \Phi \wr$  induced the representable vectors, i.e.*

- lower representable concepts  $\nabla \Phi = \{ \langle \Phi_* \Phi^* \nabla a, \Phi^* \nabla a \rangle \mid a \in A \}$
- upper representable concepts  $\Delta \Phi = \{ \langle \Phi_* \Delta b, \Phi^* \Phi_* \Delta b \rangle \mid b \in B \}$
- representable concepts  $\diamond \Phi = \nabla \Phi \cup \Delta \Phi$

**Notation.** The elements of  $\diamond \Phi$  are written in the form  $\diamond x = \langle \overleftarrow{\diamond x}, \overrightarrow{\diamond x} \rangle$ , and thus

$$\begin{aligned} \overleftarrow{\diamond a} &= \Phi_* \Phi^* \nabla a & \overrightarrow{\diamond a} &= \Phi^* \nabla a \\ \overleftarrow{\diamond b} &= \Phi_* \Delta b & \overrightarrow{\diamond b} &= \Phi^* \Phi_* \Delta b \end{aligned}$$

**Theorem 5.2** For any proxet matrix  $\Phi : A \rightsquigarrow B$ , the restriction of the decomposition  $A \xrightarrow{P^*} \mathcal{I}(\Phi) \xrightarrow{E^*} B$  from Thm. 4.10 along the inclusion  $\diamond\Phi \hookrightarrow \mathcal{I}(\Phi)$  to the representable concepts yields a decomposition  $A \xrightarrow{P} \diamond\Phi \xrightarrow{E} B$  which still satisfies Def. 4.6. More precisely, the matrices

- $P : \overline{A} \times \diamond\Phi \hookrightarrow \overline{A} \times \mathcal{I}(\Phi) \xrightarrow{P^*} [0, 1]$
- $E : \overline{\diamond\Phi} \times B \hookrightarrow \overline{\mathcal{I}(\Phi)} \times B \xrightarrow{E^*} [0, 1]$

are such that  $P : A \rightsquigarrow \diamond\Phi$  is a projection,  $E : \diamond\Phi \rightsquigarrow B$  is an embedding, and  $P;E = \Phi$ .

## 5.2 Computing proximities of representable concepts

To apply these constructions to the ratings matrix from Sec. 1, we first express the star ratings as numbers between 0 and 1.

	$n$	$c$	$i$	$b$
$a$	$\frac{4}{5}$	1	$\frac{2}{5}$	$\frac{4}{5}$
$d$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1
$s$	$\frac{2}{5}$	1	$\frac{3}{5}$	$\frac{2}{5}$
$t$	$\frac{1}{5}$	$\frac{3}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
$l$	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$

where we also abbreviated the user names to  $U = \{A, D, S, T, L\}$  and the item names to  $J = \{n, c, i, b\}$ . Now we can compute the representable concepts  $\diamond\varphi \in \diamond\Phi$  according to Def. 5.1, using (10):

$$\begin{aligned}
 \overleftarrow{(\diamond j)}_u &= \left( \bigwedge_{\ell \in J} (\Delta j)_\ell \vdash (u \models \ell) \right) = (u \models j) & \overrightarrow{(\diamond j)}_k &= \left( \bigwedge_{x \in U} (\overrightarrow{\diamond j})_x \vdash (x \models k) \right) = \left( \bigwedge_{x \in U} (x \models j) \vdash (x \models k) \right) \\
 \overrightarrow{(\diamond u)}_j &= \left( \bigwedge_{x \in U} (\nabla u)_x \vdash (x \models j) \right) = (u \models j) & \overleftarrow{(\diamond u)}_v &= \left( \bigwedge_{\ell \in J} (\overleftarrow{\diamond u})_\ell \vdash (v \models \ell) \right) = \left( \bigwedge_{\ell \in J} (u \models \ell) \vdash (v \models \ell) \right)
 \end{aligned}$$

Since  $\overleftarrow{\diamond\varphi} = \Phi_* \overrightarrow{\diamond\varphi}$  and  $\Phi^* \overleftarrow{\diamond\varphi} = \overrightarrow{\diamond\varphi}$ , it suffices to compute one component of each pair  $\diamond\varphi = \langle \overleftarrow{\diamond\varphi}, \overrightarrow{\diamond\varphi} \rangle$ , say the first one. So we get

$$\begin{aligned}
 \overleftarrow{\diamond n} &= \left( \frac{4}{5} \quad \frac{2}{5} \quad \frac{2}{5} \quad \frac{1}{5} \quad 1 \right) & \overleftarrow{\diamond c} &= \left( 1 \quad \frac{2}{5} \quad 1 \quad \frac{3}{5} \quad \frac{1}{5} \right) & \overleftarrow{\diamond t} &= \left( \frac{2}{5} \quad \frac{4}{5} \quad \frac{3}{5} \quad \frac{3}{5} \quad \frac{1}{5} \right) \\
 \overleftarrow{\diamond b} &= \left( \frac{4}{5} \quad 1 \quad \frac{2}{5} \quad \frac{4}{5} \quad \frac{1}{5} \right) & \overleftarrow{\diamond a} &= \left( 1 \quad \frac{2}{5} \quad \frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{5} \right) & \overleftarrow{\diamond d} &= \left( \frac{1}{2} \quad 1 \quad \frac{2}{5} \quad \frac{1}{2} \quad \frac{1}{4} \right) \\
 \overleftarrow{\diamond s} &= \left( \frac{2}{5} \quad \frac{2}{5} \quad 1 \quad \frac{1}{2} \quad \frac{1}{5} \right) & \overleftarrow{\diamond i} &= \left( \frac{2}{5} \quad \frac{2}{5} \quad \frac{1}{2} \quad 1 \quad \frac{1}{5} \right) & \overleftarrow{\diamond l} &= \left( \frac{4}{5} \quad \frac{2}{5} \quad \frac{2}{5} \quad \frac{1}{5} \quad 1 \right)
 \end{aligned}$$

The proximities between all representable concepts can now be computed in the form

$$(x \vdash y)_{\diamond\Phi} = (\diamond x \vdash \diamond y)_{\diamond\Phi} = \bigwedge_{u \in U} \overleftarrow{\diamond x}_u \vdash \overleftarrow{\diamond y}_u$$

since the proximity in  $\diamond\Phi$  is just the proximity in  $\nabla\Phi$ , which is a subproxet of  $\Downarrow U$ , so its proximity is by Def. 3.1 the pointwise minimum. Hence

$\vdash$	$n$	$c$	$i$	$b$	$a$	$d$	$s$	$t$	$l$
$n$	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	1
$c$	$\frac{1}{3}$	1	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{5}{12}$	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{3}$
$i$	$\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{2}{3}$	$\frac{5}{12}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{5}{6}$	$\frac{1}{3}$
$b$	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{1}{2}$	1	$\frac{5}{16}$	$\frac{5}{8}$	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{1}{4}$
$a$	$\frac{4}{5}$	1	$\frac{2}{5}$	$\frac{4}{5}$	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{5}$
$d$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1	$\frac{2}{5}$	1	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{2}{5}$
$s$	$\frac{2}{5}$	1	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{2}{5}$	1	$\frac{1}{2}$	$\frac{2}{5}$
$t$	$\frac{1}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{5}$
$l$	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	1

The bottom five rows of this table display the values of the representable concepts themselves

$$(u \vdash j)_{\diamond\Phi} = (u \models j)_{\Phi} \quad (12)$$

$$(u \vdash v)_{\diamond\Phi} = \bigwedge_{\ell \in J} (v \models \ell)_{\Phi} \vdash (u \models \ell)_{\Phi} \quad (13)$$

for  $u, v \in U$  and  $j \in J$ , because  $\left(\overleftarrow{\diamond}u \vdash \overleftarrow{\diamond}x\right)_{\diamond\Phi} = \overleftarrow{\diamond}x_u$  follows from the general fact that  $\left(\nabla a \vdash \overleftarrow{\lambda}\right)_{\Downarrow A} = \overleftarrow{\lambda}_a$ . The upper four rows display the values

$$(j \vdash k)_{\diamond\Phi} = \bigwedge_{x \in U} (x \models j)_{\Phi} \vdash (x \models k)_{\Phi} \quad (14)$$

$$(j \vdash u)_{\diamond\Phi} = \bigwedge_{x \in U} (x \models j)_{\Phi} \vdash (x \vdash u)_{\diamond\Phi} = \bigwedge_{\ell \in J} (u \models \ell)_{\Phi} \vdash (j \vdash \ell)_{\diamond\Phi} \quad (15)$$

for  $u \in U$  and  $j, k \in J$ . Intuitively, these equations can be interpreted as follows:

- (13) *the proximity  $(u \vdash v)$  measures how well  $(v \models \ell)$  approximates  $(u \models \ell)$ :*
  - $u$ 's liking  $(u \models \ell)$  of any movie  $\ell$  is at least  $(u \vdash v) \cdot (v \models \ell)$ .
- (14) *the proximity  $(j \vdash k)$  measures how well  $(x \models j)$  approximates  $(x \models k)$ :*
  - any user  $x$ 's rating  $(x \models k)$  is at least  $(x \models j) \cdot (j \vdash k)$ ,
- (15) *the proximity  $(j \vdash u)$  measures how well  $j$ 's style approximates  $u$ 's taste:*
  - any  $x$ 's proximity  $(x \vdash u)$  to  $u$  is at least  $(x \models j) \cdot (j \vdash u)$ ,
  - $j$ 's proximity  $(j \vdash \ell)$  to any  $\ell$  is at least  $(j \vdash u) \cdot (u \models \ell)$ .

Since  $(a \vdash l) = \frac{4}{5}$ , it would make sense for Abby to accept Luka's recommendations, but not the other way around, since  $(l \vdash a) = \frac{1}{5}$ . Although Temra's rating of "Ikiru" is just  $(t \vdash i) = \frac{3}{5}$ , "Ikiru" is a good test of her taste, since her rating of it is close to both Dusko's and Stefan's ratings.

**Latent concepts?** While the proximities between each pair of users and items, i.e. between the induced *representable* concepts, provide an interesting new view on their relations, the task of determining the *latent* concepts remains ahead. What are the dominant tastes around which the users coalesce? What are the dominant styles that connect the items? What will such concepts look like? Formally, a dominant concept is a highly biased cut: in a high proximity of some of the representable concepts, and distant from the others. One way to find such cuts is to define the concepts of *cohesion* and *adhesion* of a cut along the lines of [30], and solve the corresponding optimization problems. Although there is no space to expand the idea in the present paper, some of the latent concepts can be recognized already by inspection of the above proximity table (recalling that each cut is both a supremum of users' and an infimum of items' representations).

## 6 Discussion and future work

What has been achieved? We generalized posets to proxets in Sec. 2 and 3, and lifted in Sec. 4 the FCA concept lattice construction to the corresponding construction over proxets, that allow capturing quantitative information. Both constructions share the same universal property, captured by the nucleus functor in Sec. 4.3. In both cases, the concepts are captured by *cuts*, echoing Dedekind's construction of the reals, and MacNeille's minimal completion of a poset. But while finite contexts yield finite concept lattices in FCA, in our analysis they yield infinitely many quantitative concepts. This is a consequence of introducing the infinite set of quantities  $[0,1]$ . The same phenomenon occurs in LSA [10], which allows the entire real line of quantities, and the finite sets of users and items span real vector spaces, that play the same role as our proxet completions. The good news is that the infinite vector space of latent concepts in LSA comes with a canonical basis of finitely many singular vectors, and that our proxet of latent concepts also has a finite generator, spelled out in Sec. 5. The bad news is that the generator described there is not a canonical basis of dominant latent concepts, with the suitable extremal properties, but an ad hoc basis determined by the given sets of users and items. Due to a lack of space, the final step of the analysis, finding the basis of dominant latent concepts, had to be left for a future paper. This task can be reduced to some familiar optimization problems.

More interestingly, and perhaps more effectively, this task can also be addressed using qualitative FCA and its concept scaling methods [13]. The most effective form of concept analysis may thus very well be a combination of quantitative and qualitative analysis tools. Our analysis of the numeric matrix, extracted from the given star ratings, should be supplemented by standard FCA analyses of a family of relational contexts scaled by various thresholds. We conjecture that the resulting relational concepts will be the projections of the dominant latent concepts arising from quantitative analysis. If that is the case, then the relational concepts can be used to guide computation of quantitative concepts.

This view of the quantitative and the qualitative concept analyses as parts of a putative general FCA toolkit raises an interesting question of their relation with LSA and the spectral methods of concept analysis [10, 1], which seem different. Some preliminary discussions on this question can be found in [31, 32]. While FCA captures a *particle* view of network traffic, where the shortest path determines the proximity of two network nodes, LSA corresponds to the *wave* view of the traffic, where the proximity increases with the number of paths. Different application domains seem to justify different views, and call for a broad view of all concept mining methods as parts of the same general toolkit.

**Acknowledgements.** Anonymous reviewers' suggestions helped me to improve the paper, and to overcome some of my initial ignorance about the FCA literature. I am particularly grateful to Dmitry Ignatov, who steered the reviewing process with a remarkable patience and tact. I hope that my work will justify the enlightened support, that I encountered in these first contacts with the FCA community.

## References

- [1] Yossi Azar, Amos Fiat, Anna Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *Proceedings of the thirty-third annual ACM Symposium on Theory of Computing*, STOC '01, pages 619–626, New York, NY, USA, 2001. ACM.
- [2] B. Banaschewski and G. Bruns. Categorical characterization of the MacNeille completion. *Archiv der Mathematik*, 18(4):369–377, September 1967.
- [3] R. Bělohlávek. *Fuzzy relational systems: foundations and principles*, volume 20. Plenum Publishers, 2002.
- [4] R. Bělohlávek. Concept lattices and order in fuzzy logic. *Annals Pure Appl. Logic*, 128(1-3):277–298, 2004.
- [5] R. Bělohlávek and V. Vychodil. What is a fuzzy concept lattice? In et al Sergei O. Kuznetsov, editor, *Proceedings of RSFDGrC 2011*, volume 6743 of *Lecture Notes in Computer Science*, pages 19–26. Springer, 2011.
- [6] M. M. Bonsangue, F. van Breugel, and J. J. M. M. Rutten. Generalized metric spaces: completion, topology, and power domains via the yoneda embedding. *Theor. Comput. Sci.*, 193(1-2):1–51, 1998.

- [7] A. Burusco and R. Fuentes-González. Construction of the L-fuzzy concept lattice. *Fuzzy Sets and systems*, 97(1):109–114, 1998.
- [8] A. Burusco and R. Fuentes-González. The study of the L-fuzzy concept lattice. *Mathware & Soft Computing*, 1(3):209–218, 2008.
- [9] Claudio Carpineto and Giovanni Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.
- [10] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [11] Patrick du Boucher-Ryan and Derek G. Bridge. Collaborative recommending using Formal Concept Analysis. *Knowl.-Based Syst.*, 19(5):309–315, 2006.
- [12] B. Ganter and S. Kuznetsov. Pattern structures and their projections. *Conceptual Structures: Broadening the Base*, pages 129–142, 2001.
- [13] B. Ganter and R. Wille. Conceptual scaling. *Institute for Mathematics and Its Applications*, 17:139, 1989.
- [14] Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.
- [15] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin/Heidelberg, 1999.
- [16] Mai Gehrke. Generalized kripke frames. *Studia Logica*, 84(2):241–275, 2006.
- [17] Mehdi Kaytoue, Sergei O. Kuznetsov, Juraj Macko, Wagner Meira Jr., and Amedeo Napoli. Mining biclusters of similar values with triadic concept analysis. In *Proceedings of CLA 2011*. CLA, 2011.
- [18] Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli. Pattern mining in numerical data: Extracting closed patterns and their generators. Research Report RR-7416, INRIA, October 2010.
- [19] Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli. Revisiting numerical pattern mining with formal concept analysis. In *Proceedings of IJCAI 2011*, pages 1342–1347. AAAI, 2011.
- [20] Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sébastien Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Inf. Sci.*, 10(181):1989–2001, 2011.
- [21] Gregory Maxwell Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1982. Reprinted in *Theory and Applications of Categories*, No. 10 (2005) pp. 1-136.
- [22] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [23] S. Krajčí. A generalized concept lattice. *Logic Journal of IGPL*, 13(5):543–550, 2005.
- [24] H. P. Künzi and M. P. Schellekens. On the yoneda completion of a quasi-metric space. *Theor. Comput. Sci.*, 278(1-2):159–194, 2002.
- [25] F. William Lawvere. Metric spaces, generalised logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43:135–166, 1973.
- [26] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. *Conceptual structures: applications, implementation and theory*, pages 32–43, 1995.

- [27] Tom Leinster and Christina Cobbold. Measuring diversity: the importance of species similarity. *Ecology*, 2012. to appear.
- [28] Holbrook Mann MacNeille. Extensions of partially ordered sets. *Proc. Nat. Acad. Sci.*, 22(1):45–50, 1936.
- [29] Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1971. (second edition 1997).
- [30] Dusko Pavlovic. Network as a computer: ranking paths to find flows. In Alexander Razborov and Anatol Slissenko, editors, *Proceedings of CSR 2008*, volume 5010 of *Lecture Notes in Computer Science*, pages 384–397. Springer Verlag, 2008. [arxiv.org:0802.1306](http://arxiv.org/0802.1306).
- [31] Dusko Pavlovic. On quantum statistics in data analysis. In Peter Bruza, editor, *Quantum Interaction 2008*. AAAI, 2008. [arxiv.org:0802.1296](http://arxiv.org/0802.1296).
- [32] Dusko Pavlovic. Quantifying and qualifying trust: Spectral decomposition of trust networks. In Pierpaolo Degano, Sandro Etalle, and Joshua Guttman, editors, *Proceedings of FAST 2010*, volume 6561 of *Lecture Notes in Computer Science*, pages 1–17. Springer Verlag, 2011. [arxiv.org:1011.5696](http://arxiv.org/1011.5696).
- [33] Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. Formal Concept Analysis in knowledge discovery: A survey. In Madalina Croitoru, Sébastien Ferré, and Dickson Lukose, editors, *ICCS*, volume 6208 of *Lecture Notes in Computer Science*, pages 139–153. Springer, 2010.
- [34] Kim Ritter Wagner. Liminf convergence in omega-categories. *Theor. Comput. Sci.*, 184(1-2):61–104, 1997.
- [35] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470. Dan Reidel, Dordrecht, 1982.
- [36] W.A. Wilson. On quasi-metric spaces. *Amer. J. Math.*, 52(3):675–684, 1931.
- [37] Yong woon Kim. Pseudo quasi metric spaces. *Proc. Japan Acad.*, 10:1009–10012, 1968.