Formal Concept Analysis: Themes and Variations for Knowledge Processing

Sergei O. Kuznetsov[1] and Amedeo Napoli[2]

[1] Department of Data Analysis and Artificial Intelligence
Faculty of Computer Science
National Research University Higher School of Economics,
Moscow, Russia
[2] LORIA (CNRS – INRIA Nancy Grand-Est – Université de Lorraine)
B.P. 239, 54506 Vandoeuvre les Nancy, France

skuznetsov@yandex.ru;Amedeo.Napoli@loria.fr

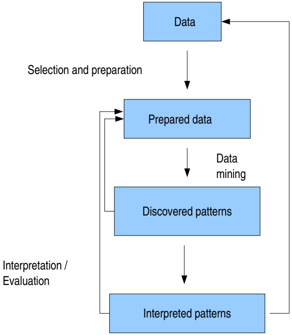
Tutorial on Formal Concept Analysis at IJCAI 2015

IJCAI 2015, Buenos Aires, July 27th 2015

Summary of the presentation

Knowledge Discovery in Databases (KDD)

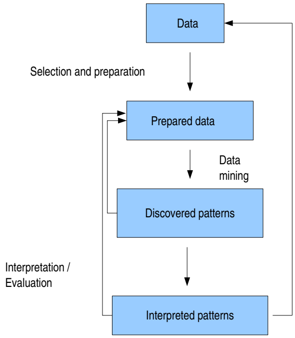
- The process of Knowledge Discovery in Databases (KDD) is applied on large volumes of complex data for discovering patterns which are significant and reusable.
- KDD is based on three main operations: data preparation, data mining, and interpretation of the extracted units.
- KDD is iterative, i.e. it can be replayed, and interactive, i.e. it is guided by an analyst.



Knowledge Discovery in Databases (KDD)

Data are diverse in nature and complexity:

- Boolean
- numbers
- symbols
- sequences (time series. . .)
- trees, graphs
- texts (images, speech. . .)
- web data (linked open data)
- ...

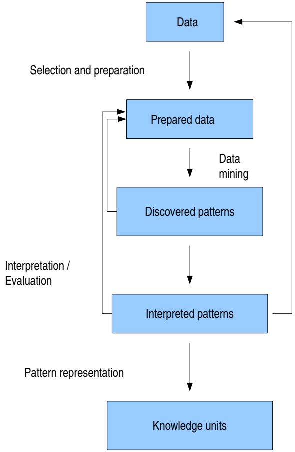


Several Approaches to KDD

- Databases: storage, access, querying, multi-dimensional databases, privacy, anonymisation.
- Artificial Intelligence: discovering actionable knowledge units, semantic aspects, embedding constraints and preferences (skylines), web data, linked open data.
- Algorithmics: scalability, distributed computing.
- Statistics and probablities: sampling, statistical processes, divergence, exploratory statistics, stochastic processes.
- Geometry: non Euclidean data spaces, metrics, geodesics.
- Visualization: interaction, interfaces,
- ...

Knowledge Discovery guided by Domain Knowledge

- The KDD process can be guided by domain knowledge at each step of the process for implementing Knowledge Discovery guided by Domain Knowledge (KDDK).
- KDDK extends KDD with a fourth step, i.e. representation and reuse of the extracted units.
- KDDK can be used for extending and updating a knowledge base: knowledge discovery and knowledge representation are complementary tasks.



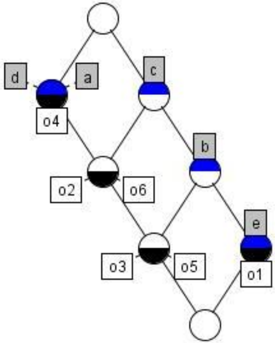
Knowledge is good for KDD

Domain knowledge can be useful for:

- Fixing thresholds in pattern mining.
- Computing similarity between objects (weighted features).
- Selecting patterns w.r.t. interest measures depending on domain knowledge (e.g. in chemistry using specific heteroatoms or functional groups), most-informative patterns, preferences.
- Using background knowledge for improving classification quality and accuracy (attribute representation).
- Dually, for efficiency reasons, reducing sets of attributes –feature selection– using classification for selecting groups of attributes.

KDD is good for Knowledge Engineering

- KDD is a learning process that can be used for knowledge engineering, information retrieval, problem solving. . .
- Formal concepts in a concept lattice can provide a basis for "partial" and "complete concepts".
- Implications also yield concept definitions.



KDDK: some application domains

- agronomy: analysis of landscape and of water quality.
- biology: resource retrieval, gene classification and similarity.
- chemistry and drug design: classification of molecules and reactions (meta-reactions).
- cooking: discovery of adaptation rules for CBR.
- medicine: text mining, management of patient trajectories.
- recommendation: biclustering, preference management (skylines).
- privacy: preserving privacy and anonymisation.
- network management: network analysis, attack prevention and prediction.
- . . .

An Ordinal View of KDDK

- How to combine discovery and representation of knowledge units?
- Classification is polymorphic and allows us to use partial orderings and their properties for dealing with KDD and KR.
- Revisiting Classification:
  - Discovery of classes for understanding data.
  - Organization of classes into a partial ordering.
  - Classification-based reasoning: recognizing the class of an individual and inserting a new class in a partial ordering
- Do we have such a "Swiss knife":
  Probably Formal Concept Analysis is of some help here. . .

What can we learn from a binary table and how?

Objects / Items	a	b	c	d	e
o1		x	x		x
o2	x		x	x	
o3	x	x	x	x	
o4	x			x	
o5	x	x	x	x	
o6	x		x	x	

The itemsets extracted from the binary table

Objects / Items	a	b	c	d	e
o1		x	x		x
o2	x		x	x	
o3	x	x	x	x	
o4	x			x	
o5	x	x	x	x	
o6	x		x	x	

The itemsets extracted from the binary table with the support threshold $\sigma_S = 2/6$ are:

- Itemsets of size 1: {a} (5/6), {b} (3/6), {c} (5/6), {d} (5/6).
- Itemsets of size 2: {ab} (2/6), {ac} (4/6), {ad} (5/6), {bc} (3/6), {bd} (2/6), {cd} (4/6).
- Itemsets of size 3: {abc} (2/6), {abd} (2/6), {acd} (4/6), {bcd} (2/6).
- Itemsets of size 4: {abcd} (2/6).

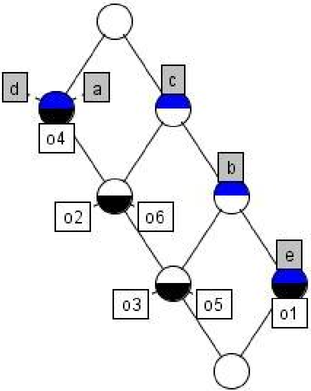
The association rules extracted from the binary table

Objects / Items	a	b	c	d	e
o1		x	x		x
o2	x		x	x	
o3	x	x	x	x	
o4	x			x	
o5	x	x	x	x	
o6	x		x	x	

The association rules extracted from the binary table with the thresholds $\sigma_S = 2/6$ (support) and $\sigma_C = 2/5$ (confidence):

- $\{a\} \longrightarrow \{b\}$ (2/6,2/5),
  $\{b\} \longrightarrow \{a\}$ (2/6,2/3),
  $\{a\} \longrightarrow \{c\}$ (4/6,4/5),
  $\{c\} \longrightarrow \{a\}$ (4/6,4/5) ...
- $\{ab\} \longrightarrow \{c\}$ (2/6,1),
  $\{ac\} \longrightarrow \{b\}$ (2/6,1/2),
  $\{bc\} \longrightarrow \{a\}$ (2/6,2/3),
  $\{c\} \longrightarrow \{ab\}$ (2/6,2/5),
  $\{b\} \longrightarrow \{ac\}$ (2/6,2/3),
  $\{a\} \longrightarrow \{bc\}$ (2/6,2/5) ...

The lattice associated to the binary table



Objects / Items	a	b	c	d	e
o1		x	x		x
o2	x		x	x	
o3	x	x	x	x	
o4	x			x	
o5	x	x	x	x	
o6	x		x	x	

FCA, Formal Concepts and Concept Lattices

- Marc Barbut and Bernard Monjardet, Ordre et classification, Hachette, 1970.
- Claudio Carpineto and Giovanni Romano, Concept Data Analysis: Theory and Applications, John Wiley & Sons, 2004.
- Bernhard Ganter and Rudolf Wille, Formal Concept Analysis, Springer, 1999.

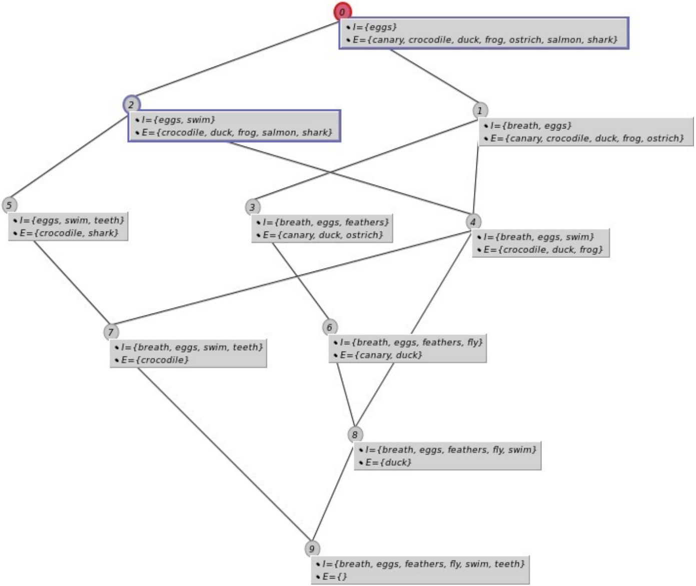
The FCA process

- The basic procedure of Formal Concept Analysis (FCA) is based on a simple representation of data, i.e. a binary table called a formal context.
- Each formal context is transformed into a mathematical structure called concept lattice.
- The information contained in the formal context is preserved.
- The concept lattice is the basis for data analysis.
  It is represented graphically to support analysis, mining, visualization, interpretation...

A concrete example

Animal/Features	eggs	feather	teeth	fly	swim	breath
ostrich	x	x				x
canary	x	x		x		x
duck	x	x		x	x	x
shark	x		x		x	
salmon	x				x	
frog	x				x	x
crocodile	x		x		x	x

A concrete example



The notion of a formal context

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

- $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ is called a formal context where $\mathtt{G}$ (*Gegenstände*) and $\mathtt{M}$ (*Merkmale*) are sets, and $\mathtt{I} \subseteq \mathtt{G} \times \mathtt{M}$ is a binary relation between $\mathtt{G}$ and $\mathtt{M}$.
- The elements of $\mathtt{G}$ are the objects, while the elements of $\mathtt{M}$ are the attributes, $\mathtt{I}$ is the incidence relation of the context $(\mathtt{G}, \mathtt{M}, \mathtt{I})$.

Two derivation operators

For $A \subseteq G$ and for $B \subseteq M$:

- $' : \wp(G) \longrightarrow \wp(M)$
  $' : A \longrightarrow A'$
  $A' = \{m \in M / (g, m) \in I \text{ for all } g \in A\}$
- $' : \wp(M) \longrightarrow \wp(G)$ with
  $' : B \longrightarrow B'$
  $B' = \{g \in G / (g, m) \in I \text{ for all } m \in B\}$

Computing the images of sets of objects and attributes

$\{g2\}' = \{m1, m3, m4\}$:

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

- $A' = \{m \in M / (g, m) \in I \text{ for all } g \in A\}$
- $B' = \{g \in G / (g, m) \in I \text{ for all } m \in B\}$

Computing the images of sets of objects and attributes

$\{m3\}' = \{g1, g2, g3, g5, g6\}$:

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

- $A' = \{m \in M / (g, m) \in I \text{ for all } g \in A\}$
- $B' = \{g \in G / (g, m) \in I \text{ for all } m \in B\}$

Computing the images of sets of objects and attributes

$\{g3, g5\}' = \{m1, m2, m3, m4\}$:

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

Computing the images of sets of objects and attributes

$\{m3, m4\}' = \{g2, g3, g5, g6\}$:

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

The derivation operators and the Galois connection

- $' : \wp(G) \longrightarrow \wp(M)$ with $A \longrightarrow A'$
- $' : \wp(M) \longrightarrow \wp(G)$ with $B \longrightarrow B'$
- These two applications induce a Galois connection between $\wp(G)$ and $\wp(M)$ when sets are ordered by set inclusion.

A Galois connection is defined as follows:

- Let $(P, \leq)$ and $(Q, \leq)$ be two partially ordered sets.
- A pair of maps $\phi : P \longrightarrow Q$ and $\psi : Q \longrightarrow P$ is called a Galois connection between P and Q if:
- (i) $p_1 \leq p_2 \Longrightarrow \phi(p_1) \geq \phi(p_2)$ (decreasing).
- (ii) $q_1 \leq q_2 \Longrightarrow \psi(q_1) \geq \psi(q_2)$ (decreasing).
- (iii) $p \leq \psi \circ \phi(p)$ and $q \leq \phi \circ \psi(q)$ (increasing).

Iterating the derivation

- $A' = \{m \in M / (g, m) \in I \text{ for all } g \in A\}$
- $B' = \{g \in G / (g, m) \in I \text{ for all } m \in B\}$
- The derivation operators can be composed, i.e. iterated: starting with a set $A \subseteq G$, we obtain that A' is a subset of M.
- Applying the second operator on this set, we get $(A')'$, or A'' for short, which is a set of objects.
- Continuing, we obtain A''', A'''', and so on.

Iterating the derivation

Objects / Attributes	m1	m2	m3	m4	m5
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

- $\{g3\}'' = \{\mathtt{m1}, \mathtt{m2}, \mathtt{m3}, \mathtt{m4}\}' = \{g3, g5\}$
- $\{g1, g3, g5\}'' = \{\mathtt{m2}, \mathtt{m3}\}' = \{g1, g3, g5\}$
- $\{\mathtt{m3}, \mathtt{m4}\}'' = \{g2, g3, g5, g6\}' = \{\mathtt{m1}, \mathtt{m3}, \mathtt{m4}\}$
- $\{\mathtt{m3}\}'' = \{g1, g2, g3, g5, g6\}' = \{\mathtt{m3}\}$

Properties of the derivation operators

- $A' = \{m \in M / (g, m) \in I \text{ for all } g \in A\}$
- $B' = \{g \in G / (g, m) \in I \text{ for all } m \in B\}$

The derivation operators $'$ satisfy the following rules:

- $A_1 \subseteq A_2 \Longrightarrow A_2' \subseteq A_1'$ (decreasing).
- $B_1 \subseteq B_2 \Longrightarrow B_2' \subseteq B_1'$ (decreasing).

- $A \subseteq A''$ and $A' = A'''$ (increasing and fix point).
- $B \subseteq B''$ and $B' = B'''$ (increasing and fix point).

Examples

G / M	m1	m2	m3	m4	m5
g1		×	×		×
g2	×		×	×	
g3	×	×	×	×	
g4	×			×	
g5	×	×	×	×	
g6	×		×	×	

- $A_1 \subseteq A_2 \Longrightarrow A_2' \subseteq A_1'$
- $B_1 \subseteq B_2 \Longrightarrow B_2' \subseteq B_1'$
- $A \subseteq A''$ and $A' = A'''$
- $B \subseteq B''$ and $B' = B'''$

Other properties of the derivation operators

For $A_1, A_2 \subseteq G$, and dually for $B_1, B_2 \subseteq M$, we have:

- $A_1 \subseteq A_2 \Longrightarrow A_1'' \subseteq A_2''$ (increasing).
- $B_1 \subseteq B_2 \Longrightarrow B_1'' \subseteq B_2''$ (increasing).
- $(A'')'' = A''$ (fix point).
- $(B'')'' = B''$ (fix point).

Formal concept

Given a formal context $(\mathtt{G}, \mathtt{M}, \mathtt{I})$:

- $\mathtt{A}' = \{\mathtt{m} \in \mathtt{M}/(\mathtt{g}, \mathtt{m}) \in \mathtt{I} \texttt{ for all } \mathtt{g} \in \mathtt{A}\}$
- $\mathtt{B}' = \{\mathtt{g} \in \mathtt{G}/(\mathtt{g}, \mathtt{m}) \in \mathtt{I} \texttt{ for all } \mathtt{m} \in \mathtt{B}\}$
- $(\mathtt{A}, \mathtt{B})$ is a formal concept of $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ iff:
  $\mathtt{A} \subseteq \mathtt{G}$, $\mathtt{B} \subseteq \mathtt{M}$, $\mathtt{A}' = \mathtt{B}$, and $\mathtt{A} = \mathtt{B}'$.
- $\mathtt{A}$ is the extent and $\mathtt{B}$ is the intent of $(\mathtt{A}, \mathtt{B})$.
- The mappings $\mathtt{A} \longrightarrow \mathtt{A}''$ and $\mathtt{B} \longrightarrow \mathtt{B}''$ are closure operators.

The Galois connection and the closure operators

More generally, a closure operator on a set S is a map κ such that:

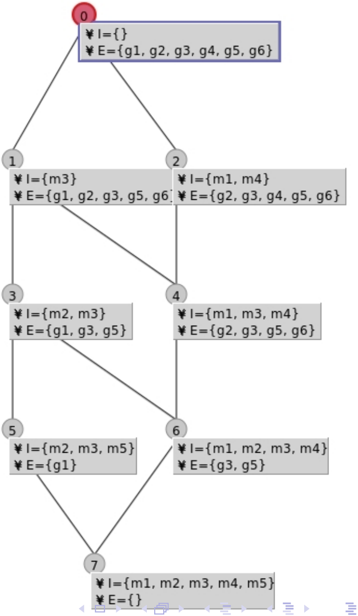
- $\kappa : \wp(S) \longrightarrow \wp(S)$
- For all $S_1, S_2 \subseteq S$:
  - (i) $S_1 \subseteq \kappa(S_1)$ (extensivity: $S_1 \subseteq S_1^{''}$)
  - (ii) $S_1 \subseteq S_2$ then $\kappa(S_1) \subseteq \kappa(S_2)$ (monotonicity: $S_1 \subseteq S_2 \Longrightarrow S_1^{''} \subseteq S_2^{''}$)
  - (iii) $\kappa(\kappa(S_1)) = \kappa(S_1)$ (idempotency: $(S_1^{''})^{''} = S_1^{''}$)
- S_i is a closed set whenever $\kappa(S_i) = S_i$ or $S_i^{''} = S_i$.
- The composition operators $''$, i.e. the composition of $'$ and $'$, are closure operators.

The concept lattice

- Formal concepts can be partially ordered by:
  $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2$ (dually $B_2 \subseteq B_1$).
- The set $\underline{\mathfrak{B}}(G, M, I)$ of all formal concepts of (G, M, I) with this order is a complete lattice called the concept lattice of (G, M, I).
- Every complete lattice has a top (or unit) element denoted by $\top$, and a bottom (or zero) element denoted by $\bot$.

The concept lattice

G / M	m1	m2	m3	m4	m5
g1		×	×		×
g2	×		×	×	
g3	×	×	×	×	
g4	×			×	
g5	×	×	×	×	
g6	×		×	×	



The basic theorem of FCA

- The concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice in which the infimum and the supremum are given by:
- $\bigwedge_{k \in K}(A_k, B_k) = (\bigcap_{k \in K} A_k, (\bigcup_{k \in K} B_k)'')$
- $\bigvee_{k \in K}(A_k, B_k) = ((\bigcup_{k \in K} A_k)'', \bigcap_{k \in K} B_k)$
- Note: an intersection of closed sets is a closed set but a union of closed sets is not necessarily a closed set.

The structure of the concept lattice

The object concept

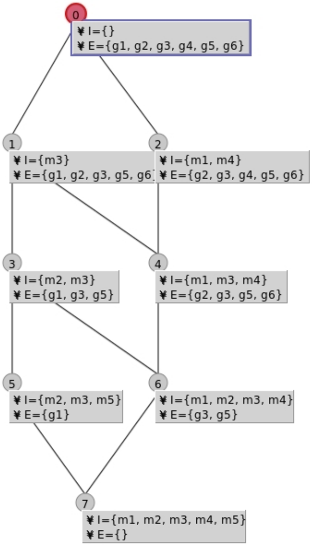
- The name of the object g is attached to the "lower half" of the corresponding object concept `$(g) = (\{g\}'', \{g\}')$.
- The object concept of an object $g \in \mathtt{G}$ is the concept $(\{g\}'', \{g\}')$ where $\{g\}'$ is the object intent $\{\mathtt{m} \in \mathtt{M}/\mathtt{gIm}\}$ of g.
- The object concept of g, denoted by `(g), is the smallest concept (for the lattice order) with g in its extent.

The object concept

- Example:

- `(g4) $= (\{g4\}'', \{g4\}') = (\{g2, g3, g4, g5, g6\}, \{m1, m4\})$

- `(g1) $= (\{g1\}'', \{g1\}') = (\{g1\}, \{m2, m3, m5\})$

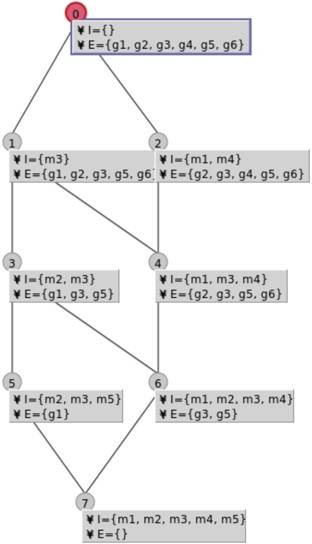


The attribute concept

- The name of the attribute $\mathtt{m}$ is located to the "upper half" of the corresponding attribute concept $\mu(\mathtt{m}) = (\{\mathtt{m}\}', \{\mathtt{m}\}'')$.
- Correspondingly, the attribute concept of an attribute $\mathtt{m} \in \mathtt{M}$ is the concept $(\{\mathtt{m}\}', \{\mathtt{m}\}'')$ where $\{\mathtt{m}\}'$ is the attribute extent $\{\mathtt{g} \in \mathtt{G}/\mathtt{gIm}\}$ of $\mathtt{m}$.
- The attribute concept of $\mathtt{m}$, denoted by $\mu(\mathtt{m})$ is the largest concept (for the lattice order) with $\mathtt{m}$ in its intent.

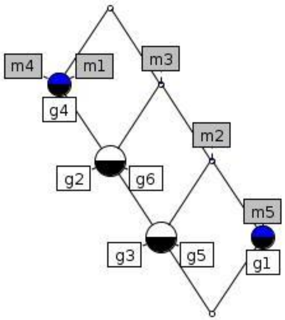
The attribute concept

- Example:
- $\mu(\mathtt{m1}) = (\{\mathtt{m1}\}', \{\mathtt{m1}\}'') = (\{\mathtt{g2}, \mathtt{g3}, \mathtt{g4}, \mathtt{g5}, \mathtt{g6}\}, \{\mathtt{m1}, \mathtt{m4}\})$
- $\mu(\mathtt{m1}) = \mu(\mathtt{m4})$
- $\mu(\mathtt{m2}) = (\{\mathtt{m2}\}', \{\mathtt{m2}\}'') = (\{\mathtt{g1}, \mathtt{g3}, \mathtt{g5}\}, \{\mathtt{m2}, \mathtt{m3}\})$



The reduced labeling

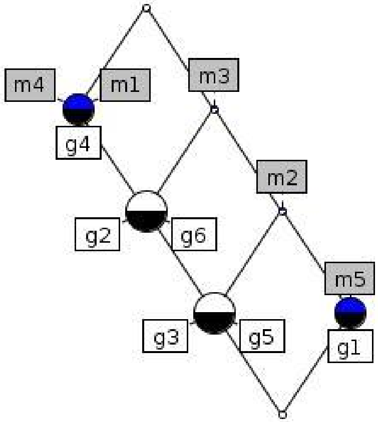
G / M	m1	m2	m3	m4	m5
g1		×	×		×
g2	×		×	×	
g3	×	×	×	×	
g4	×			×	
g5	×	×	×	×	
g6	×		×	×	



- A reduced labeling may be used allowing that each object and each attribute is entered only once in a diagram.
- Reduced labeling: intuitively, the attributes are "at the highest" and the objects are "at the lowest".

The reduced labeling

G / M	m1	m2	m3	m4	m5
g1		×	×		×
g2	×		×	×	
g3	×	×	×	×	
g4	×			×	
g5	×	×	×	×	
g6	×		×	×	



- For any concept (A, B) we have:
- $g \in A \iff `(g) \leq (A, B)$
- $m \in B \iff (A, B) \leq \mu(m)$

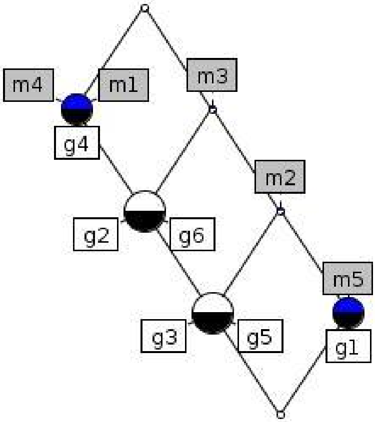
An extent is an ideal (down-set)

- Let $(P, \leq)$ be an ordered set.
  A subset $Q \subseteq P$ is an order ideal or a down-set if $x \in Q$ and $y \leq x$ imply that $y \in Q$.
- $\downarrow Q = \{y \in P / \exists x \in Q : y \leq x\}$
  $\downarrow x = \{y \in P / y \leq x\}$
- The extent of an arbitrary concept can be found as the set of objects in the principal ideal generated by the concept.
- For example, the extent of a concept X is composed of all objects which are in the extents of the descendants Y of X.

An intent is a filter (up-set)

- Let $(P, \leq)$ be an ordered set.
  A subset $Q \subseteq P$ is an order filter or an up-set if $x \in Q$ and $x \leq y$ imply that $y \in Q$.
- $\uparrow Q = \{y \in P / \exists x \in Q : x \leq y\}$
  $\uparrow x = \{y \in P / x \leq y\}$
- The intent of an arbitrary concept can be found as the set of objects in the principal filter generated by the concept.
- For example, the intent of a concept X is composed of all attributes which are in the intents of the ascendants Y of X.

Ideals and filters

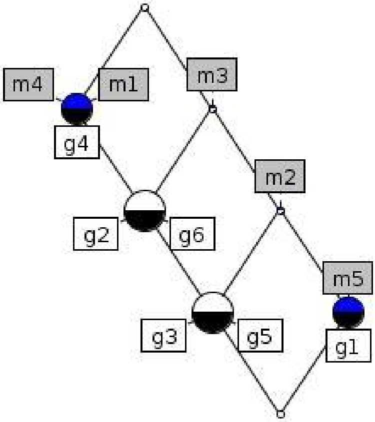


- The extent of concept C_1 is composed of g_4 and all objects which are in the extents of the descendants C_i of C_1, i.e. g_2, g_6 and then g_3, g_5.
- The intent of a concept C_5 is composed of all attributes which are in the intents of the ascendants C_i of C_5, i.e. m_2, m_1, m_4 and m_3.

Types of attributes

- Introducing and attribute: an attribute α is introduced in a concept C when it is not present in any ascendant (super-concept) of C, i.e. the concept C corresponds to the attribute concept of α (sometimes called the introducer of α).
- Inheriting an attribute: an attribute α is inherited by a concept C when it is already present in an ascendant of C, i.e. C is lower for the lattice order than the attribute-concept or introducer of α.

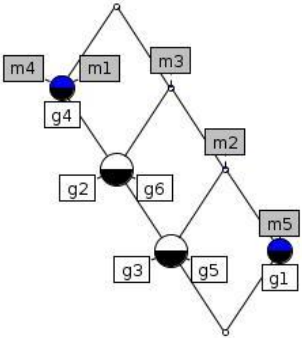
Types of attributes (example)



- m3 is an attribute introduced in the concept $(g12356, m3)$, m1 and m4 are attributes introduced in the concept $(g23456, m14)$,
- m2 is an attribute introduced in the concept $(g135, m23)$.
- m3 is an attribute inherited by $(g135, m23)$, m1, m3, and m4, are attributes inherited by $(g2356, m134)$, and so on.

Extracting rules from a concept lattice

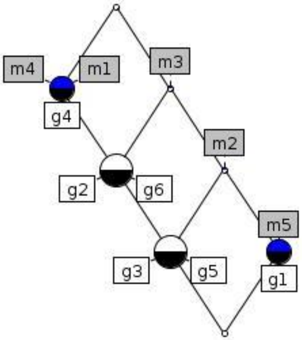
Mutual implications between attributes having the same attribute-concept



- Attributes having the same attribute-concept or introducer are equivalent:
- for example $\mathtt{m1} \longleftrightarrow \mathtt{m4}$ for $(\mathtt{g23456}, \mathtt{m14})$.

Extracting rules from a concept lattice (continued)

Introduced attributes imply inherited attributes



- When an attribute α is introduced, it implies every inherited attribute in the attribute-concept of α:
- for example $\mathtt{m2} \longrightarrow \mathtt{m3}$ for $(\mathtt{g135}, \mathtt{m23})$ and $\mathtt{m5} \longrightarrow \mathtt{m23}$ for $(\mathtt{g1}, \mathtt{m235})$.

Scaling

Conceptual scaling

- The formal context is the basic data type of Formal Concept Analysis.
- However data are often given in form of a many-valued context.
- Many-valued contexts are translated to one-valued context via conceptual scaling.
- But this is not automatic and some arbitrary choices have to be made.
- Examples of scalings:
  - Nominal: $\mathbb{K} = (\mathbb{N}, \mathbb{N}, =)$
  - Ordinal: $\mathbb{K} = (\mathbb{N}, \mathbb{N}, \leq)$
  - Interordinal: $\mathbb{K} = (\mathbb{N}, \mathbb{N}, \leq \cup \geq)$

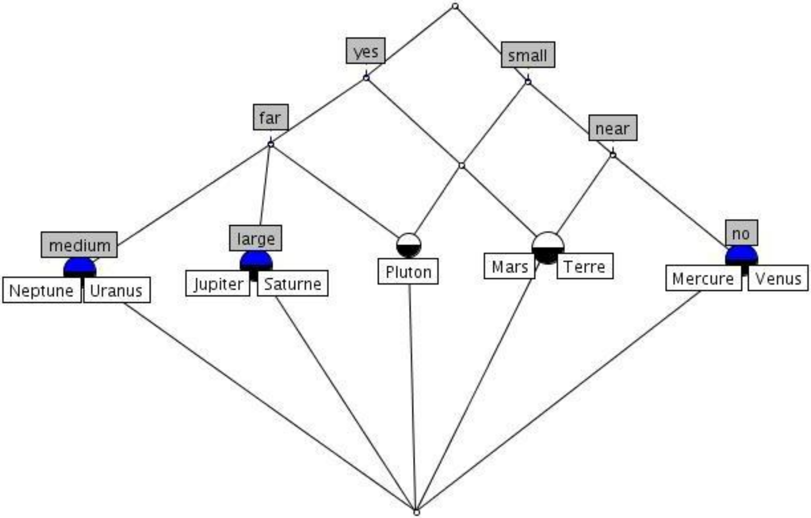
The example of the context of planets

Planet	Size	Distance to Sun	Moon(s)
Jupiter	large	far	yes
Mars	small	near	yes
Mercury	small	near	no
Neptune	medium	far	yes
Pluto	small	far	yes
Saturn	large	far	yes
Earth	small	near	yes
Uranus	medium	far	yes
Venus	small	near	no

The context of planets after nominal scaling

Planet	Size			Distance to Sun		Moon(s)	
	small	medium	large	near	far	yes	no
Jupiter			x		x	x	
Mars	x			x		x	
Mercury	x			x			x
Neptune		x			x	x	
Pluto	x				x	x	
Saturn			x		x	x	
Earth	x			x		x	
Uranus		x			x	x	
Venus	x			x			x

The concept lattice of planets (after scaling)



A numerical example

G / M	m1	m2	m3
g1	1	3	4
g2	2	2	3
g3	4	1	1
g4	3	2	1

Nominal Scaling:

G / M	m1=1	m1=2	m1=4	m2=1	m2=2	m2=3	m3=1	m3=3	m3=4
g1	×					×			×
g2		×			×			×	
g3			×	×			×		
g4		×			×		×		

Interordinal Scaling:

G / M	m1.lt.1	m1.gt.1	m1.lt.2	m1.gt.2	m1.lt.3	m1.gt.3	m1.lt.4	m1.gt.4	m2.lt.1
g1	×	×	×		×		×		
g2		×	×	×	×		×		
g3		×		×		×	×	×	×
g4		×	×	×	×		×		

A simple algorithm for discovering formal concepts and building the concept lattice

An algorithm for computing the formal concepts

- A rectangle in a binary table corresponds to a pair $(\mathtt{X}, \mathtt{Y})$ –where $\mathtt{X}$ denotes an extension and $\mathtt{Y}$ denotes an intension– only contains crosses x.
  Such an extension and intension are not necessarily extents and intents respectively.
- A rectangle $(\mathtt{X}, \mathtt{Y})$ is contained in another rectangle $(\mathtt{X}_1, \mathtt{Y}_1)$ whenever $\mathtt{X} \subseteq \mathtt{X}_1$ and $\mathtt{Y} \subseteq \mathtt{Y}_1$.
- A rectangle $(\mathtt{X}, \mathtt{Y})$ is maximal when it is not included in any other rectangle: any rectangle $(\mathtt{X}_1, \mathtt{Y}_1)$ containing a maximal rectangle $(\mathtt{X}, \mathtt{Y})$ is such that $\mathtt{X}_1$ and/or $\mathtt{Y}_1$ contain at least a "void place", i.e. a place without a cross x.

An algorithm for constructing the concept lattice

Set $\mathtt{L}^1 = \{(\mathtt{X_i}, \mathtt{Y_i}), \mathtt{i} = 1, \ldots, \mathtt{n}\}$ ($\mathtt{n}$ = number of objects)
$\mathtt{L}^1$ = set of rectangles $(\mathtt{X_i}, \mathtt{Y_i})$ of size 1 with $\mathtt{Y_i} = \mathtt{X_i'}$
Set $\mathtt{k} = 1$
While the size of $\mathtt{L^k}$ is strictly greater than 1 **do**
Set $\mathtt{L^{k+1}} = \emptyset$
  For all $\mathtt{i} < \mathtt{j}$ index of elements of $\mathtt{L^k}$ which are not marked **do**
  $\mathtt{Y_{ij}} = \mathtt{Y_i} \cap \mathtt{Y_j}$
  If $\mathtt{Y_{ij}} \neq \emptyset$ **then**
    If $\mathtt{Y_{ij}} \in \mathtt{L^{k+1}}$ **then** $\mathtt{X_{ij}} = \mathtt{X_{ij}} \cup \mathtt{X_j}$
      $\mathtt{L^{k+1}} = \mathtt{L^{k+1}} \cup (\mathtt{X_{ij}}, \mathtt{Y_{ij}})$
      If $\mathtt{Y_{ij}} = \mathtt{Y_i}$ **then** mark $(\mathtt{X_i}, \mathtt{Y_i})$ in $\mathtt{L^k}$ **endif**
      If $\mathtt{Y_{ij}} = \mathtt{Y_j}$ **then** mark $(\mathtt{X_j}, \mathtt{Y_j})$ in $\mathtt{L^k}$ **endif**
    endif
  endfor
endwhile
$\mathtt{L}$ is the set of elements which are not marked in the set of $\mathtt{L^k}$.

An example of construction of a concept lattice (1)

G / M	m1 (a)	m2 (b)	m3 (c)	m4 (d)	m5 (e)
g1		x	x		x
g2	x		x	x	
g3	x	x	x	x	
g4	x			x	
g5	x	x	x	x	
g6	x		x	x	

For better readability: $\mathtt{M} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$

The rectangles of size 1:

$\mathtt{L}^1 = \{(\mathtt{g1}, \mathtt{bce}), (\mathtt{g2}, \mathtt{acd}), (\mathtt{g3}, \mathtt{abcd}), (\mathtt{g4}, \mathtt{ad}), (\mathtt{g5}, \mathtt{abcd}), (\mathtt{g6}, \mathtt{acd})\}$

An example of construction of a concept lattice (2)

- The rectangles of size 1:
- $\mathtt{L}^1 =$
  $\{(g1, bce), (g2, acd), (g3, abcd), (g4, ad), (g5, abcd), (g6, acd)\}$
- Build the rectangles of size 2 by union of rectangles of size 1:
- $\mathtt{L}^2 = \emptyset$
  $\mathtt{i} = 1, \ldots, 5$;
  $\mathtt{i} = 2, \ldots, 6$;
  $\mathtt{i} < \mathtt{j}$
- $\mathtt{Y}_{12} = \mathtt{c}$; $\mathtt{L}^2 = \{(g12, c)\}$
- $\mathtt{Y}_{13} = \mathtt{bc}$; $\mathtt{L}^2 = \{(g12, c), (g13, bc)\}$
- $\mathtt{Y}_{14} = \emptyset$
- $\mathtt{Y}_{15} = \mathtt{bc}$; $\mathtt{X}_{13} = \mathtt{X}_{13} \cup \mathtt{X}_5$ and $\mathtt{L}^2 = \{(g12, c), (g135, bc)\}$
- $\mathtt{Y}_{16} = \mathtt{c}$; $\mathtt{X}_{12} = \mathtt{X}_{12} \cup \mathtt{X}_6$ and $\mathtt{L}^2 = \{(g126, c), (g135, bc)\}$

An example of construction of a concept lattice (3)

- The rectangles of size 2 (continued):
- $Y_{23} = \text{acd}$; $L^2 = \{(\text{g126}, \text{c}), (\text{g135}, \text{bc}), (\text{g23}, \text{acd})\}$
  as $Y_{23} = Y_2$ mark $(\text{g2}, \text{acd})$ in L^1
- $Y_{24} = \text{ad}$; $L^2 = \{(\text{g126}, \text{c}), (\text{g135}, \text{bc}), (\text{g23}, \text{acd}), (\text{g24}, \text{ad})\}$
  as $Y_{24} = Y_4$ mark $(\text{g4}, \text{ad})$ in L^1
- $Y_{25} = \text{acd}$;
  $L^2 = \{(\text{g126}, \text{c}), (\text{g135}, \text{bc}), (\text{g235}, \text{acd}), (\text{g24}, \text{ad})\}$
- $Y_{26} = \text{acd}$;
  $L^2 = \{(\text{g126}, \text{c}), (\text{g135}, \text{bc}), (\text{g2356}, \text{acd}), (\text{g24}, \text{ad})\}$
  as $Y_{26} = Y_6$ mark $(\text{g6}, \text{acd})$ in L^1

An example of construction of a concept lattice (4)

- The rectangles of size 2 (continued):
- $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g24}, \mathtt{ad})\}$
- $\mathtt{Y}_{35} = \mathtt{abcd}$;
  $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g24}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$
  as $\mathtt{Y}_{35} = \mathtt{Y}_3$ mark $(\mathtt{g3}, \mathtt{abcd})$ in $\mathtt{L}^1$
  as $\mathtt{Y}_{35} = \mathtt{Y}_5$ mark $(\mathtt{g5}, \mathtt{abcd})$ in $\mathtt{L}^1$
- $\mathtt{Y}_{36} = \mathtt{acd}$; do nothing as
  $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g24}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$
- $\mathtt{Y}_{45} = \mathtt{ad}$;
  $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g245}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$
- $\mathtt{Y}_{46} = \mathtt{ad}$;
  $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g2456}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$
- $\mathtt{Y}_{56} = \mathtt{ad}$; do nothing as
  $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g2456}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$

An example of construction of a concept lattice (5)

- The rectangles of size 2 (end):
- $\mathtt{L}^2 = \{(\mathtt{g126}, \mathtt{c}), (\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g2456}, \mathtt{ad}), (\mathtt{g35}, \mathtt{abcd})\}$
- $\mathtt{L}^1 = \{(\mathtt{g1}, \mathtt{bce})\}$ (all other elements are marked)
- The rectangles of size 3 and more:
- $\mathtt{L}^3 = \emptyset$

An example of construction of a concept lattice (6)

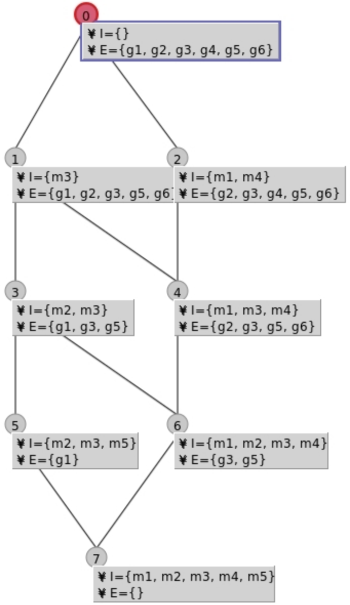
- The rectangles of size 3 and more:
- $\mathtt{L}^2 = \{(\mathrm{g126}, \mathrm{c}), (\mathrm{g135}, \mathrm{bc}), (\mathrm{g2356}, \mathrm{acd}), (\mathrm{g2456}, \mathrm{ad}), (\mathrm{g35}, \mathrm{abcd})\}$
- $\mathtt{L}^3 = \emptyset$
- $\mathtt{Y}_{12} = \mathtt{c}$ in $\mathtt{L}^2$; $\mathtt{L}^3 = \{(\mathrm{g12356}, \mathrm{c})\}$
- $\mathtt{Y}_{13} = \mathtt{c}$ in $\mathtt{L}^2$; then do nothing
- $\mathtt{Y}_{14} = \mathtt{c}$ in $\mathtt{L}^2$; then do nothing
- $\mathtt{Y}_{15} = \emptyset$
- $\mathtt{Y}_{23} = \mathtt{c}$ in $\mathtt{L}^2$; then do nothing
- $\mathtt{Y}_{24} = \emptyset$
- $\mathtt{Y}_{25} = \mathtt{bc}$ in $\mathtt{L}^2$; then do nothing
- $\mathtt{Y}_{34} = \mathtt{ad}$ in $\mathtt{L}^2$; $\mathtt{L}^3 = \{(\mathrm{g12356}, \mathrm{c}), (\mathrm{g23456}, \mathrm{ad})\}$
  as $\mathtt{Y}_{34} = \mathtt{Y}_4$ mark $(\mathrm{g2456}, \mathrm{ad})$ in $\mathtt{L}^2$
- $\mathtt{Y}_{35} = \mathtt{acd}$ in $\mathtt{L}^2$; then do nothing
- $\mathtt{Y}_{45} = \mathtt{ad}$ (in $\mathtt{L}^2$) ; then do nothing

An example of construction of a concept lattice (5)

The list of maximal rectangles:

- $\mathtt{L}^1 = \{(\mathtt{g1}, \mathtt{bce})\}$
- $\mathtt{L}^2 = \{(\mathtt{g135}, \mathtt{bc}), (\mathtt{g2356}, \mathtt{acd}), (\mathtt{g35}, \mathtt{abcd})\}$
- $\mathtt{L}^3 = \{(\mathtt{g12356}, \mathtt{c}), (\mathtt{g23456}, \mathtt{ad})\}$

An example of construction of a concept lattice (6)



Relational Concept Analysis

- Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli and Petko Valtchev. Relational Concept Analysis: Mining Concept Lattices From Multi-Relational Data, Annals of Mathematics and Artificial Intelligence, 67(1):81–108, 2013.
- Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli and Petko Valtchev. Soundness and Completeness of Relational Concept Analysis, in Proceedings of ICFCA 2013, LNAI 7880, Springer, pages 228-243, 2013.
- Victor Codocedo and Amedeo Napoli. A proposition for combining pattern structures and relational concept analysis, in Proceedings of ICFCA 2014, LNAI 8478, Springer, pages 96-111, 2014.

Introducing Relational Concept Analysis (RCA)

- The objective of RCA is to take into account relations between objects within the FCA framework.
- The RCA process relies on the following main points:
  - a relational model which can be seen as a kind of entity-relationship model,
  - a conceptual scaling process allowing to represent relations between objects as relational attributes,
  - an iterative process for designing a concept lattice where concept intents include binary and relational attributes.
- The RCA process provides "relational structures" that can be represented as ontology concepts within a knowledge representation formalism such as description logics (DLs).

The RCA data model

- The RCA data model relies on a so-called relational context family denoted by $\mathcal{RCF} = (\mathbf{K}, \mathbf{R})$, where:
- $\mathbf{K}$ is a set of formal contexts $\mathcal{K}_i = (\mathtt{G_i}, \mathtt{M_i}, \mathtt{I_i})$,
- $\mathbf{R}$ is a set of relations $\mathtt{r_k} \subseteq \mathtt{G_i} \times \mathtt{G_j}$, where $\mathtt{G_i}$ and $\mathtt{G_j}$ are sets of objects from the formal contexts $\mathcal{K}_i$ and $\mathcal{K}_j$.
- A relation $\mathtt{r} \subseteq \mathtt{G_i} \times \mathtt{G_j}$ has a domain and a range where:
- $\mathtt{dom(r)} = \mathtt{G_i}$ and $\mathtt{ran(r)} = \mathtt{G_j}$.

An example

Suppose that we have a context Papers $\times$ Topics where:

- Papers denotes a set of papers –indexed from "a" to "ℓ"–
- Topics denotes a set of three attributes, namely "lt" for "lattice theory", "mmi" for "man-machine interface", and "se" for "software engineering".
- There are two relations:
  - cites $\subseteq$ Papers $\times$ Papers indicates that a paper is citing another paper,
  - develops $\subseteq$ Papers $\times$ Papers indicates that a paper is developing another paper.

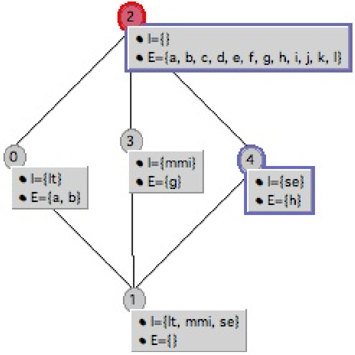
The initial relational context

	lt	mmi	se	a	b	g	h	c	d	i	j
a	x										
b	x										
c				x		x					
d					x		x				
e								x			
f									x		
g		x									
h			x								
i				x							
j					x						
k										x	
ℓ											x

- Relational context: $(\mathbf{K}, \mathbf{R}) = (\mathcal{K}_0, \{\texttt{cites}, \texttt{develops}\})$
- $\mathbf{K} = \mathcal{K}_0 = (\texttt{Papers}, \texttt{Topics}, \texttt{I})$
- $\mathbf{R} = \{\texttt{cites}, \texttt{develops}\}$

The $\mathcal{L}_0$ concept lattice built from formal context $\mathcal{K}_0$

	lt	mmi	se
a	x		
b	x		
c			
d			
e			
f			
g		x	
h			x
i			
j			
k			
ℓ			



Introducing relational scaling

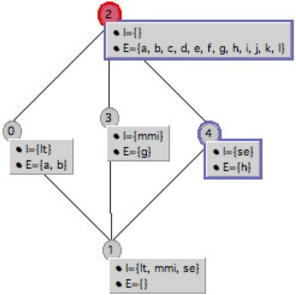
- The first step consists in building an initial concept lattice $\mathcal{L}_0$ from the the initial context $\mathcal{K}_0$ using FCA algorithms.
- The second step takes into account relations $\mathtt{r}(\mathtt{o_i}, \mathtt{o_j})$ for building a new context $\mathcal{K}_1$:
  - $\mathtt{r}(\mathtt{o_i}, \mathtt{o_j})$ means that object $\mathtt{o_i} \in \mathtt{G_i}$ is related through relation $\mathtt{r}$ with object $\mathtt{o_j} \in \mathtt{G_j}$,
  - then a relational attribute of the form $\exists \mathtt{r}.\mathtt{C_k}$ is associated to object $\mathtt{o_i}$ in $\mathcal{K}_1$, where $\mathtt{C_k}$ is **any** concept instantiating $\mathtt{o_j}$ in $\mathcal{L}_0$.
- When all relations between objects have been examined, the next context $\mathcal{K}_1$ is completed and a new concept lattice $\mathcal{L}_1$ is built accordingly.

The relational context $\mathcal{K}_0$

	lt	mmi	se	a	b	g	h	c	d	i	j
a	x										
b	x										
c				x		x					
d					x		x				
e								x			
f									x		
g		x									
h			x								
i				x							
j					x						
k										x	
ℓ											x

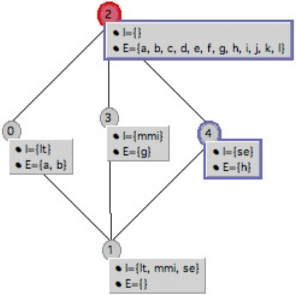
- c cites a and g, d cites b and h,
- i cites a and j cites b.

Relational scaling in $\mathcal{L}_0$



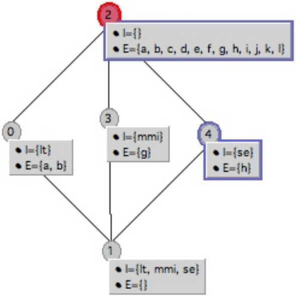
- Object c is in relation with objects a and g through relation cites.
- Object a is in the extent of concepts C_0 and C_2 in $\mathcal{L}_0$ while object g is in the extent of concepts C_3 and C_2 in $\mathcal{L}_0$.
- Thus, object c is given three new relational attributes, namely $\exists$cites:C_0, $\exists$cites:C_2, and $\exists$cites:C_3.

Relational scaling in $\mathcal{L}_0$



- Object d is in relation with objects b and h through relation cites.
- Object b is in the extent of concepts C_0 and C_2 in $\mathcal{L}_0$ while object h is in the extent of concepts C_4 and C_2 in $\mathcal{L}_0$.
- Thus, object d is given three new relational attributes, namely $\exists$cites:C_0, $\exists$cites:C_2, and $\exists$cites:C_4.

Relational scaling in $\mathcal{L}_0$



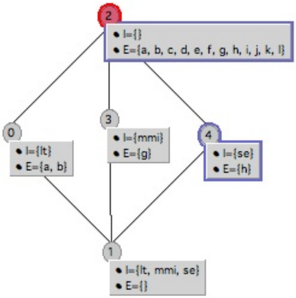
- Object i is in relation with object a through relation cites.
- Object a is in the extent of concepts C_0 and C_2 in $\mathcal{L}_0$.
- Thus, object i is given two new relational attributes, i.e. $\exists$cites:C_0 and $\exists$cites:C_2.
- In the same way, j in relation with b through cites is given the two relational attributes $\exists$cites:C_0 and $\exists$cites:C_2.

The relational context $\mathcal{K}_0$

	lt	mmi	se	a	b	g	h	c	d	i	j
a	x										
b	x										
c				x		x					
d					x		x				
e								x			
f									x		
g		x									
h			x								
i				x							
j					x						
k										x	
ℓ											x

- e develops c and f develops i,
- k develops j and ℓ develops j.

Relational scaling in $\mathcal{L}_0$

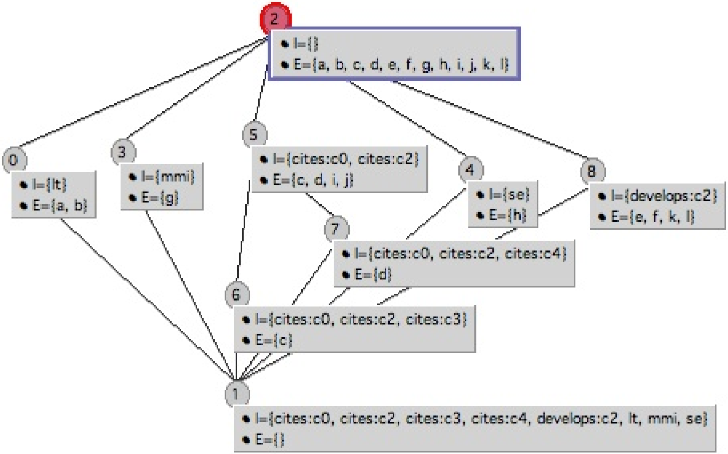


- The same process is applied to develops:
- e is in relation through develops with c (in the extent of C_2),
- f is in relation through develops with d (in the extent of C_2),
- k is in relation through develops with i (in the extent of C_2),
- ℓ is in relation through develops with j (in the extent of C_2),
- The four objects e, f, k, and ℓ, are given the relational attribute $\exists$develops:C_2.

The new relational context $\mathcal{K}_1$

	lt	mmi	se	cites:c2	cites:c0	cites:c3	cites:c4	develops:c2
a	x							
b	x							
c				x	x	x		
d				x	x		x	
e								x
f								x
g		x						
h			x					
i				x	x			
j				x	x			
k								x
ℓ								x

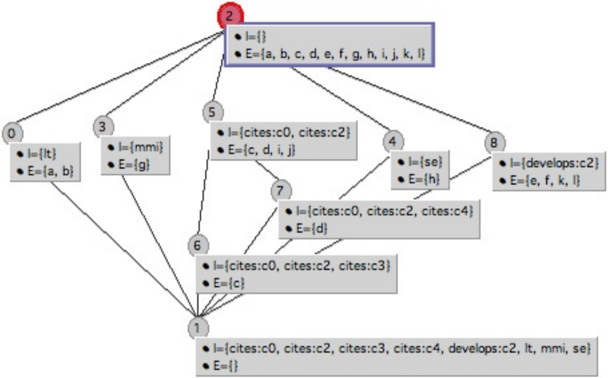
The concept lattice $\mathcal{L}_1$



Following the construction of the lattice

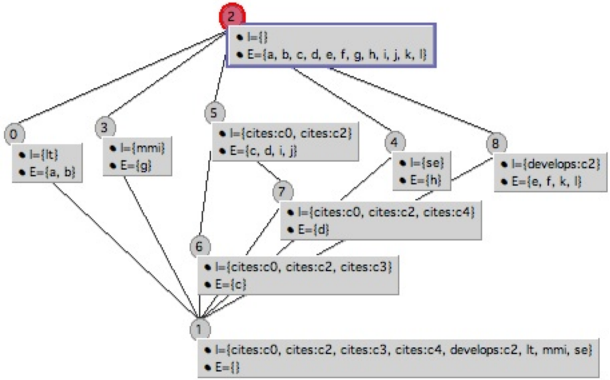
- The numbering of concepts is kept all along the whole process.
- The relational scaling process is continued as soon as the "instantiation" of one of the objects which is in the range of a relation has changed.
- In $\mathcal{L}_1$, no instantiation of objects in the range of the `cites` relation is changed: thus, there will be no other modification for the `cites` and relational scaling is done.
- Actually: object `c` is in relation with `a` and `g` while object `d` is in relation with `b` and `h`, but the instantiations of `a`, `g`, `b`, and `h` are not changed.
- Object `i` is in relation with `a` and `j` with `b`, but the instantiation of `a` and `b` are not changed.

Relational scaling in $\mathcal{L}_1$



- The object e develops c whose instantiation has changed, i.e. c is in the extents of concepts C_2, C_5, and C_6.
- Thus object e is in addition given the relational attributes $\exists$develops:C_5 and $\exists$develops:C_6.

Relational scaling in $\mathcal{L}_1$

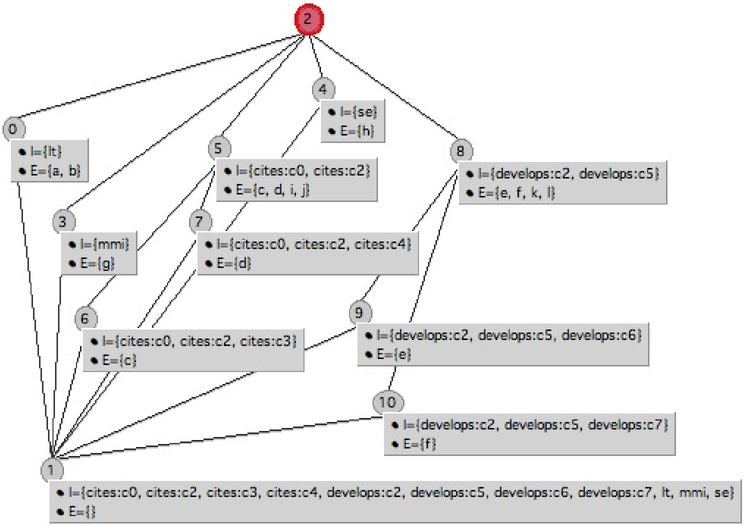


- Object `f` `develops` `d` whose instantiation is in the extent of concepts C_2, C_5, and C_7.
- Object `k` `develops` `i` whose instantiation is in the extent of concepts C_2 and C_5.
- Object ℓ `develops` `j` whose instantiation is in the extent of concepts C_2 and C_5.

The new relational context $\mathcal{K}_2$

	lt	mmi	se	develops:c2	develops:c5	develops:c6	develops:c7
a	x						
b	x						
c							
d							
e				x	x	x	
f				x	x		x
g		x					
h			x				
i							
j							
k				x	x		
ℓ				x	x		

The concept lattice $\mathcal{L}_2$



The completion of the RCA process

- Relational scaling is still applied for `cites` and `develops` but the final context and the associated concept lattice are obtained after the second step.
- More generally, relational scaling is applied and either there are modifications in the instantiations, i.e. RCA process continues, or there are no more modifications, i.e. RCA fix-point is reached.
- The relational scaling process reaches a fix-point when no more changes in instantiations occur, i.e. the final relational lattice is reached and the relational scaling process terminates.

Three forms of relational attributes

- Existential scaling $\exists \texttt{r.C}$: $\texttt{r(o)} \cap \texttt{Extent(C)} \neq \emptyset$
- Universal scaling $\forall \texttt{r.C}$: $\texttt{r(o)} \subseteq \texttt{Extent(C)}$
- Universal-Existential scaling $\forall\exists$r.C: $\texttt{r(o)} \subseteq \texttt{extent(C)}$ and $\texttt{r(o)} \neq \emptyset$
- With relational scaling, the homogeneity of concept descriptions is kept: all attributes –included relational attributes– are considered as binary attributes.
- Standard FCA algorithms for building concept lattices can be straightforwardly reused.

From a relational concept lattice to an ontology schema

- The concepts of the final concept lattice can be represented within a DL formalism such as $\mathcal{ALE}$ for designing an ontology schema supported by the lattice.
- Some problems about knowledge representation are arising for representing binary and relational attributes.
- Binary attributes can be represented as atomic concepts.
- Thanks to the semantics associated with relational scaling and operators, roles can be attached to defined concepts in a "natural" way using a construction such as $\exists \texttt{r.C}$.

Pattern Structures

- Bernhard Ganter and Sergei O. Kuznetsov. Pattern Structures and Their Projections, in Proceedings of the 9th International Conference on Conceptual Structures (ICCS-2001), LNCS 2120, pages 129–142, 2001.
- Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli and Sébastien Duplessis. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis, Information Science, 181(10):1989–2001, 2011.
- Mehdi Kaytoue, Sergei O. Kuznetsov and Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis, in Proceedings of 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), Barcelona, Spain, 2011.

Computing similarity between descriptions

Intersection considered as a similarity operator:

- $\cap$ behaves like a *similarity operator*:

$$\{m_1, m_2\} \cap \{m_1, m_3\} = \{m_1\}$$

	m_1	m_2	m_3
g_1	×		×
g_2	×	×	
g_3		×	×
g_4		×	×
g_5	×	×	×

- $\cap$ induces a partial ordering relation $\subseteq$ as follows:

$$S_1 \cap S_2 = S_1 \iff S_1 \subseteq S_2$$
$$\{m_1\} \cap \{m_1, m_2\} = \{m_1\} \iff \{m_1\} \subseteq \{m_1, m_2\}$$

- $\cap$ has the properties of a meet $\sqcap$ in a semi lattice, i.e. a commutative, associative and idempotent operation:

$$c \sqcap d = c \iff c \sqsubseteq d$$

The definition of a Pattern Structure

A pattern structure $(G, (D, \sqcap), \delta)$ is composed of:

- G a set of *objects*,
- $(D, \sqcap)$ a semi-lattice of descriptions or patterns,
- δ a mapping such as $\delta(g) \in D$ describes object g.

The Galois connection for $(G, (D, \sqcap), \delta)$ is defined as:

- The maximal description representing the similarity of a set of objects:
$$A^{\square} = \sqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G$$
- The maximal set of objects sharing a given description:
$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$$

Standard FCA as a Pattern Structure $(G, (D, \sqcap), \delta)$

Considering a standard formal context (G, M, I):

- G is the set of *objects*,
- $(D, \sqcap)$ corresponds to $\wp(M)$ where M is the set of attributes.
- $\delta(g)$ corresponds to the description of g in terms of attributes.

	m_1	m_2	m_3
g_1	$\times$		$\times$
g_2	$\times$	$\times$	
g_3		$\times$	$\times$
g_4		$\times$	$\times$
g_5	$\times$	$\times$	$\times$

The Galois connection:

- $A^{\square} = \sqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G$
- $\{g_1, g_2\}' = g_1' \cap g_2' = \{m_1, m_2\} \cap \{m_1, m_3\} = \{m_1\}$
- $d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$
- $\{m_1\}' = \{g_i \in G | \{m_1\} \subseteq g_i'\} = \{g_1, g_2, g_5\}$

From FCA to Pattern Structures

- A *formal context* (G, M, I) is based on a set of objects G, a set of attributes M, and a binary relation $I \subseteq G \times M$.
- Two derivation operators are defined as follows, $\forall A \subseteq G, B \subseteq M$:

$$A' = \{m \in M | \forall g \in A, (g, m) \in I\}$$

$$B' = \{g \in G | \forall m \in B, (g, m) \in I\}$$

- A formal concept (A, B) verifies $A' = B$ and $A = B'$.
- Formal concepts are partially ordered w.r.t. inclusion of extents (or dually of intents):

$$(A_1, B_1) \leq (A_2, B_2) \text{ iff } A_1 \subseteq A_2$$

- A *pattern structure* $(G, (\mathcal{D}, \sqcap), \delta)$ is based on a set of objects G, a meet semi-lattice of *object descriptions* $(\mathcal{D}, \sqcap)$, and a mapping $\delta : G \longrightarrow \mathcal{D}$ which associates a description to each object.
- Two derivation operators are defined as follows, $\forall A \subseteq G, d \in (\mathcal{D}, \sqcap)$:

$$A^{\square} = \sqcap_{g \in A} \delta(g)$$

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\}$$

- A formal concept (A, d) verifies $A^{\square} = d$ and $A = d^{\square}$
- Pattern concepts are partially ordered w.r.t. inclusion of extents (or dually inclusion of intents):

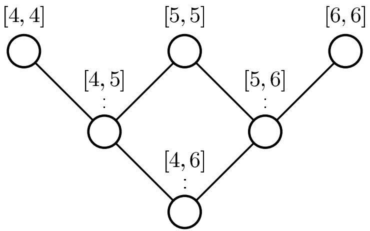
$$(A_1, d_1) \leq (A_2, d_2) \text{ iff } A_1 \subseteq A_2$$

Interval Pattern Structure

- Let D be a set of intervals with integer bounds (for simplicity),
- let $\sqcap$ be a meet operator defined on D as the convex hull of intervals:

$$\begin{aligned} [a_1, b_1] \sqcap [a_2, b_2] &= [min(a_1, a_2), max(b_1, b_2)] \\ [4, 5] \sqcap [5, 5] &= [4, 5] \end{aligned}$$

$$\begin{aligned} [a_1, b_1] \sqsubseteq [a_2, b_2] &\iff [a_2, b_2] \subseteq [a_1, b_1] \\ [4, 5] \sqsubseteq [5, 5] &\iff [5, 5] \subseteq [4, 5] \end{aligned}$$



Interval Pattern Structures for classifying a numerical context

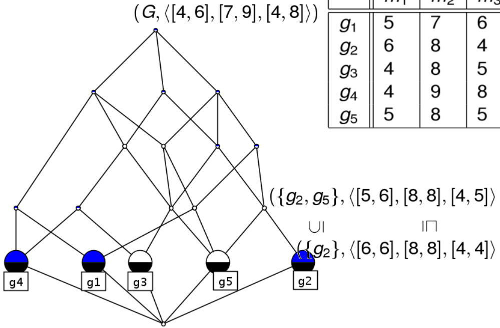
	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

$$\begin{aligned} \{g_1, g_2\}^{\square} &= \sqcap_{g \in \{g_1, g_2\}} \delta(g) \\ &= \langle 5, 7, 6 \rangle \sqcap \langle 6, 8, 4 \rangle \\ &= \langle [5, 6], [7, 8], [4, 6] \rangle \end{aligned}$$

$$\begin{aligned} \langle [5, 6], [7, 8], [4, 6] \rangle^{\square} &= \{g \in G | \langle [5, 6], [7, 8], [4, 6] \rangle \sqsubseteq \delta(g)\} \\ &= \{g_1, g_2, g_5\} \end{aligned}$$

$(\{g_1, g_2, g_5\}, \langle [5, 6], [7, 8], [4, 6] \rangle)$ **is a pattern concept**

Interval pattern concept lattice



	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

- Highest concepts: largest extents and smallest intents (but the largest intervals),
- Lowest concepts: smallest extents and largest intents (but the smallest intervals),
- Problem: efficient pattern mining.

Some applications of pattern structures

- Text mining with tree-based pattern structures.
  Artuur Leeuwenberg, Aleksey Buzmakov, Yannick Toussaint and Amedeo Napoli Exploring Pattern Structures of Syntactic Trees for Relation Extraction, in ICFCA 2015, LNAI 9113, 2015.
- Mining sequential data for analyzing patient trajectories (with selection of interesting concepts using stability measure).
  Aleksey Buzmakov, Elias Egho, Nicolas Jay, Sergei O. Kuznetsov, Amedeo Napoli and Chedy Raïssi. On Mining Complex Sequential Data by Means of FCA and Pattern Structures, in International Journal of General Systems (To be published), 2015.
- Information Retrieval and Recommendation.
  Victor Codocedo, Ioanna Lykourentzou and Amedeo Napoli. A semantic approach to concept lattice-based information retrieval, in Annals of Mathematics and Artificial Intelligence, 72:169–195, 2014.
- Discovery of Functional Dependencies.
  Jaume Baixeries, Amedeo Napoli and Mehdi Kaytoue. Characterizing functional dependencies in formal concept analysis with pattern structures, Annals of Mathematics and Artificial Intelligence, 72:129-149, 2014.
- Biclustering and Triadic Analysis.
  Mehdi Kaytoue, Sergei O. Kuznetsov, Juraj Macko and Amedeo Napoli. Biclustering meets triadic concept analysis, Annals of Mathematics and Artificial Intelligence, 70(1-2):55-79, 2014.
  Victor Codocedo and Amedeo Napoli. Lattice-based biclustering using Partition Pattern Structures, in Proceedings of ECAI 2014, IOS Press, pages 213-218, 2014.

Heterogeneous Pattern Structures

- Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli and Petko Valtchev. Relational Concept Analysis: Mining Concept Lattices From Multi-Relational Data, Annals of Mathematics and Artificial Intelligence, 67(1):81–108, 2013.
- Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli and Petko Valtchev. Soundness and Completeness of Relational Concept Analysis, in Proceedings of ICFCA 2013, LNAI 7880, Springer, pages 228-243, 2013.
- Victor Codocedo and Amedeo Napoli. A proposition for combining pattern structures and relational concept analysis, in Proceedings of ICFCA 2014, LNAI 8478, Springer, pages 96-111, 2014.

Latent Semantic Indexing

- Let us consider a document-term matrix, i.e. the representation of a set of documents w.r.t. a set of attributes through a set of weights (representation of documents as vectors in a vector space).
- Latent Semantic Indexing (LSI) is based on the Singular Value Decomposition process of a matrix.
- LSI searches for the lower-rank approximation of the document-term matrix.

Latent Semantic Indexing

	patient	laparoscopy	scan	user	medicine	response	time	MRI	practice	complication	arthroscopy	infection
g_1	0.25	0.25	0.25	0	0	0	0	0	0	0.25	0	0
g_2	0	0	0.16	0.16	0.16	0.16	0.16	0	0.16	0	0	0
g_3	0	0.25	0	0.25	0.25	0	0	0.25	0	0	0	0
g_4	0.3	0	0	0	0.3	0	0	0.3	0	0	0	0
g_5	0	0	0	0.3	0	0.3	0.3	0	0	0	0	0
g_6	0	0	0	0	0	0	0	0	0.5	0	0.5	0
g_7	0	0	0	0	0	0	0	0	0	0.5	0.5	0
g_8	0	0	0	0	0	0	0	0	0	0.3	0.3	0.3
g_9	0	0	0	0	0	0	0	0	0	0	0.5	0.5

Table : Document-term matrix A.

LSI and lower-rank approximation of a matrix

The SVD Process:

$$A_{(9\times 12)} = U_{(9\times 9)} \cdot \Sigma_{(9\times 12)} \cdot V^T_{(12\times 12)} \quad (1)$$

$$\tilde{A}_{(9\times 12)} = U_{(9\times k)} \cdot \Sigma_{(k\times k)} \cdot V^T_{(k\times 12)} \quad \text{(with } k \ll min(9,12)\text{)} \quad (2)$$

$$A \sim \tilde{A} \quad (3)$$

$$\tilde{A} \cdot \tilde{A}^T = U_{(9\times k)} \cdot \Sigma_{(k\times k)} \cdot V^T_{(k\times 12)} \cdot V_{(12\times k)} \cdot \Sigma^T_{(k\times k)} \cdot U^T_{(k\times 9)} \quad (4)$$

$$\tilde{A} \cdot \tilde{A}^T = (U_{(9\times k)} \cdot \Sigma_{(k\times k)}) \cdot (U_{(9\times k)} \cdot \Sigma_{(k\times k)})^T \quad (5)$$

Classifying documents

	k1	k2
g_1	0.118	-0.238
g_2	0.046	-0.271
g_3	0.014	-0.413
g_4	0.014	-0.368
g_5	0.008	-0.277
g_6	0.519	0.002
g_7	0.603	-0.017
g_8	0.469	0.02
g_9	0.588	0.092

Table : Documents in 2 LVs. (k=2)

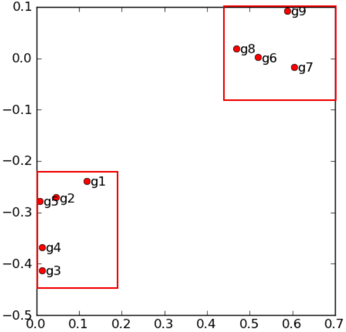


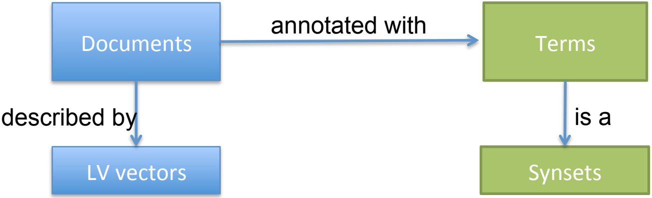
Figure : Graphical representation of documents as points in a 2 dimensional LV space.

What about the semantics?

- Latent variables are abstractions.
- A given LV or a convex region in a LV-space can represent a topic, but this lacks a proper characterization.
- It is not possible to introduce external domain knowledge.
- FCA provides a formal characterization of concepts through the dual extent/intent descriptions.
- FCA allows the introduction of external knowledge sources through object relations (RCA).
- FCA allows the analysis of complex data such as convex regions in a vector space (interval pattern structures).

A possible scenario

Can we relate abstractions such as LVs to external domain knowledge?



In fact, this scenario fits with the Relational Concept Analysis process.

Relational Concept Analysis (RCA)

RCA describes an iterative scaling process to obtain a family of related concept lattices from a relational context family.

	k1	k2
g_1	×	
g_2		×
g_3		×
g_4	×	×
g_5		×
g_6		×
g_7	×	
g_8	×	×
g_9	×	

Table : Formal Context $\mathcal{K}_1 = (G_1, M_1, I_1)$

	patient	laparoscopy	scan	user	medicine	response	time	MRI	practice	complication	arthroscopy	infection
g_1	×	×	×							×		
g_2			×	×	×	×	×		×			
g_3		×		×	×			×				
g_4	×				×			×				
g_5				×		×	×					
g_6									×		×	
g_7										×	×	
g_8										×	×	×
g_9											×	×

Table : Relational Context aw $= (G_1, G_2, I_{aw})$

	Person	Surgery	Illness	Artefact	Event	Activity
patient	×					
laparoscopy		×				×
scan				×		
user	×					
medicine						×
response					×	
time					×	
MRI				×		
practice						×
complication			×			
arthroscopy		×				×
infection			×			

Table : Formal Context $\mathcal{K}_2 = (G_2, M_2, I_2)$

Relational Concept Analysis (RCA)

- A relational context family (RCF) is composed by:
  - A set of formal contexts $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$.
  - A set of binary relations $\mathbf{R} = \{\mathtt{aw}\}$.
- A relational context is interpreted through the relation $\mathtt{aw} : \mathtt{G}_1 \rightarrow \mathtt{G}_2$, where $\mathtt{dom}(\mathtt{aw}) = \mathtt{G}_1$ and $\mathtt{ran}(\mathtt{aw}) = \mathtt{G}_2$.
- A set of relational attributes is built w.r.t. (G_1, M_1, I_1), (G_2, M_2, I_2), and the relation $\mathtt{aw}$.
- The relational scaling process applied in (G_1, M_1, I_1) assigns a set of relational attributes to an object $\mathtt{g} \in \mathtt{G}_1$ whenever $\mathtt{aw}(\mathtt{g}) \cap \mathtt{extent}(\mathtt{C}) \neq \emptyset$ ($\exists$ quantifier), where $\mathtt{C}$ is a concept for (G_2, M_2, I_2).
- e.g. $\mathtt{g}_1$ is described by $\exists \mathtt{aw}.\mathtt{C}$ iff $\mathtt{aw}(\mathtt{g}_1) \cap \mathtt{extent}(\mathtt{C}) \neq \emptyset$.

Relational Concept Lattice

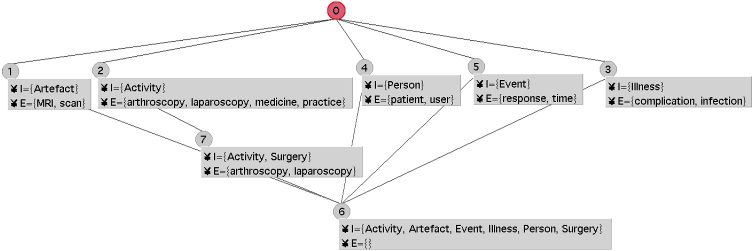


Figure : Concept Lattice for Taxonomic annotations $\mathcal{L}_2$.

RCA - Relational Scaling

$$\mathtt{aw}(\mathtt{g}_1) \cap \mathtt{extent}(\mathtt{C4}) = \{\mathtt{patient}, \mathtt{user}\}$$

$$\Rightarrow \mathcal{K}_1^{(1)} = (\mathtt{G}_1, \mathtt{M}_1 \cup \{\mathtt{aw}\colon \mathtt{C4}\}, \mathtt{I}_1 \cup \{(\mathtt{g}_1, \mathtt{aw}\colon \mathtt{C4})\})$$

Relational Concept Analysis

- Formal concepts in $\mathcal{K}_1^{(1)}$ have intents which relate LV with taxonomical annotations in $\mathcal{K}_2$.
- Nevertheless, $\mathcal{K}_1$ is a many-valued context. Convex regions in a LV-space are better described with interval pattern structures.
- An adaptation should be done when we apply relational scaling in a many-valued formal context.

Heterogeneous formal context

	D		Pr						
	k1	k2	aw : C1	aw : C2	aw : C3	aw : C4	aw : C5	aw : C6	aw : C7
g_1	0.118	-0.238	×	×	×	×			×
g_2	0.046	-0.271	×	×		×	×		
g_3	0.014	-0.413	×	×		×			×
g_4	0.014	-0.368	×	×		×			
g_5	0.008	-0.277				×	×		
g_6	0.519	0.002		×	×				×
g_7	0.603	-0.017		×	×				×
g_8	0.469	0.02		×	×				×
g_9	0.588	0.092		×	×				×

Table : Heterogeneous formal context.

Problems

- Objects are described by heterogeneous patterns mixing values and binary attributes.
- It becomes necessary to define a proper pattern structure which is able to deal with heterogeneous object descriptions.

Proposition

In the example:

- $(\mathtt{G}_1, (\mathtt{D}, \sqcap), \delta)$ is an interval pattern structure of documents described by convex regions in a LV space.
- $\mathcal{K}_2$ is a formal context of terms and taxonomical annotations (Wordnet synsets).
- $\mathtt{aw} : \mathtt{G}_1 \rightarrow \mathtt{G}_2$ relates documents with a set of annotations (terms).
- An heterogeneous pattern concept (hp-concept) $(\mathtt{A}, \mathtt{h})$ describes in its intent a relation between a convex region in the LV space and a set of taxonomical annotation.
- The set of all hp-concepts generates a set of "labeled clusters" in the LV space.

Proposition

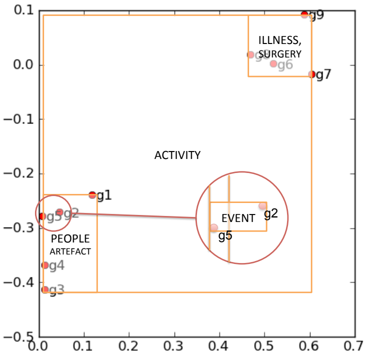


Figure : Labeled document clusters using association rules from the hp-lattice with magnification on documents g_2 and g_5.

Dealing with big and complex data

FCA: dealing with complex and big data

- Vertical dimensionality reduction (sampling): reduction of the set of objects.
- Horizontal dimensionality reduction (attribute selection): reduction of the set of attributes (dimensionality reduction can be guided by domain knowledge).
- Factorization and Intelligent Sampling: computing "factors" from large tables (LSA, LDA, LSI) for facilitating classification and interpretation.
- Projections for building simplified descriptions and simplified concept lattices.
- Iceberg lattices for considering concept lattice "level by level" (w.r.t. support of intents).
- Stability measure for selecting interesting concepts in large concept lattices.

Measures for selecting interesting concepts in "big data"

- Projections allow to consider only intents which can be of interest, e.g. the longest subsequences in sequence classification.
- The stability measure allows to consider and to rank the most stable concepts:

$$Stab(C) := \frac{|\{x \in \wp(extent(C)) \mid x' = intent(C)\}|}{|\wp(extent(C))|}$$

- Aleksey Buzmakov, Sergei O. Kuznetsov and Amedeo Napoli. Scalable Estimates of Stability, in Proceedings of ICFCA 2014, LNAI 8478, Springer, pages 157-172, 2014.

Using pattern mining algorithms for building concept lattices

- Computing closed itemsets (FCIs) with e.g. Charm algorithm ("vertical search").
- Computing minimal generators (FGIs) with reverse pre-order traversal.
- Associating closed itemsets and generators to form equivalence classes.
- Computing precedence links between equivalence classes with hypergraph techniques (transversals).
- Laszlo Szathmary, Petko Valtchev, Amedeo Napoli, Robert Godin, Alix Boc and Vladimir Makarenkov. A fast compound algorithm for mining generators, closed itemsets, and computing links between equivalence classes, Annals of Mathematics and Artificial Intelligence, 70(1–2):81–105, 2014.

FCA: dealing with complex and big data

- Anytime algorithms: compute a partial solution that is completed w.r.t. remaining resources.
- Parallelization of algorithms for dealing with large and distributed data.
- Combining numerical and symbolic methods: e.g. clustering, SVM and FCA.
- Interactivity and Visualization: visualization and replay remain essential in KDDK and in the interpretation of concept lattices.

"Big Users" for Big Data Applications

- Mining Social Networks
- Preferences ("multidimensional mining")
- Sentiment Analysis
- ...

Conclusion

- FCA is a well-founded mathematical theory equipped with efficient algorithmic tools.
- FCA is a polymorphic process and addresses problems ranging from knowledge discovery to knowledge representation and reasoning, and pattern recognition as well.
- FCA has two important variations for dealing with complex data: i.e. RCA and pattern structures (numbers, intervals, sequences...).
- There is still room for many improvements, especially in dealing with trees and graphs, in taking into account domain knowledge, similarity, and in combining FCA with numerical processes.

Tools for building and visualizing concept lattices

- The Conexp program:
  http://sourceforge.net/projects/conexp
- The Galicia Platform:
  http://www.iro.umontreal.ca/~galicia/
- The Toscana platform:
  http://tockit.sourceforge.net/toscanaj/index.html
- The Formal Concept Analysis Homepage:
  http://www.upriss.org.uk/fca/fca.html

Elements of bibliography on concept lattices

- Marc Barbut and Bernard Monjardet, Ordre et classification, Hachette, 1970.
- Introduction to Formal Concept Analysis. Radim Belohlavek, Palacky University, Olomouc, http://phoenix.inf.upol.cz/esf/ucebni/formal.pdf
- Claudio Carpineto and Giovanni Romano, Concept Data Analysis: Theory and Applications, John Wiley & Sons, 2004.
- Finite Ordered Sets: Concepts, Results and Uses. Nathalie Caspard, Bruno Leclerc and Bernard Monjardet, Cambridge University Press, 2012.
- Bernhard Ganter and Rudolf Wille, Formal Concept Analysis, Springer, 1999.
- Applied Lattice Theory: Formal Concept Analysis. Bernhard Ganter and Rudolf Wille, www.math.tu-dresden.de/~ganter/psfiles/concept.ps
- Formal Concept Analysis, Foundations and Applications. Bernhard Ganter, Gerd Stumme and Rudolf Wille editors, Lecture Notes in Computer Science 3626, Springer, 2005.