Essentials of Formal Concept Analysis (FCA)

January 23, 2024

1 Introduction

The Formal Concept Analysis (FCA) is a mathematical tool for exploring the structure of *binary relations* (i.e. tables "units $\times$ attributes"), with the aim to provide a more expressive and informative representation of the data, by building so-called *formal concepts* (special pairs "sets of units"-"sets of attributes") and arranging them into a relational network known as *concept lattice*. In this respect, FCA is not properly a dimensionality reduction algorithm, in that the delivered conceptual representation is mathematically equivalent to the input binary relation. FCA allows to classify objects/attributes into hierarchies of specializing/generalizing concepts (i.e. to define a *taxonomy*) and to identify logic implications between attributes, making it possible to extract association rules from the input relation and to use them for reasoning, in different application domains (e.g. to develope recommendation engines). The mathematics behind FCA is quote different from that employed in classical statistical tools and involve concepts from *relation theory*, *partially ordered set theory* and *lattice theory*. Before presenting the FCA, we thus provide an essential introduction to this background material.

2 Binary relations

Let X and Y be two sets; a *binary relation* R on X and Y is a subset of their cartesian product:

$$R \subseteq X \times Y. \qquad (1)$$

So R is a set of pairs of elements of type (x, y), with $x \in X$ and $y \in Y$. Usually, the more formal expression $(x, y) \in R$ is rendered in the more expressive way xRy, to mean that x and y are related. When $Y = X$, $R \subseteq X \times X = X^2$ is said to be a *binary relation on X*. Here below we mainly focus on this special case. We are in particular interested in relations satisfying different combinations of the following properties:

1. *Reflexivity*: aRa, $\forall a \in X$.
2. *Symmetry*: $aRb \Leftrightarrow bRa$, $\forall a, b \in X$.
3. *Antisymmetry*: aRb & $bRa \Rightarrow a = b$, $\forall a, b \in X$.
4. *Transitivity*: aRb & $bRc \Rightarrow aRc$, $\forall a, b.c \in X$.

Examples. Let $X = \{a, b, c, d\}$ and

$$\begin{aligned} R_1 &= \{(a,a),(b,b),(c,c),(a,c)\} \\ R_2 &= \{(a,b),(b,c),(a,c)\} \\ R_3 &= \{(a,b),(b,a),(c,c)\} \end{aligned}$$

R_1 is antisymmetric; R_2 is transitive; R_3 is symmetric. *Divisibility* between natural numbers defines a reflexive, antisymmetric and transitive relation *div* on $\mathbb{N}$ (i.e. m *div* n if m divides n). *Orthogonality* between elements of a vector space V with scalar product induces a symmetric relation $\perp$ on V.

2.1 Equivalence relations

Equivalence relations are among the most fundamental concepts of all mathematics.

Definition. A binary relation R on X is called *equivalence relation* if it is reflexive, symmetric and transitive. Equivalence relations are here denoted $\sim$.

Let $\sim$ be an equivalence relation on X and let $a \in X$. The set $\{x \in X : x \sim a\}$ of elements of X equivalent to a is called the *equivalence class of a* and is denoted by $[a]$ (a is called *representative* of the equivalence class). If $a \sim b$, then $[a] = [b]$, because any element equivalent to a is also equivalent to b (and viceversa), by transitivity. On the other hand, if $[a] \neq [b]$, then $[a] \cap [b] = \emptyset$, because if they shared a common element c, then $a \sim b$, again by transitivity (through c) and so $[a] = [b]$. Thus, either two equivalence classes coincide or they are disjoint. Moreover, due to reflexivity, any element of X belongs to one (and only one) equivalence class. So, $\sim$ induces a *partition* of X (i.e. a decomposition of X as the union of disjoint subsets of it). Viceversa, any partition $X = \cup S$ of disjoint subsets of X defines an equivalence relation by $a \sim b \Leftrightarrow \exists\, S : a, b \in S$.

Definition. The set of equivalence classes of $\sim$ on X is called the *quotient set* of X with respect to $\sim$ and is denoted by $X/\sim$.

Examples.

1. Two sets are *equipollent* if there is a bijective correspondence between them. Being equipollent defines an equivalence relation.

2. Two fractions are equivalent if they represent the same number (e.g. $2/3 \sim 4/6$). So fractions can be partitioned into equivalence classes and, in computations, any fraction can be substituted with one equivalent to it.

3. Two algebraic expressions are *equivalent* if they can be reduced to the same expression, by applying the rules of calculus, e.g. $(a+b)(a-b) \sim a \cdot a - b^2$, because both reduce to $a^2 - b^2$. Algebraic computations are just "trajectories" within equivalence classes of expressions, in search of the "simplest" (i.e. non-reducible) result.

4. Two computer programs are *denotationally equivalent*, if they denote the same function, i.e. if they give the same output, for the same input, irrespective of the details of the algorithms actually implemented. So, denotationally equivalent programs can be very different in terms of computation time or memory space required, features that can be instead very relevant, from other points of view.

5. Any function $f : X \to Y$ induces an equivalence relation on X by $a \sim b \Leftrightarrow f(a) = f(b)$. The equivalence classes of $\sim$ are the counterimages $f^{-1}(y)$, $y \in Y$. On each of these classes, the input function f is constant and on different classes it assumes different values. Let π and h be the functions defined by $\pi : a \to [a]$ and $h : [a] \to f(a)$, then the function f can be "factorized" as $f = h \circ \pi$, where $\circ$ is the usual function composition. $X/\sim$ is termed "quotient", just because it allows f to be factorized this way.

Remark. Performing a cluster analysis on a dataset D is thus nothing but defining an equivalence relation $\sim$ on the set of units U and taking the corresponding quotient set $U/\sim$.

2.2 Partial order relations

Partial order relations play a key role in the development of FCA and of other tools for the treatment of multidimensional systems of ordinal data.

Definition. A binary relation on a set X is called a *partial order relation* if it is reflexive, antisymmetric and transitive. Partial order relations are here denoted $\trianglelefteq$. A set X endowed with a partial order relation is termed *partially ordered set* (or *poset*, for short).

Let $\Pi = (X, \trianglelefteq)$ be a poset on X. If $a \trianglelefteq b$, we say that a and b are *comparable* and that b *dominates* a; if neither $a \trianglelefteq b$ nor $b \trianglelefteq a$, a and b are termed *incomparable* (written $a||b$). When $a \trianglelefteq b$ and $a \neq b$, we write $a \triangleleft b$. In practice, a poset is a set where some pairs of elements can be ordered, but others cannot so that the resulting ordering is only partial.

Examples.

1. Try to order ice-cream tastes based on your preferences; maybe *lemon* $\triangleleft$ *chocolate*, but *strawberry*$||$*peach*; so the set of tastes is just partially ordered.

2. The divisibility relation introduced earlier is a partial order relation on $\mathbb{N}$.

3. Let X be a set and $\mathbb{P}(X)$ its powerset (i.e. the set of all of its subsets). $\mathbb{P}(X)$ is turned into a poset by setting $A \trianglelefteq B \Leftrightarrow A \subseteq B$, for $A, B \in \mathbb{P}(X)$. The powerset poset will play a key role in the subsequent developments.

4. The set $\{3, 5.7.11.13\}$ under the usual order on $\mathbb{N}$ forms a partial order with no incomparabilities, i.e. a so-called *total* (or *complete*, or *linear*) order, or also a *chain.*

5. The set $\{3, 5.7.11.13\}$ under the divisibility relation is a partial order with no comparabilities (all elements are pairwise incomparable, being prime numbers), i.e. a so-called *antichain.*

6. A vector $\boldsymbol{p} = (p_1, \ldots, p_k)$ *componentwise* dominates $\boldsymbol{q} = (q_1, \ldots, q_k)$ ($\boldsymbol{q} \triangleleft_{cmp} \boldsymbol{p}$) if $q_i \leq p_i$, $\forall i = 1, \ldots, k$ and there exists j such that $q_j < p_j$. The set of *strings* of length 3, over the alphabet $\{0, 1\}$ ordered *componentwise* is a partial order of 8 elements (see Figure 1).

Let $\Pi = (X, \trianglelefteq)$ be a poset on X. Its *dual* is the poset $\Pi^d = (X, \trianglelefteq^d)$, where $a \trianglelefteq^d b \Leftrightarrow b \trianglelefteq a$ (in practice, the dual of Π is obtained by inverting its order). An element a such that $b \trianglelefteq a \ \forall b \in X$ (if existing) is called the *maximum*, or the *top*, of Π and is denoted by $\top$; it is called *maximal* if $a \trianglelefteq b \ \Rightarrow \ a = b$. In other words, the maximum dominates all of the other elements, while maximal elements are dominated by no other elements. *Minimum*, or *bottom*, (denoted by $\bot$) and *minimal* elements are defined dually. Finite posets have necessarily maximal/minimal elements, but not a maximum/minimum one. A subset of mutually comparable elements of Π is called a *chain*, while a subset of mutually incomparable elements is called an *antichain*. A *downset* D of Π is a subset of X such that $b \in D \,\&\, a \trianglelefteq b \Rightarrow a \in D$, i.e. it is a subset downward closed. Dually, an *upset* U is a subset upward closed, i.e. such that $a \in U \,\&\, a \trianglelefteq b \Rightarrow b \in U$. Given an element $a \in X$, the set $a\downarrow = \{b \in X : b \trianglelefteq a\}$ is the downset *generated* by a. The upset generated by a is defined dually as $a\uparrow = \{b \in X : a \trianglelefteq b\}$.

Posets are usually graphically represented as *Hasse diagrams*, which are *directed graphs* depicted according to the following two rules (see Figures 1 and 2, for a few examples):

1. If $a \trianglelefteq b$, the node corresponding to a is placed below that corresponding to b.

2. If b *covers* a (i.e. if $a \trianglelefteq b$ and there is no c such that $a \triangleleft c \triangleleft b$), then an edge is inserted between the two nodes.

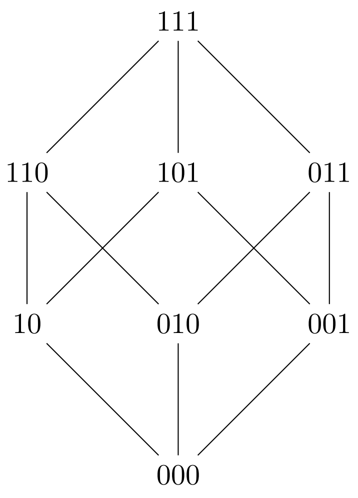

Figure 1: The "cube" poset (notice that the edges are implicitly oriented downward).

Let $\Pi_1 = (X_1, \trianglelefteq_1)$ and $\Pi_2 = (X_2, \trianglelefteq_2)$ be two posets. A map $\varphi : \Pi_1 \to \Pi_2$ is called *order-preserving* if $a \trianglelefteq_1 b \Rightarrow \varphi(a) \trianglelefteq_2 \varphi(b)$. Notice that an

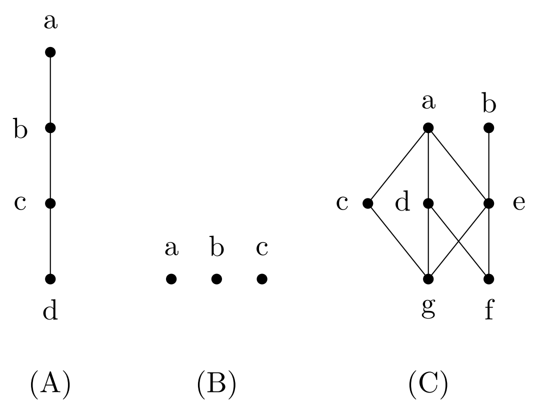

Figure 2: (A) - A linear order; (B) - An antichain; (C) - A poset with both comparabilities and incomparabilities.

order-preserving map can transform incomparable elements into comparable ones, but it cannot "destroy" or "invert" existing dominances. An *order isomorphism* between $\Pi_1 = (X_1, \trianglelefteq_1)$ and $\Pi_2 = (X_2, \trianglelefteq_2)$ is an invertible order-preserving map φ, whose inverse φ^{-1} is also order-preserving (notice, that to get an isomorphism it is not sufficient for the order-preserving map φ to be invertible, i.e. bijective, as a map between sets). Two isomorphic posets have the same "relational" structure, although their elements can have a totally different nature.

Examples. Let Π_1 be the set $\{(0,0), (0,1), (1,0), (1,1)\}$ ordered componentwise and let Π_2 be the set $\{(a,a), (a,b), (b,a), (b,b)\}$ totally ordered *lexicographically*, i.e. $(b,b) \triangleleft (b,a) \triangleleft (a,b) \triangleleft (a,a)$. Then the map φ defined by

$$\begin{array}{ccc}
(1,1) & \longrightarrow & (a,a) \\
(1,0) & \longrightarrow & (b,a) \\
(0,1) & \longrightarrow & (a,b) \\
(0,0) & \longrightarrow & (b,b)
\end{array}$$

is invertible and order-preserving, but it is not an order isomorphism (its inverse is not order-preserving). The map $\psi : \Pi_1 \to \Pi_1$ defined by $(x,y) \longrightarrow (y,x)$ is an order isomorphism of Π_1 in itself, i.e. an order *automorphism*.

Draw the Hasse diagrams of the posets of the previous examples and check the assertions graphically.

3 Lattices

Let $\Pi = (X, \trianglelefteq)$ be a poset and let $S \subseteq X$. An element $b \in X$ such that $a \trianglelefteq b$, $\forall a \in S$ is called an *upper bound* of S. If the set $S\!\uparrow = \{b \in X : a \trianglelefteq b, \ \forall a \in S\}$ has a minimum, this is called the *least upper bound* (*l.u.b*), or the *supremum*, of S. Dually, if the set $S\!\downarrow = \{b \in X : b \trianglelefteq a, \ \forall a \in S\}$ of *lower bounds* of S has a maximum, this is called the *greatest lower bound* (*g.l.b.*), or the *infimum*, of S.

In general posets, l.u.b and g.l.b. need not exist (e.g. in antichains). The class of posets where they exist is extremely important and plays a key role in Formal Concept Analysis. This motivates the following definition.

Definition A poset $\Pi = (X, \trianglelefteq)$ where the l.u.b. and g.l.b. exist for any pair $\{a, b\}$ of elements of X is called a *lattice*. If the l.u.b. and g.l.b. exist for any $S \subseteq X$, Π is called a *complete* lattice.

In the context of lattices, the l.u.b. of a, b is called the *join* of a and b and is denoted by $a \vee b$; their g.l.b. is called the *meet* of a and b and is denoted by $a \wedge b$. We write $\bigvee S$ and $\bigwedge S$ for the join and the meet of a general subset S.

Examples.

1. The poset $\mathbb{P}(X)$ of subsets of X partially ordered by inclusion is a lattice, with $A \vee B = A \cup B$ and $A \wedge B = A \cap B$ (i.e. the join of two sets is their union and the meet is their intersection). This lattice is complete.
2. Chains are lattices (but while finite chains are complete - see below - infinite chains need not; e.g. the set of rational numbers $\mathbb{Q}$ is not a complete lattice, because of... reals: can you explain this?).
3. The (unlabeled) Hasse diagrams of Figure 3, but one, are lattices. Which is the wrong one? Put labels on them and check whether the lattice axioms are fulfilled.

Lattice as algebraic structures. It is very useful to look at the join and the meet as to two binary operations on the lattice L:

$$\begin{aligned} \vee &: L \times L \to L \\ &: (a, b) \mapsto \vee(a, b) = a \vee b \end{aligned}$$

$$\begin{aligned} \wedge &: L \times L \to L \\ &: (a, b) \mapsto \wedge(a, b) = a \wedge b \end{aligned}$$

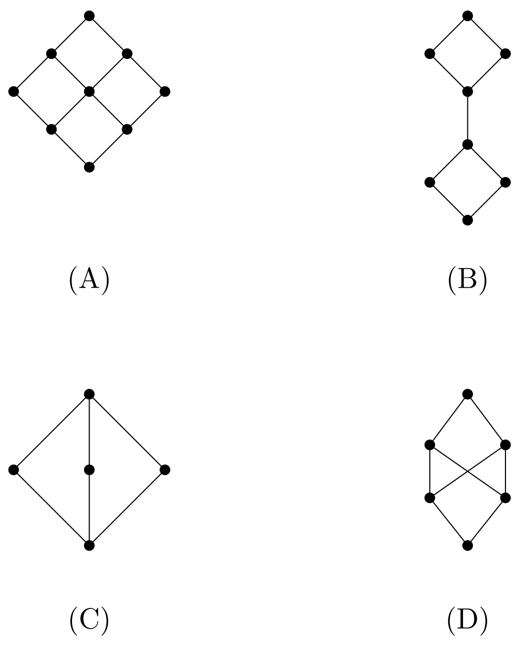

Figure 3: Four posets; all are lattices but one; which?

As binary operations, join and meet satisfy the following properties (the proof are trivial, from the definitions):

1. $a \vee a = a$; $a \wedge a = a$ (*idempotency*).
2. $a \vee b = b \vee a$; $a \wedge b = b \wedge a$ (*commutativity*).
3. $(a \vee b) \vee c = a \vee (b \vee c)$; $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ (*associativity*).

Join and meet also satisfy the so-called *absorption* rule, that links them as follows:

$$\begin{aligned} a \vee (a \wedge b) &= a \\ a \wedge (a \vee b) &= a \end{aligned}$$

(the proof of these last statements is simple, observing that by definition $a \wedge b \trianglelefteq a$ and $a \trianglelefteq a \vee b$). Notice that by associativity, the join and the meet of finite subsets of L always exist, so that *a finite lattice is always complete.*

A simple exercise. Show that $a = a \wedge b \Leftrightarrow b = a \vee b$. Solution: $a = a \wedge b \Rightarrow a \vee b = (a \wedge b) \vee b = b$ (by the absorption rule). The viceversa is proved analogously.

Looking at $\vee$ and $\wedge$ as binary operations allows lattices to be seen not just as special kinds of posets, but as *algebraic structures*, i.e. as "sets + operations". We can then define a lattice as a triple $L = (X, \vee, \wedge)$, where $\vee$ and $\wedge$ satisfy the properties of idempotency, commutativity, associativity and the absorption rule. The two ways of defining lattices are equivalent, in fact:

1. The $\trianglelefteq$ relation defined by $a \trianglelefteq b \Leftrightarrow a \wedge b = a \Leftrightarrow a \vee b = b$ is a partial order relation. In fact, (i) $a = a \Rightarrow a \wedge a = a \Rightarrow a \trianglelefteq a$ (reflexivity), (ii) $a \trianglelefteq b \,\&\, b \trianglelefteq a \Rightarrow a = a \wedge b = b$ (antisymmetry) and (iii) $a \trianglelefteq b \,\&\, b \trianglelefteq c \Rightarrow a = a \wedge b \,\&\, b = b \wedge c \Rightarrow b = a \vee b \ \&\ c = b \vee c \Rightarrow a \wedge c = a \wedge (b \vee c) = a \wedge ((a \vee b) \vee c) = a \wedge (a \vee (b \vee c)) = a \Rightarrow a \trianglelefteq c$ (transitivity).

2. The join and meet induced back by $a \trianglelefteq b \Leftrightarrow a \wedge b = a \Leftrightarrow a \vee b = b$ are just $\vee$ and $\wedge$ themselves. In fact, since by the absorption rule $a = a \wedge (a \vee b)$ and $b = b \wedge (a \vee b)$, it holds $a, b \trianglelefteq a \vee b$. Moreover, $a, b \trianglelefteq c \Rightarrow c \vee (a \vee b) = (c \vee a) \vee b = c \vee b = c \Rightarrow a \vee b \trianglelefteq c$. So $a \vee b$ is the l.u.b of $\{a, b\}$, i.e. the supremum of a and b. The proof that also the induced meet is the same as the original one is analogous.

In summary, we can define lattices as special types of posets or as algebraic structures. The two definitions are completely equivalent, since they induce each other, and can be used interchangeably, based on convenience.

Example. The powerset lattice $(\mathbb{P}(X), \subseteq)$, as a poset, is the same as the lattice $(\mathbb{P}(X), \cup, \cap)$, as an algebraic structure. If we take the dual, we get the lattice $(\mathbb{P}(X), \supseteq)$, which algebraically corresponds to $(\mathbb{P}(X), \cap, \cup)$.

An important point now arises. As posets, two lattices L_1 and L_2 are equivalent if they are order-isomorphic. As algebraic structures, they are equivalent if they are *algebraically* isomorphic, i.e. if there exist bijective maps $\psi : L_1 \to L_2$ and $\varphi : L_2 \to L_1$ such that:

1. $\psi(a \vee_1 b) = \psi(a) \vee_2 \psi(b)$
2. $\psi(a \wedge_1 b) = \psi(a) \wedge_2 \psi(b)$
3. $\varphi(a \vee_2 b) = \varphi(a) \vee_1 \varphi(b)$
4. $\varphi(a \wedge_2 b) = \varphi(a) \wedge_1 \varphi(b)$
5. $\varphi \circ \psi = \mathrm{id}_1$
6. $\psi \circ \varphi = \mathrm{id}_2$

where id_1 and id_2 are the identity maps on L_1 and L_2, respectively (thus ψ and φ are called *lattice isomorphisms* and $\psi = \varphi^{-1}$).

The question is: are order isomorphisms also algebraic isomorphisms (and viceversa)? The question is important because in many cases we prove properties of a lattice by showing that it is isomorphic (as a poset, or as an algebraic structure) to another lattice enjoying those properties and it would be very disturbing whether the equivalence between the definitions of lattice would break down, when considering isomorphisms. In practice the question is whether or not order isomorphisms preserve joins and meets (since it is clear that preserving them preserves the induced order relation). The answer is indeed positive and so order and algebraic isomorphisms are equivalent concepts.

Proposition. Let L_1 and L_2 be lattices and let $\varphi : L_1 \to L_2$ be an order isomorphism, then it preserves (i) joins and (ii) meets.

Proof. To prove (i), let $a, b \in L_1$:

1. $a, b \trianglelefteq_1 a \vee_1 b$
2. $\Rightarrow \psi(a), \psi(b) \trianglelefteq_2 \psi(a \vee_1 b)$
3. $\Rightarrow \psi(a) \vee_2 \psi(b) \trianglelefteq_2 \psi(a \vee_1 b)$.

On the other hand, φ^{-1} being order-preserving:

1. $\psi(a), \psi(b) \trianglelefteq_2 \psi(a) \vee_2 \psi(b)$
2. $\Rightarrow a, b \trianglelefteq_1 \psi^{-1}(\psi(a) \vee_2 \psi(b))$
3. $\Rightarrow a \vee_1 b \trianglelefteq_1 \psi^{-1}(\psi(a) \vee_2 \psi(b))$
4. $\Rightarrow \psi(a \vee_1 b) \trianglelefteq_2 \psi(a) \vee_2 \psi(b)$.

Thus, $\psi(a \vee_1 b) = \psi(a) \vee_2 \psi(b)$. The proof of (ii) is analogous.

■

4 Closure operators

We now introduce a fundamental tool, for the development of the Formal Concept Analysis.

Definition. Let Π be a poset. A *closure* operator $c : \Pi \to \Pi$ is a map satisfying the following three properties, for all $a, b \in \Pi$:

1. $a \trianglelefteq c(a)$.
2. $a \trianglelefteq b \Rightarrow c(a) \trianglelefteq c(b)$ (*monotonicity*).
3. $c^2(a) = c(a)$ (*idempotency*)

An element such that $a = c(a)$ (i.e. a *fixed point*, or *fixpoint*, of the closure operator) is called *closed*. From property (iii) above, we see that $c(a)$ is a fixpoint of $c(\cdot)$, for any $a \in \Pi$.

Examples. Let X be a set and $\mathbb{P}(X)$ be its powerset, seen as a poset under the inclusion ordering. A family $M \subseteq \mathbb{P}(X)$ of subsets of X is a *top intersection structure*, if:

1. it is closed under intersection: $\cap S \in M, \forall S \subseteq M$.
2. $X \in M$ (so for any $A \in \mathbb{P}(X)$, there is $B \in M : A \subseteq B$).

Then the operator

$$\begin{aligned} C &: X \to M \\ &: A \mapsto C(A) = \bigcap_{B:A\subseteq B, B\in M} B \end{aligned}$$

is a closure operator on X (think of the way the topological closure of an open set, or the linear closure of a set of vectors are defined).

Exercise, Prove that the above operator is indeed a closure.

Lemma. Let Π be a poset, $c(\cdot)$ be a closure operator on it and K be the set of closed elements of Π. Then for any $a \in \Pi$, $c(a) = \bigwedge\{b \in K : a \trianglelefteq b\}$, i.e. the closure of an element of Π is the l.u.b. of the set of closed elements dominating it.

Proof. Let $K_a = \{b \in K : a \trianglelefteq b\}$. By property 1 of closure operators, $a \leq c(a) \Rightarrow c(a) \in K_a$; then, by properties 2 and 3 we have $b \in K_a \Rightarrow a \trianglelefteq b \Rightarrow c(a) \trianglelefteq c(b) = b \Rightarrow c(a) \trianglelefteq b, \forall\, b \in K_a \Rightarrow c(a) = \bigwedge K_a$, as stated.

■

The notion of closure operator is very general and indeed has been given for posets. When the domain is a complete lattice, however, closure operators enjoy additional properties; in particular, it turns out that the subset of closed elements of a complete lattice L is itself a complete lattice, whose join and meet are related to those of L.

Theorem. Let $L = (X, \vee, \wedge)$ be a complete lattice and let $c(\cdot)$ be a closure operator on it. Let L_c be the set of closed elements of L under $c(\cdot)$. Then:

1. L_c is a complete lattice.
2. Its meet $\bigwedge$ coincides with that of L, i.e. $\bigwedge_c S = \bigwedge S$, for $S \subseteq L_c$.
3. Its join $\bigvee_c$ is the closure of the join of L, i.e. $\bigvee_c S = c(\bigvee S)$, for $S \subseteq L_c$.

Proof. Let us first prove that $\bigwedge_c = \bigwedge$. Let $A \subseteq L_c$; we know that $\bigwedge A$ exists, since L is a complete lattice. We want to show that $\bigwedge A$ is closed. Clearly, $\bigwedge A \trianglelefteq c(\bigwedge A)$; on the other hand, $\bigwedge A \trianglelefteq a, \forall\, a \in A \Rightarrow c(\bigwedge A) \trianglelefteq c(a) = a, \forall\, a \in A \Rightarrow c(\bigwedge A) \trianglelefteq \bigwedge A$ and so $c(\bigwedge A) = \bigwedge A$, i.e. $\bigwedge A$ is closed. This means that the meet in L serves also as the meet in L_c and this proves statement 2. Since $\bigvee A$ (which exists L being complete) is defined as the g.l.b. of $A\!\uparrow$ and we have just proved that the g.l.b. exists in L_c for any subset of L_c, it follows that L_c is a complete lattice, proving statement 1. To prove statement 3: $\bigvee_c A = \bigwedge_c\{b \in L_c : a \trianglelefteq b, \forall\, a \in A\} = \bigwedge\{b \in L_c : a \trianglelefteq b, \forall\, a \in A\} = \bigwedge\{b \in L_c : \bigvee A \trianglelefteq b\} = c(\bigvee a)$, by the previous lemma.

■

In practice, the proof shows that to get the join and the meet of L_c it is sufficient to close the join and the meet of L, but since the meet of L is already closed, it coincides with the meet of L_c.

Example. For the lattice $(\mathbb{P}(X), \cup, \cap)$ and a generic closure operator associated to a top intersection structure on it, the previous theorem simply states that the meet of two closed subsets A and B is their intersection and that their join is the closure of their union, i.e. (by the lemma) the intersection of all of the closed subsets containing $A \cup B$.

5 Formal Concept Analysis

We now apply the former mathematical machinery to the development of Formal Concept Analysis. As it will be seen, FCA is a straightforward application of the properties of suitable closure operators on the powersets of the set of units (here called *objects*) and of the set of attributes.

A - Formal contexts and the polar maps. Let G be a set of *objects* and M a set of attributes and let $R \subseteq G \times M$ be a binary relation on them (so that gRm means that object g enjoys attribute m). The triple (G, M, R) is here called *formal context.* Let us define two maps (denoted with the same symbol, with a useful abuse of notation), called *polar* or *derivation* maps, as follows:

$$\begin{aligned} {}' &: \mathbb{P}(G) \to \mathbb{P}(M) \\ &: A \mapsto A' = \{m \in M : gRm\, \forall g \in A\} \end{aligned}$$

$$\begin{aligned} {}' &: \mathbb{P}(M) \to \mathbb{P}(G) \\ &: B \mapsto B' = \{g \in G : gRm \, \forall m \in B\}. \end{aligned}$$

For $A \subseteq G$, the polar map returns the set of attributes shared by all the objects in A; for $B \subseteq M$, the polar map returns the set of objects sharing all of the attributes in M. Applying the polar maps twice, we get the following sequences:

$$A \to A' \to (A')' = A''$$

and

$$B \to B' \to (B')' = B''.$$

A'', which is an element of $\mathbb{P}(G)$, is "the set of objects sharing all of the attributes shared by the objects in A". In other words, we start from A, consider the attributes shared by its elements, check whether there are other objects sharing them and, in the affirmative, we add these new objects to it, "completing" A and getting A''. Similarly, B'', which is an element of $\mathbb{P}(M)$, is the set of attributes shared by the objects sharing the attributes in B.

Operators $'$ and $''$ satisfy the following important properties (the sets involved below can be thought of as elements of $\mathbb{P}(G)$ or $\mathbb{P}(M)$, based on which polar map is considered):

1. $X_1 \subseteq X_2 \Rightarrow X_2' \subseteq X_1'$ (if the set of objects (attributes) is enlarged, the set of shared attributes (objects) is restricted).

2. $X \subseteq X''$ (indeed, X'' is obtained from X by adding no, or some elements).

3. $X_1 \subseteq X_2 \Rightarrow X_1'' \subseteq X_2''$. This follows from $X_1 \subseteq X_2 \Rightarrow X_2' \subseteq X_1' \Rightarrow X_1'' = (X_1')' \subseteq (X_2')' = X_2''$.

4. $X''' = X'$. By item 2, $X' \subseteq (X')'' = X'''$; on the other hand, $X \subseteq X'' \Rightarrow X''' = (X'')' \subseteq X'$.

5. $(X'')'' = X''$. We have: $(X'')'' = (X''')' = (X')' = X''$.

From these results, it follows that

$$\begin{aligned} {}'' &: \mathbb{P}(G) \to \mathbb{P}(G) \\ &: A \mapsto A'' \end{aligned}$$

and

$$\begin{aligned} {}'' &: \mathbb{P}(M) \to \mathbb{P}(M) \\ &: B \mapsto B'' \end{aligned}$$

are closure operators on $\mathbb{P}(G)$ and $\mathbb{P}(M)$, respectively.

Remark. Properties 4 and 5 above have been proved in a rather abstract way, exploiting the properties of polar maps and of their compositions. They certainly could be proved using a more concrete and analytical approach, as we show below for the equality $X''' = X'$. After reading this alternative derivation, you should appreciate abstraction...

Let $X \in \mathbb{P}(G)$ be a set of objects (similarly, if X is a set of attributes):

1. $X' = \bigcap_{g \in X} g'$ (by definition; *notice that we write g' instead of $\{g\}'$ and later do the same with single attributes and with the cloosure operators*).

2. $g \in X'' \Rightarrow gRm \, \forall m \in X'$ (by definition).

3. $X''' = \bigcap_{g \in X''} g'$ (by definition).

4. By the proprieties of intersection
$$\bigcap_{g \in X''} g' = \left(\bigcap_{g \in X} g' \right) \bigcap \left(\bigcap_{g \in X'' \setminus X} g' \right).$$

5. By virtue of item 2, it is
$$\bigcap_{g \in X} g' \subseteq \bigcap_{g \in X'' \setminus X} g'$$
and so
$$\left(\bigcap_{g \in X} g' \right) \bigcap \left(\bigcap_{g \in X'' \setminus X} g' \right) = \bigcap_{g \in X} g'.$$

6. $\Rightarrow X''' = X'$.

Example. Table 1 (see Davey B.A. & Priestley H. A., *Introduction to lattices and order*, CUP 2002) provides a formal context for planets. Here G is the set of planets in the solar system and M is a set of attributes concerning their size, distance from the Sun and presence of a moon. A planet g has attribute m if an "x" is placed in the corresponding cell. It can be easily checked that:

- {Pl}$'$={small, far, yes}.
- {Ju}$'$={large, far, yes}.
- {Pl, Ju}$'$={far, yes}={Pl}$'$ $\cap$ {Ju}$'$.
- {far, yes}$'$={Ju, Sa, Ur, Ne, Pl}.

	Size			Distance		Moon	
	s	m	l	near	far	yes	no
Me	x			x			x
Ve	x			x			x
Ea	x			x		x	
Ma	x			x		x	
Ju			x		x	x	
Sa			x		x	x	
Ur		x			x	x	
Ne		x			x	x	
Pl	x				x	x	

Table 1: Planets of the solar system and their basic features: size (small-medium-large), distance from the Sun (near-far), with moon (yes-no).

- {Ju, Sa, Ur, Ne, Pl}$'$={far, yes}.

B - Formal concepts. A *formal concept* of the context (G, M, R) is a pair (A, B), where $A \in \mathbb{P}(G)$ and $M \in \mathbb{P}(M)$, with the property that $B = A'$ and $A = B'$. So, a formal concept is a pairing of a set of object and a set of attributes, linked by the above conditions. Sets A and B are called the *extent* and the *intent* of the concept, respectively. Notice that

$$B = A' \,\&\, A = B' \Leftrightarrow B = B'' \,\&\, A = A''$$

so that any concept (A, B) can be alternatively written as (A'', A') or (B', B''). The set of concepts of the context (G, M, R) is denoted by $\mathbb{B}(G, M, R)$, while the set of their extents and of their intents by $\mathbb{B}_G$ and $\mathbb{B}_M$, respectively.

Very important remark! $\mathbb{B}_G$ and $\mathbb{B}_M$ are the images under $''$ of $\mathbb{P}(G)$ and $\mathbb{P}(M)$; since powersets are complete lattices and $''$ is a closure operator, $\mathbb{B}_G$ and $\mathbb{B}_M$ are themselves complete lattices $(\mathbb{B}_G, \vee_G, \wedge_G)$ and $(\mathbb{B}_M, \vee_M, \wedge_M)$. Since the elements of $\mathbb{B}_G$ are closed (see the previous discussion on closure operators), it is $A_1 \wedge_G A_2 = A_1 \cap A_2$ (i.e., their meet in $\mathbb{P}(G)$) and $A_1 \vee_G A_2 = (A_1 \cup A_2)''$ (i.e., the closure of their join in $\mathbb{P}(G)$).

So, formally a concept is a pair objects-attributes, such that no attribute shared by the objects is missing in the intent and no object sharing the attributes is missing in the extent. But why are these pairs of interest and what do they represent? These questions will be implicitly answered later, as the theory gets developed and the properties of $\mathbb{B}(G, M, R)$ worked out. Intuitively, a formal concept is a *unit* of (formal) knowledge that identifies *classes* of objects (based on their shared attributes) and, jointly, classes of attributes (based on the objects sharing them). As it will be seen below, the co-implication between extents and intents makes it possible to put a partial order relation on the set of concepts, relating them in terms of generalization/specification. Such a partial order provides an alternative but equivalent representation of the input binary relation, as a network of prototypical “entities”or “types”, linked in a sort of *taxonomy.*

Example. Previously, we have highlighted that

$$\{\text{Ju, Sa, Ur, Ne, Pl}\}' = \{\text{far, yes}\}$$

$$\{\text{far, yes}\}' = \{\text{Ju, Sa, Ur, Ne, Pl}\}$$

so that ({Ju, Sa, Ur, Ne, Pl},{far, yes}) is a formal concept of the planet context. Another concept is provided by

$$(\{\text{Me, Ve}\}, \{\text{small, near, no}\}),$$

as it can be checked directly.

C - $\mathbb{B}(G, M, R)$ is a poset. Let (A_1, B_1) and (A_2, B_2) be two concepts of the context (M, G, R). We put $(A_1, B_1) \trianglelefteq_{\mathbb{B}} (A_2, B_2)$ if and only if $A_1 \subseteq A_2$ or $B_2 \subseteq B$. The two conditions are equivalent, since enlarging the extent implies restricting the intent and viceversa or, more formally, since $A_1 \subseteq A_2 \Leftrightarrow B_2 = A_2' \subseteq A_1' = B_1$. We say that (A_1, B_1) *specializes* (A_2, B_2) and that (A_2, B_2) *generalizes* (A_1, B_1) (or that (A_1, B_1) is a *sub-concept* of (A_2, B_2) and that the latter is a *super-concept* of the former).

Example. Figure 4 depicts the Hasse diagram of the concept poset, for the planet context. Now, since

$$\{\text{small, near}\} \subset \{\text{small, near, yes}\}$$

and

$$\{\text{small, near, yes}\}' = \{\text{Ea, Ma}\},$$

it follows that

$$(\{\text{Me, Ve, Ea, Ma}\},\{\text{small, near}\})$$

is a super-concept of

$$(\{\text{Ea, Ma}\},\{\text{small, near, yes}\}),$$

providing an example of ordering between concepts.

D - The poset $(\mathbb{B}(G,M,R),\trianglelefteq_{\mathbb{B}})$ *is a complete lattice.* From the previous point, it holds that the map

$$\begin{aligned}\pi_G \;:\;& (\mathbb{B}(G,M,R),\trianglelefteq_{\mathbb{B}}) \to (\mathbb{B}_G,\subseteq)\\ & (A,B)\mapsto A,\end{aligned}$$

which *extracts* the extent of a concept, is an order isomorphism (in fact, a concept is uniquely identified by its extent, or by its intent). Since $\mathbb{B}_G$ is a complete lattice, and join/meets are preserved by order isomorphisms, then also $(\mathbb{B}(G,M,R),\trianglelefteq_{\mathbb{B}})$ is a complete lattice $(\mathbb{B}(G,M,R),\vee_{\mathbb{B}},\wedge_{\mathbb{B}})$. Alternatively, we could have got to the same result by considering the map

$$\begin{aligned}\pi_M \;:\;& (\mathbb{B}(G,M,R),\trianglelefteq_{\mathbb{B}}) \to (\mathbb{B}_M,\subseteq)^d = (\mathbb{B}_M,\supseteq)\\ & (A,B)\mapsto B,\end{aligned}$$

which extracts the intent of a concept and which is an order isomorphism, between the poset of concepts and the *dual* of the complete lattice of their intents.

The join $(A_1,B_1)\vee_{\mathbb{B}}(A_2,B_2)$ of two concepts is obtained by observing that π_G^{-1} is a lattice isomorphism, so that

$$(A_1,B_1)\vee(A_2,B_2)=\pi_G^{-1}(A_1\vee_G A_2).$$

Its extent $\pi_G((A_1,B_1)\vee_{\mathbb{B}}(A_2,B_2))$ is thus $\pi_G(\pi_G^{-1}(A_1\vee_G A_2)) = A_1\vee_G A_2 = (A_1\cup A_2)''$; its intent is given by $\pi_M((A_1,B_1)\vee_{\mathbb{B}}(A_2,B_2)) = B_1\vee_M B_2 = B_1\cap B_2$ (recall that π_M maps over the dual of B_M, whose join is the intersection).

The meet of two concepts is similarly obtained from

$$(A_1,B_1)\wedge_{\mathbb{B}}(A_2,B_2)=\pi_G^{-1}(A_1\wedge_G A_2).$$

The extent $(A_1,B_1)\wedge_{\mathbb{B}}(A_2,B_2)$ is $A_1\cap A_2$ and its intent is $B_1\wedge_M B_2=(B_1\cup B_2)''$.

In summary, the join and the meet of two concepts are given by:

$$\begin{aligned}(A_1,B_1)\vee_{\mathbb{B}}(A_2,B_2) &= ((A_1\cup A_2)'',B_1\cap B_2)\\ (A_1,B_1)\wedge_{\mathbb{B}}(A_2,B_2) &= (A_1\cap A_2,(B_1\cup B_2)'').\end{aligned}$$

Examples. Consider the following two concepts, in the planet context:

$$\begin{aligned}C_1 &= \big(\{\text{Me, Ve, Ea, Ma}\},\{\text{small, near}\}\big)\\ C_2 &= \big(\{\text{Ea, Ma, Pl}\},\{\text{small, yes}\}\big).\end{aligned}$$

The intersection of the intents equals {small}, so that

$$\begin{aligned}C_1\vee_{\mathbb{B}}C_2 &= \big(\text{small}',\text{small}\big)=\\ &= \big(\{\text{Me, Ve, Ea, Ma, Pl}\},\text{small}\big).\end{aligned}$$

In this case, the extent of the join of the input concepts equals the union of their extents. The intersection of the extents of C_1 and C_2 is {Ea, Ma}, whose corresponding set of attributes {Ea, Ma}$'$ equals {small, near, yes}, so that

$$C_1\wedge_{\mathbb{B}}C_2=\big(\{\text{Ea, Ma}\},\{\text{small, near, yes}\}\big).$$

Consider now the following two concepts:

$$\begin{aligned}C_3 &= \big(\{\text{Ju, Sa}\},\{\text{large, far, yes}\}\big)\\ C_4 &= \big(\{\text{Ur, Ne}\},\{\text{medium, far, yes}\}\big).\end{aligned}$$

The intersection of their intents is {far, yes} to which the set of planets {far, yes}$'$={Ju, Sa, Ne, Ur, Pl} is associated, so that

$$C_3\vee_{\mathbb{B}}C_4=\big(\{\text{Ju, Sa, Ne, Ur, Pl}\},\ \{\text{far, yes}\}\big)$$

As it can be seen, the extent of this object is larger than the union of the extents of the input concepts, in that also Pluto shares the attributes in the intersection of their intents. In this example, the intersection of the extents of C_3 and C_4 is empty, so that the $C_3\wedge_{\mathbb{B}}C_4=\{\emptyset,\text{all attributes}\}$, i.e. it is the bottom of the concept lattice, which is, in this case, a vacuous concept.

Remarks.

1. The above derivation could have been made starting from π_M^{-1}, rather than from π_G^{-1}. The two options are completely symmetric.
2. Willingly, we have derived joins and meets of concepts in a rather abstract way, by using maps π_G and π_M. This has been done to present a powerful way of thinking, that avoids repeating "analytical" computations, turning the problem of solving "complex" problems into that of hierarchically composing simple maps. Once such maps are defined and their properties are worked out, computation becomes map composition.
3. Admittedly, however, the derivation of joins and meets between concepts can be made easier as follows. Once observed that the extent of $(A_1,B_1)\vee_{\mathbb{B}}(A_2,B_2)$ equals $(A_1\cup A_2)''$, we have that its intent is given by $((A_1\cup A_2)'')'=(A_1\cup A_2)'''=(A_1\cup A_2)'$. This is the set of attributes common to the objects in $A_1\cup A_2$, i.e. it is the intersection $B_1'\cap B_2'$. Similarly, for the meet.

E - The concept lattice reproduces exactly the input binary relation. The concept lattice provides an alternative representation of the input data, with no information loss, in that from it the input relation can be obtained. In this respect, FCA is not a dimensionality reduction tool, but a way to represent the input data in a more expressive way. To see this, let us introduce two maps:

$$\begin{aligned}\gamma \;&:\; G\to\mathbb{B}(G,M,R)\\ &:\; g\mapsto(g'',g')\end{aligned}$$

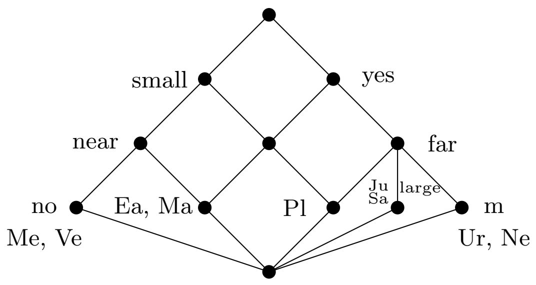

Figure 4: Concept lattice for the planet context. As explained in the following, the extent of a node (i.e. of a concept) is the collection of the planet labels of the labeled nodes in its downset and its intent is the collection of the attribute labels of the labeled nodes in its upset. For example, the unlabeled node in the middle of the Hasse diagram corresponds to concept ({Ea, Ma, Pl}, {small, yes}); the intent of the node labeled "Ea, Ma" is {small, near, yes}; the extent of the node labeled "far" is {Ju, Sa, Ur, Ne, Pl}, while its intent is {far, yes}. The bottom node is the vacuous concept (∅, all attributes), while the top node represents the similarly vacuous concept (all planets, ∅).

$$\begin{aligned} \mu &: M \to \mathbb{B}(G,M,R) \\ &: m \mapsto (m', m''). \end{aligned}$$

So γ returns the concept induced by the input object g and μ does the same, with the input attribute m. Now, if gRm, i.e. if object g has attribute m, then the following implications hold:

$$gRm \Rightarrow m \in g' \Rightarrow g'' \subseteq m' \Rightarrow \gamma(g) \trianglelefteq_{\mathbb{B}} \mu(m)$$

so that gRm implies the concept generated by m to dominate that generated by g. On the other hand:

$$\gamma(g) \trianglelefteq_{\mathbb{B}} \mu(m) \Rightarrow m'' \subseteq g' \Rightarrow gRm$$

so that if the concept generated by object g is dominated by that dominated by attribute m, then g enjoys m in the input relation. In summary, it holds $gRm \Leftrightarrow \gamma(g) \trianglelefteq_{\mathbb{B}} \mu(m)$ and one can recover the input relation, from the order structure of the concept lattice.

F - "Elementary" concepts $\gamma(g)$ and $\mu(m)$ generate the concept lattice. As single objects/attributes are the building blocks of G and M, so the concepts generated by them are the building blocks of the concept lattice, in the sense explained below. Let $(A,B) \in \mathbb{B}(G,M,R)$ be a concept. We know that $\bigcap_{g\in A} g' = A' = B$ and since it holds

$$\bigvee_{\mathbb{B}\atop g\in A} \gamma(g) = \left(\left(\bigcup_{g\in A} g'' \right)'', \bigcap_{g\in A} g' \right),$$

we see that $\bigvee_{\mathbb{B}_{g\in A}} \gamma(g)$ and (A,B) have the same intent and thus are the same concept. So, any concept is the join of the "elementary"concepts generated by the objects in its extent. The set $\{\gamma(g) : g \in G\}$ is said to be *join-dense*, as any element of the concept lattice can be recovered as the join af a suitable subset of it. Analogously, it can be proved that any concept of the context $\mathbb{B}(G,M,R)$ is the meet of the elementary concepts generated by the attributes of its intent, i.e.

$$(A,B) = \bigwedge_{\mathbb{B}\atop m\in B} \mu(m).$$

We thus say that the set $\{\mu(m) : m \in M\}$ is *meet-dense* in $\mathbb{B}(G,M,R)$.

The above results have a nice consequence on the way the concept lattice can be read, when it is represented as a Hasse diagram. Let (A,B) be a concept, with $g \in A$ an object in its extent and $m \in B$ an attribute in its intent. We have

$$g \in A \Leftrightarrow g'' \subseteq A \Leftrightarrow \gamma(g) \trianglelefteq_{\mathbb{B}} (A,B)$$

$$m \in B \Leftrightarrow m'' \subseteq B \Leftrightarrow (A,B) \trianglelefteq_{\mathbb{B}} \mu(m).$$

Thus, the concept (A,B) can be represented as follows:

$$(A,B) = (\{g; \gamma(g) \trianglelefteq_{\mathbb{B}} A\}, \{m : B \trianglelefteq_{\mathbb{B}} \mu(m)\}).$$

In practice, to label the Hasse diagram of a concept lattice, it is sufficient to:

1. Compute $\gamma(g)$ for any object in G and label with g'' the corresponding node.

2. Compute $\mu(m)$ for any attribute in M and label with m'' the corresponding node.

3. Once the nodes corresponding to single objects/attributes have been labeled, the extent and the intent of any other node (A,B) are obtained by collecting the labels of $\gamma(g)$-nodes in the downset of A and the labels of the $\mu(m)$-nodes in the upset of A.

Exercise (Davey and Priestley). Consider the concept lattice depicted in Figure 5, with objects $A,\ldots G$ and attributes 1,. . . ,7. The lattice is labeled in the wrong way; can you guess why? (*Hint*: check whether both the object-labeled and the attribute-labeled nodes are join-dense and meet-dense, respectively).

Exercise (Davey and Priestley). Let $G = \{A,B,C,D\}$ the set of object and $M = \{w,x,y,z\}$ the set of attributes, in a relation $R \subseteq G \times M$. Build a relation R such that the concept lattice associated to context (G,M,R) is isomorphic to that given in Figure 6. (*Hint*: (i) identify the smallest join-dense subset in the diagram, (ii) observe that the extents of some of its elements contain the extent of others and (iii) define the relation, in such a way that the function $\gamma(\cdot)$ maps correctly to these nodes. Alternatively, do the same starting from the smallest *meet-dense* subset and the $\mu(\cdot)$ function).

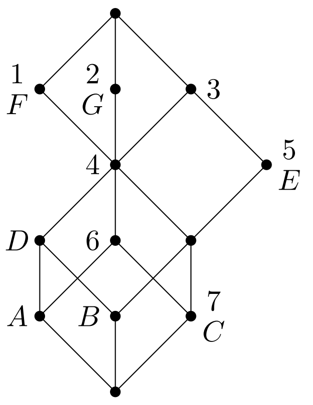

Figure 5: A wrongly labeled concept lattice.

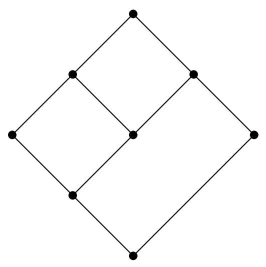

Figure 6: An unlabeled concept lattice

6 Use of the FCA: from concept lattices to implication logic

FCA produces a structural representation of the input relation as a lattice over the set of formal concepts; when its complexity is small, the lattice can be visualized and explored through as a Hasse diagram, so as to identify relevant groups of objects/attributes and relevant hierarchies among concepts, to be possibly interpreted based on the attributes that are removed or added along the chains they belong to. A different but typical use of FCA is to derive a *logic*, i.e. to identify the set of *implications* between subsets of attributes that hold in a context (e.g. with applications to recommendation systems).

Let $A, B \subset M$ be two subsets of the attribute set M. We say that A *implies* B within the formal context (G, M, R) (written $A \to B$), if every object $g \in G$ having all the attributes in A has also all the attributes in B, i.e. if $gRm \ \forall m \in A \to gRp \ \forall p \in B$.

Proposition. $A \to B$ if and only if (i) $B \subseteq A''$ or, equivalently, (ii) $A' \subseteq B'$.

Proof. Recall that A'' is the set of all of the attributes shared by all of the objects sharing all of the attributes in A. (i) $A \to B$ means that all of the objects sharing all of the attributes in A also share all of the attributes in B, i.e. that $B \subseteq A''$. In the opposite direction, $B \subseteq A''$ says that all of the objects sharing all of the attributes in A also share all of the attributes in B, i.e. that $A \to B$. (ii) By the properties of the polar map $'$, we have $B \subseteq A'' \Leftrightarrow A' = A''' \subseteq B'$.

■

So, to any element A of the powerset $\mathbb{P}(M)$ (i.e. to any subset of attributes) we can associate a set of *valid* implications, in (G, M, R), i.e the implications $A \to B$ for suitable Bs, with the convention that when $A' = \emptyset$, then $A \to M$ holds (*ex falso, quodlibet*: from false premises - here from combinations of attributes not shared by any object in the context - any property can be deduced).

The number of implications in the context (G, M, R) can be huge and impossible to manage in practice; one thus tries to reduce the number of implications to be considered. This can be done according to three main criteria.

1. *Small premises* - One can select/generate only those implications $A \to B$ where the cardinality $|A|$ of the premise is "small" (for example, by choosing a threshold τ_p and imposing $|A| \leq \tau_p$). The idea is that "simpler" implications are more interpretable and possibly more robust than implications requiring large sets of conditions (somehow as deep classification trees may suffer of overfitting).

2. *Large support* - The *support* of the implication $A \to B$ is the fraction of objects to which the implication applies. More formally, we define

$$supp(A \to B) = \frac{|A' \cap B'|}{|G|} \qquad (2)$$

   and put a threshold τ_s, selecting/generating the implications $A \to B$, such that $\tau_s \leq supp(A \to B)$. These *large support* implications are in general deemed more relevant, as they involve many objects and so are the "most" frequent ones.

   Remark: since $A \to B$ implies $A' \subseteq B'$, it holds $A' \cap B' = A'$ and so $supp(A \to B) = supp(A)$.

3. *Partial implications* - Insofar, only valid implications have been considered, i.e. implications that correspond to the condition $A' \subseteq B'$. However, we can also consider implications that are valid *to a certain extent*. For example, it can be the $A \to B$ does not hold in $B(G, M, R)$, because there are just a few objects that (i) share all of the attributes in A but (ii) lack some attributes in B. These objects prevent the implication to be valid (because they constitute *counterexamples* to it). Without those few objects, however, the implication would be valid and is thus termed *partial implication* (or also *association rule*). Partial implications are relevant since they can reveal hidden structures that, for example, could be masked by errors in the data. Indeed, mistakenly giving to/removing from an object attributes that actually it lacks/possesses may have a dramatic impact on the structure of the resulting concept

lattice and on the derived logic. So an implication $A \rightarrow B$ that holds true for "almost" all of the objects in A' can be inferred as "true", although formally it is not valid in the input context. In this spirit, we define the *confidence* of $A \rightarrow B$ as

$$conf(A \rightarrow B) = \frac{|A' \cap B'|}{|A'|} \tag{3}$$

and select/generate all the implications whose confidence is not lower than a threshold τ_c. We can combine the large confidence criterion and the large support criterion, finally searching for all the implications $A \rightarrow B$ such that $\tau_c \leq conf(A \rightarrow B)$ and $\tau_s \leq supp(A \rightarrow B)$. Notice that for partial implications $supp(A \rightarrow B) \leq supp(A)$, because $A' \not\subseteq B'$; notice also that the confidence of valid implications is equal to 1.

Exercise. Prove that $conf(A \rightarrow B) = conf(A'' \rightarrow (A \cup B)'')$.

Example. The simplest implication valid in the planet context are:

1. {no moon} $\rightarrow$ {near}.
2. {far} $\rightarrow$ {yes}.
3. {large} $\rightarrow$ {far, yes}.
4. {medium} $\rightarrow$ {far, yes}.

From these implications, others can be deduced, e.g. {$\neg$ small} $\rightarrow$ {far, yes} (where $\neg$ is the logical *not*), or {large} $\rightarrow$ {far}. So, from a smaller set (a *basis*) of implications, we can reconstruct the others. To show how to construct a basis for the set of all the valid implications of a context is, however, outside of the goal of these notes.

The implication {small} $\rightarrow$ {near} is not valid, because of Pluto that is small, but also the farthest planet in the solar system. The confidence of this implication, however, is quite high and equal to 4/5=0.8 (in fact, A'={Me, Ve, Ea, Ma, Pl} and B'={Me, Ve, Ea, Ma}).