Algorithmic Knowledge Discovery with Lattices of Closed Descriptions

Sergei O. Kuznetsov

Department of Data Analysis and Artificial Intelligence
National Research University Higher School of Economics, Moscow, Russia

Logic and Applications (LAP 2016),
Dubrovnik, September 22, 2016

Outline

- Concept lattices: concepts and implications
- Computing concept lattices and implication bases
- Learning association rules
- Pattern structures: Relational Knowledge Discovery

Implicative Dependencies and Closures

- Concept (Galois) lattices: concept hierarchies and implicational dependencies
- functional dependencies
- Horn theories
- association rules
- version spaces
- emerging patterns

Basics of FCA

Formal Concept Analysis. 1. Data or "Context"

[Wille 1982], [Ganter, Wille 1996]

	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Objects:

1 – equilateral triangle,

2 – rectangle triangle,

3 – rectangle,

4 – square

Attributes:

a – has 3 vertices,

b – has 4 vertices,

c – has a direct angle,

d – equilateral

Formal Concept Analysis. 2

[Wille 1982], [Ganter, Wille 1996]

For a given context $\mathbb{K} := (G, M, I)$ consider two mappings $\varphi\colon 2^G \to 2^M$ and $\psi\colon 2^M \to 2^G$:

$$\varphi(A) \overset{\text{def}}{=} \{m \in M \mid gIm \ \text{ for all } g \in A\}, \psi(B) \overset{\text{def}}{=} \{g \in G \mid gIm \ \text{ for all } m \in B\}.$$

The following properties hold for all $A_1, A_2 \subseteq G$, $B_1, B_2 \subseteq M$

1. $A_1 \subseteq A_2 \Rightarrow \varphi(A_2) \subseteq \varphi(A_1)$
2. $B_1 \subseteq B_2 \Rightarrow \psi(B_2) \subseteq \psi(B_1)$
3. $A_1 \subseteq \psi\varphi(A_1)$ и $B_1 \subseteq \varphi\psi(B_1)$

Mappings φ and ψ define **Galois connection** between $(2^G, \subseteq)$ and $(2^M, \subseteq)$.

Abstract Galois Connections

For two ordered sets $(P, \leq_p)$ and $(Q, \leq_q)$ a pair of mappings

$\varphi\colon P \to Q, \quad \psi\colon Q \to P$ makes a **Galois connection** if

$x \leq_p \psi(y) \Leftrightarrow y \leq_q \varphi(x)$

Formal Concept Analysis. 3.

[Wille 1982], [Ganter, Wille 1996]

Let two sets G and M, called set of **objects** and set of **attributes**, be given. Let $I \subseteq G \times M$. If $(g, m) \in I$, one says that object g has attribute m. Triple $\mathbb{K} := (G, M, I)$ is called a **(formal)context**.

$$A' \stackrel{\text{def}}{=} \{m \in M \mid gIm \text{ for all } g \in A\}, \quad B' \stackrel{\text{def}}{=} \{g \in G \mid gIm \text{ for all } m \in B\}.$$

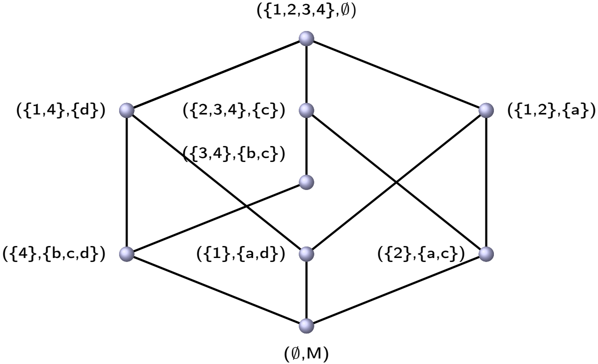
(Formal) concept is a pair (A, B):

$$A \subseteq G, \; B \subseteq M, \; A' = B, \; \text{ and } \; B' = A.$$

A is called a **(formal) extent**, and B is called a **(formal) intent** of a concept (A, B).

Concepts are partially ordered by relation
$(A_1, B_1) \geq (A_2, B_2) \iff A_1 \supseteq A_2 \quad (B_2 \supseteq B_1)$.

Example. Diagram of the ordered set of concepts



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

a – has 3 vertices,

b – has 4 vertices,

c – has a direct angle,

d – equilateral

Properties of operation $(\cdot)'$

Let (G, M, I) be a formal context, $A, A_1, A_2 \subseteq G$ be subsets of objects, $B \subseteq M$ be subsets of attributes, then

1. If $A_1 \subseteq A_2$, then $A_2' \subseteq A_1'$;
2. If $A_1 \subseteq A_2$, then $A_1^{''} \subseteq A_2^{''}$
3. $A \subseteq A''$
4. $A''' = A'$ (hence, $A'''' = A''$);
5. $(A_1 \cup A_2)' = A_1' \cap A_2'$;
6. $A \subseteq B' \Leftrightarrow B \subseteq A' \Leftrightarrow A \times B \subseteq I$.

Similar properties hold for subsets of attributes.

Closure operator on a set

A **closure operator** on set G is a mapping $\varphi\colon \mathcal{P}(G) \to \mathcal{P}(G)$ with the following properties:

1. $\varphi\varphi X = \varphi X$ (**idempotency**)
2. $X \subseteq \varphi X$ (**extensity**)
3. $X \subseteq Y \Rightarrow \varphi X \subseteq \varphi Y$ (**monotonicity**)

For a closure operator φ the set φX is called **closure** of X.
A subset $X \subseteq G$ is called **closed** if $\varphi X = X$.
Example. Let (G, M, I) be a context, then operators $(\cdot)''\colon 2^G \to 2^G$, $(\cdot)''\colon 2^M \to 2^M$ are closure operators.

Basic Theorem of Formal Concept Analysis

[Wille 1982], [Ganter, Wille 1996]

Concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice. For arbitrary sets of formal concepts

$$\{(A_j, B_j) \mid j \in J\} \subseteq \underline{\mathfrak{B}}(G, M, I)$$

infimums and supremums are given in the following way:

$$\bigwedge_{j \in J} (A_j, B_j) = (\bigcap_{j \in J} A_j, (\bigcup_{j \in J} B_j)''),$$
$$\bigvee_{j \in J} (A_j, B_j) = ((\bigcup_{j \in J} A_j)'', \bigcap_{j \in J} B_j).$$

A complete lattice V is isomorphic to a lattice $\underline{\mathfrak{B}}(G, M, I)$ iff there are mappings $\gamma\colon G \to V$ and $\mu\colon M \to V$ such that $\gamma(G)$ is supremum-dense in V, $\mu(M)$ is infimum-dense in V, and $gIm \Leftrightarrow \gamma g \leq \mu m$ for all $g \in G$ and all $m \in M$.
In particular, $V \cong \underline{\mathfrak{B}}(V, V, \leq)$.

FCA in Data Analysis. An Example

Example: treating acute limphoblastic leukemia (ALL)

Data from randomized clinical studies for two protocols: MB-ALL-2002 (more than 1500 patients) and MB-ALL-2008 (more than 2000 patients) conducted in Russia and Germany under surveillance of the Federal Center for Children's Immunology, Hematology and Oncology, Moscow.

Example: treating acute limphoblastic leukemia (ALL)

Data on treatment protocol for acute limphoblastic leukemia ALL-MB-2008.

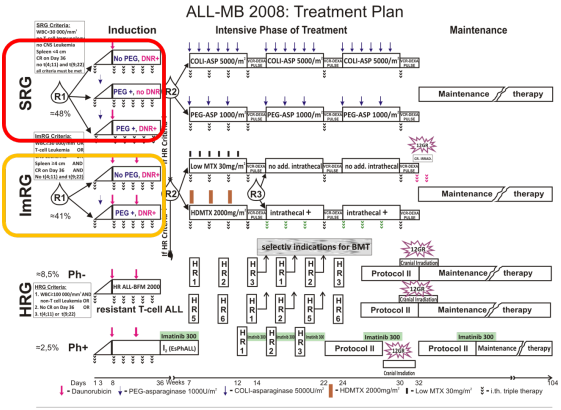


Figure: Plan of protocol ALL-MB-2008.

Physiological and diagnostics features as attributes

- Sex: categorical
- Age: integer
- Initial leukosis: “real”
- Immunophenotype: categorical
- Palpable liver size: “real”
- Palpable spleen size: “real”
- State of central neural system: categorical
- State of mediastinum: categorical

Example: treating ALL, standard risk group (SRG)

Totally 1121 patients

- strategy 100: 387 patients
- strategy 200: 366 patients
- strategy 300: 368 patients

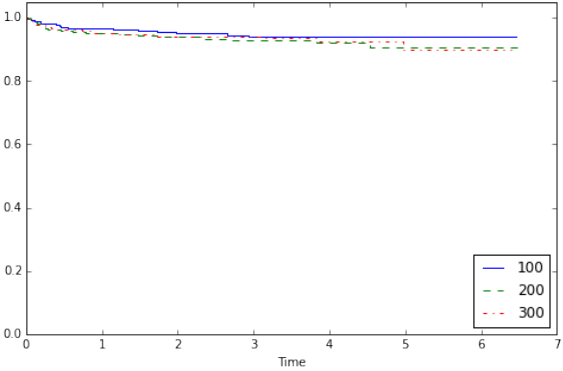


Figure: General survival

Looking for subgroups of patients

Standard approaches:

- distance-based clustering. Drawbacks: metrics-based artifacts
- SVMs, neural networks. Drawbacks: poor interpretability of results
- decision trees, random forests. Drawbacks: missing solutions

Hence, due to the high cost of an error, we need to perform global search for statistically significant groups defined by all possible attribute combinations, the exhaustive search.

This search can be however reduced (without losing any particular group) by taking only one representative attribute combination for every group. A natural candidate for this representative: inclusion-maximal description. It is unique: a closed description!

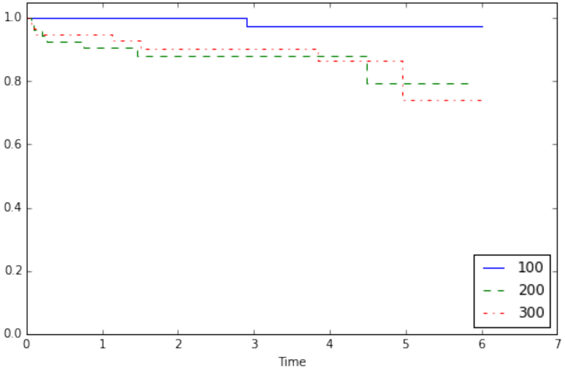
Example: general survival analysis for SRG

Strategy 100 is better than 200 and 300 if

1. $4 \le age \le 13$, $3 \le liver$
   - number of patients: 100 - 63, 200 - 52, 300 - 60
   - long-term survival: 100 - 99.9%, 200 - 80.9%, 300 - 73.5%
   - p-value: 100 vs 200 - 2.17%, 100 vs 300 - 0.52%
   - cardinality: 100 vs 200 - 91%, 100 vs 300 - 98%
2. $4 \le age$, $3 \le liver$, $cns = 1$
   - number of patients: 100 - 67, 200 - 61, 300 - 59
   - long-term survival: 100 - 97.2%, 200 - 79.1%, 300 - 73.4%
   - p-value: 100 vs 200 - 1.54%, 100 vs 300 - 1.63%
   - cardinality: 100 vs 200 - 86%, 100 vs 300 - 94%
3. $4 \le age$, $3 \le liver \le 7$
   - number of patients: 100 - 66, 200 - 61, 300 - 62
   - long-term survival: 100 - 97.2%, 200 - 79.1%, 300 - 73.9%
   - p-value: 100 vs 200 - 1.54%, 100 vs 300 - 1.96%
   - cardinality: 100 vs 200 - 86%, 100 vs 300 - 94%
4. $4 \le age$, $3 \le liver$, $spleen \le 4$
   - number of patients: 100 - 66, 200 - 61, 300 - 62
   - long-term survival: 100 - 97.2%, 200 - 79.1%, 300 - 73.9%
   - p-value: 100 vs 200 - 1.54%, 100 vs 300 - 1.96%
   - cardinality: 100 vs 200 - 86%, 100 vs 300 - 94%

Example: general survival analysis in SRG

Strategy 100 is better than 200 and 300 when $4 \leq$ *age*, $3 \leq$ *spleen* and one of the following conditions hold:
1) $age \leq 13$, 2) $cns = 1$, 3) $liver \leq 7$, 4) $spleen \leq 4$



FCA in Machine Learning

JSM-method of hypothesis generation

FCA translation of [Finn 1991]

A **target attribute** $w \notin M$,

- **positive examples**: Set $G_+ \subseteq G$ of objects known to have w,
- **negative examples**: Set $G_- \subseteq G$ of objects known not to have w,
- **undetermined examples**: Set $G_\tau \subseteq G$ of objects for which it is unknown whether they have the target attribute or do not have it.

Three subcontexts of $\mathbb{K} = (G, M, I)$: $\mathbb{K}_\varepsilon := (G_\varepsilon, M, I_\varepsilon)$, $\varepsilon \in \{-, +, \tau\}$ with respective derivation operators $(\cdot)^+$, $(\cdot)^-$, and $(\cdot)^\tau$.

A **positive hypothesis** $H \subseteq M$ is an intent of $\mathbb{K}_+$ not contained in the intent g^- of any negative example $g \in G_-$: $\forall g \in G_- \quad H \not\subseteq g^-$. Equivalently,

$$H^{++} = H, \quad H' \subseteq G_+ \cup G_\tau.$$

Example of a learning context

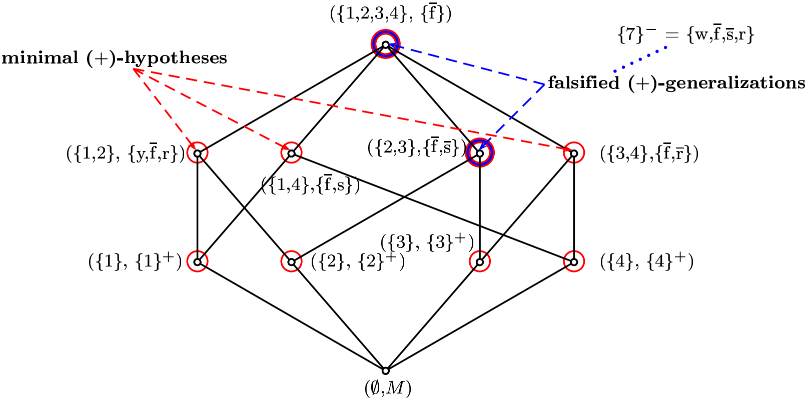
	G \ M	color	firm	smooth	form	**fruit**
1	apple	yellow	no	yes	round	+
2	grapefruit	yellow	no	no	round	+
3	kiwi	green	no	no	oval	+
4	plum	blue	no	yes	oval	+
5	toy cube	green	yes	yes	cubic	−
6	egg	white	yes	yes	oval	−
7	tennis ball	white	no	no	round	−
8	mango	green	no	yes	oval	τ

Natural scaling (binarization) of the context

	G \ M	w	y	g	b	f	$\bar{\text{f}}$	s	$\bar{\text{s}}$	r	$\bar{\text{r}}$	**fruit**
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	−
6	egg	×				×		×			×	−
7	tennis ball	×					×		×	×		−
8	mango			×			×	×			×	τ

Abbreviations:
“g” for green, “y” for yellow, “w” for white, “f” for firm, “$\bar{\text{f}}$” for nonfirm,
“s” for smooth, “$\bar{\text{s}}$” for nonsmooth, “r” for round,
“$\bar{\text{r}}$” for nonround.

Positive Concept Lattice



	G \ M	w	y	g	b	f	$\overline{f}$	s	$\overline{s}$	r	$\overline{r}$	fruit
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	−
6	egg	×				×		×			×	−
7	tennis ball	×					×		×	×		−
8	mango			×			×	×			×	τ

Implications

Implication $A \to B$, where $A, B \subseteq M$ holds in context (G, M, I) if $A' \subseteq B'$, i.e., each object having all attributes from A also has all attributes from B.

Implications and concept lattice: If $A \to B$, then the meet of all attribute concepts for attributes from A in the lattice diagram lies below the meet of all attribute concepts of attributes in B.

Implications satisfy **Armstrong rules**:

$$\frac{}{X \to X}\ , \quad \frac{X \to Y}{X \cup Z \to Y}\ , \quad \frac{X \to Y, Y \cup Z \to W}{X \cup Z \to W},$$

An **implication cover** is a subset of implications from which all other implications can be derived by means of Armstrong rules.
An **implication base** is a minimal (by inclusion) implication cover.

Hypotheses vs. implications

A positive hypothesis h corresponds to an implication $h \to \{w\}$ in the context $K_+ = (G_+, M \cup \{w\}, I_+ \cup G_+ \times \{w\})$.
A negative hypothesis h corresponds to an implication $h \to \{\bar{w}\}$ in the context $K_- = (G_-, M \cup \{\bar{w}\}, I_- \cup G_- \times \{\bar{w}\})$.
Hypotheses are special implications: their premises are closed (in K_+ or in K_-).

	G \ M	w	y	g	b	f	$\bar{\text{f}}$	s	$\bar{\text{s}}$	r	$\bar{\text{r}}$	**fruit**	**nonfruit**
1	apple		×				×	×		×		×	
2	grapefruit		×				×		×	×		×	
3	kiwi			×			×		×		×	×	
4	plum				×		×	×			×	×	
5	toy cube			×		×		×			×		×
6	egg	×				×		×			×		×
7	tennis ball	×					×		×	×			×

Functional dependencies

For a relational datatable (G, M, W, I), where $I \subseteq G \times M \times W$, and $X, Y \subseteq M$

$X \Rightarrow Y$ is a **functional dependency** if the following holds for every pair of objects $g, h \in G$:

$$(\forall m \in X \;\; m(g) = m(h)) \to (\forall n \in Y \;\; n(g) = n(h)).$$

Functional dependencies $\to$ Implications

For a relational database $K = (G, M, W, I)$ one can construct a context $K_N := (\mathcal{P}_2(G), M, I_N)$, where $\mathcal{P}_2(G)$ is a set of pairs of different objects from G, the relation I_N is defined as

$$\{g, h\} I_N m :\Leftrightarrow m(g) = m(h).$$

so that a functional dependency $X \Rightarrow Y$ holds in K iff implication $X \to Y$ holds in K_N.

Functional dependencies → Implications

G \ M	a	b	c	d
1	0	0	1	1
2	0	2	0	0
3	3	0	3	0

A context with the "same" implications:

G \ M	a	b	c	d
12	×			
13		×		
23				×

Implications → Functional dependencies

G \ M	a	b	c	d
1	×	×		
2	×		×	×
3		×		×
4	×		×	

A relational database with the "same" functional dependencies:

G \ M	a	b	c	d
1	1	1	2	2
2	1	3	1	1
3	4	1	4	1
4	1	5	1	5
5	1	1	1	1

Minimum implication base

[Duquenne, Guigues 1986]

Duquenne-Guigues base is an implication base where each implication premise is a pseudo-intent:

$$\{P \to (P'' \setminus P) \mid P \text{ is a pseudo-intent}\}.$$

A subset of attributes $P \subseteq M$ is called a **pseudo-intent** if $P \neq P''$ and for any pseudo-intent Q such that $Q \subset P$ one has $Q'' \subset P$.

Duquenne-Guigues base is a minimum (cardinality minimal) implication base.
Computing Duquenne-Guigues base $\sim$ learning a propositional Horn theory from models

Two closures

Given $K = (G, M, I)$, $B \subseteq M$, an implication base IB

- "double prime" closure B''
- implication closure: iterate
  $B := B \cup \bigcup\{D \mid E \to D, E \subseteq B, E \to D \text{ from } IB\}$

Two closures are equivalent

Computing concepts

Main intractability results for concept lattices

- Consider context $K = (G, G \neq)$ for an arbitrary finite set G. Then $\underline{\mathfrak{B}}(K)$ is isomorphic to the Boolean lattice 2^G with $2^{|G|}$ concepts.
- Given a context $K = (G, M, I)$, computing the size of $\underline{\mathfrak{B}}(K)$ is #P-complete [Kuznetsov 1989, 2001].

#P and #P-completeness [L.Valiant, 1979]

Definition: #P is the class of counting problems associated with the decision problems in NP. More formally, a problem is in #P if there is a non-deterministic, polynomial time Turing machine that, for each instance I of the problem, has a number of accepting computations that is exactly equal to the number of distinct solutions for instance I.

A problem is #P-**complete** if it is in #P and it is #P-**hard**, i.e., any problem in #P can be reduced by Turing to it. In particular, a problem in #P is #P-complete if a #P-complete problem can be reduced to it. Obviously, $\#\mathrm{P} = \mathrm{P} \Longrightarrow \mathrm{NP} = \mathrm{P}$.

Examples of #P-complete problems:

- Given a matrix, output its permanent
- Given a bipartite graph, output the number of its perfect matchings
- Given a CNF, output the number of its satisfying assignments
- Given a graph, output the number of its vertex covers

#P-completeness of counting concepts.

First, the problem of recognizing whether a pair (A, B) is a concept of context K is solvable in polynomial time, therefore, the problem

INPUT Context $K = (G, M, I)$.
OUTPUT The number of all concepts of the context K.

is in #P. Its completeness is shown by reducing to it the following problem

INPUT A set of binary variables $X = \{x_1, \ldots, x_n\}$, and C, a monotone CNF with two variables in each conjunction:
$C = \bigwedge_{i=1}^{s}(x_{i_1} \vee x_{i_2})$, with $x_{i_1}, x_{i_2} \in X$ for all $i = \overline{1, s}$.
OUTPUT Number of binary n-vectors (binary assignments of variables from X) that satisfy CNF C.

Polynomial-delay algorithms for computing concepts

Definition

*An algorithm for listing a family of combinatorial structures has **delay** d [Johnson et al. 1988] if it executes at most d many computation steps before either outputting each next structure or halting. **Polynomial delay**: d is bounded from above by a polynomial from input size.*

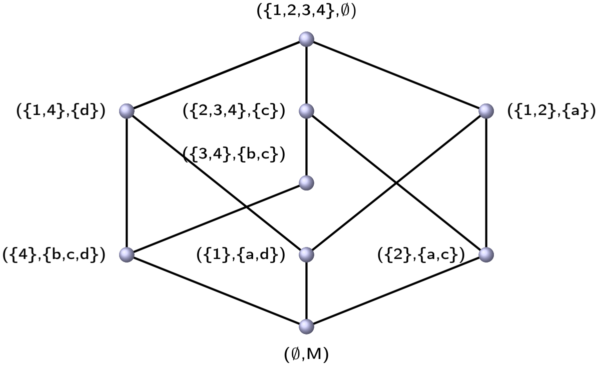
.

Generation of concepts and hypotheses can be done with polynomial delay $O(|G|^2 \cdot |M|)$ or $O(|G|^3 \cdot |M|)$. Many batch algorithms have polynomial delay, among them

- B. Ganter 1984 (NextClosure)
- J.-P. Bordat 1986
- S. Kuznetsov 1993 (CbO),
- D. van der Merwe et al. 2004 (AddIntent)
- J. Outrata and V. Vychodil, 2010 (FCbO)
- ...

All these algorithms have $O(|G|^2 \cdot |M| \cdot |L|)$ total runtime complexity. The algorithm of L.Nourine and O.Raynaud (2000) has $O(|G| \cdot |M| \cdot |L|)$ complexity, but is very slow in practice.

Example. A context and a concept lattice



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

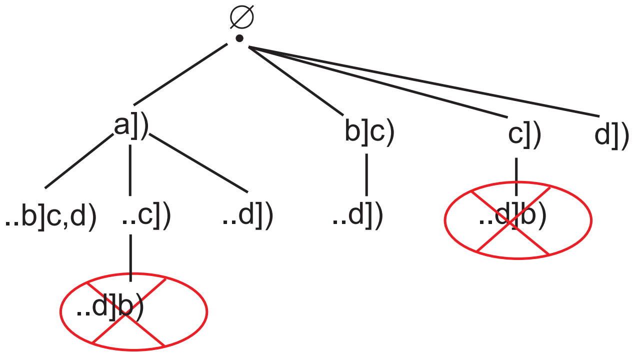
a – has 3 vertices,

b – has 4 vertices,

c – has a direct angle,

d – equilateral

CbO in the strategy "top-down" (attribute-wise)



Fast computation of concepts under monotone constraints

- Computing only concepts with big extents (“large support”).
- Computing hypotheses (intents not contained in special intents called “negative examples”) quality indices.

can be performed with polynomial delay by changing the branching condition of an algorithm for computing concepts.

Computing implication bases

Duquenne-Guigues base can be exponential in the context size

$G \setminus M$	m_0	$m_1 \dots m_n$	$m_{n+1} \dots m_{2n}$
g_1 $\vdots$ g_n		$\neq$	$\neq$
g_{n+1} $\vdots$ $\vdots$ $\vdots$ g_{3n}	$\times$ $\vdots$ $\vdots$ $\vdots$ $\times$	$\neq$	

The set $\{m_1, \dots, m_n\}$ is a pseudo-intent. Replacing m_i with m_{n+i} independently for each i, one obtains all 2^n pseudo-intents of the context.

An example: $K_{\mathrm{exp},3}$

$G \setminus M$	m_0	m_1	m_2	m_3	m_4	m_5	m_6
g_1			×	×		×	×
g_2		×		×	×		×
g_3		×	×		×	×	
g_4	×		×	×	×	×	×
g_5	×	×		×	×	×	×
g_6	×	×	×		×	×	×
g_7	×	×	×	×		×	×
g_8	×	×	×	×	×		×
g_9	×	×	×	×	×	×	

Here, we have $2^3 = 8$ pseudo-intents: $\{m_1, m_2, m_3\}$, $\{m_1, m_2, m_6\}$, $\{m_1, m_5, m_3\}$, $\{m_1, m_5, m_6\}$, $\{m_4, m_2, m_3\}$, $\{m_4, m_2, m_6\}$, $\{m_4, m_5, m_3\}$, $\{m_4, m_5, m_6\}$.

Complexity of Computing Minimal Implication Bases

Recall that pseudo-intents are premises of the cardinality-minimal implication base (Duquenne-Guigue base).

- The number of pseudo-intents can be exponential, the problem of counting pseudo-intents is #P-hard [Kuznetsov 2004].
- Pseudo-intents cannot be generated in lexicographic order with polynomial delay unless P=NP, generating pseudo-intents is TRANSENUM-hard [F.Distel, B.Sertkaya 2010]
- Recognizing whether a subset of attributes is a pseudo-intent is coNP-complete [Kuznetsov, Obiedkov 2006], [Babin, Kuznetsov 2010]

Efficient lazy classification with implications

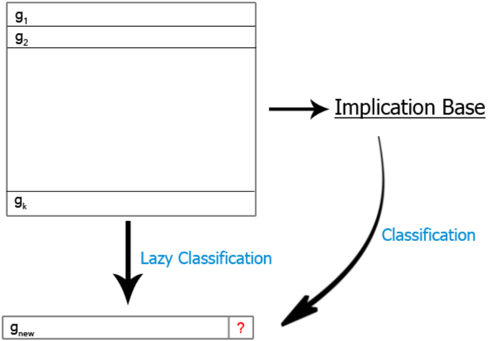
Lazy evaluation vs. computing bases

If you need implications for classification, make classification directly from data

If you need "understanding" data, compute a subbase: small-premise implications or those implicitly used for classification (anecdotal base)

Short-Premise Base

a	$\to$	$\dots$
ab	$\to$	$\dots$
abc	$\to$	$\dots$
$abcd$	$\to$	$\dots$



Good News: Classification with $(\cdot)''$-closure

For an arbitrary subset of attributes $A \subseteq M$ the maximal set which can be deduced with implications of the context is A''

What is A''? Take all objects that contain A and intersect them.

This takes $O(|G| \cdot |M|)$ time.

Lazy classification: An example

$G \setminus M$	target	m_1	m_2	m_3	m_4	m_5	m_6
g_1			×	×		×	×
g_2		×		×	×		×
g_3		×	×		×	×	
g_4	×		×	×	×	×	×
g_5	×	×		×	×	×	×
g_6	×	×	×		×	×	×
g_7	×	×	×	×		×	×
g_8	×	×	×	×	×		×
g_9	×	×	×	×	×	×	

$2^{|M|/2} = 8$ implications in the minimal base, with the premises

$$\{m_1, m_2, m_3\},\ \{m_1, m_2, m_6\},\ \{m_1, m_5, m_3\},\ \{m_1, m_5, m_6\},$$

$$\{m_4, m_2, m_3\},\ \{m_4, m_2, m_6\},\ \{m_4, m_5, m_3\},\ \{m_4, m_5, m_6\}.$$

To classify a new object g w.r.t. target t, compute $(g' \cap g_i')''$, which takes $O(|G|^2 \cdot |M|)$ time.

If $g' = \{m_1, m_2, m_5\}$, then for all g_i one has $t \not\subseteq (g_i' \cap \{m_1, m_2, m_5\})''$, hence g is classified negatively.

Lazy classification: An example

$G \setminus M$	target	m_1	m_2	m_3	m_4	m_5	m_6
g_1			×	×		×	×
g_2		×		×	×		×
g_3		×	×		×	×	
g_4	×		×	×	×	×	×
g_5	×	×		×	×	×	×
g_6	×	×	×		×	×	×
g_7	×	×	×	×		×	×
g_8	×	×	×	×	×		×
g_9	×	×	×	×	×	×	

$2^{|M|/2} = 8$ implications in the minimal base, with the premises

$$\{m_1, m_2, m_3\}, \{m_1, m_2, m_6\}, \{m_1, m_5, m_3\}, \{m_1, m_5, m_6\},$$

$$\{m_4, m_2, m_3\}, \{m_4, m_2, m_6\}, \{m_4, m_5, m_3\}, \{m_4, m_5, m_6\}.$$

To classify a new object h w.r.t. target t, compute $(h' \cap g_i')''$, which takes $O(|G|^2 \cdot |M|)$ time. If $h' = \{m_1, m_2, m_3, \}$, then for g_7 one has $(g_7 \cap \{m_1, m_2, m_3\})'' = \{m_1, m_2, m_3, t\}$, hence h is classified positively.

Data Mining. Learning Association Rules

Outline

- Learning association rules
- Learning JSM-hypotheses
- Relational learning with pattern structures
- Applications

Lattices in Data Mining. Association rules

In mid 1990s in papers of R. Agrawal et al. on association rules "partial implications" from FCA were rediscovered.

A partial implication (association rule) of context (G, M, I) is an expression $A \rightarrow_{c,s} B$, where

- $c, s \in [0, 1]$;
- $c = \frac{|(A \cup B)'|}{|A'|}$, called **confidence**, $\mathrm{conf}(A \rightarrow B)$;
- $s = \frac{|(A \cup B)'|}{|G|}$, called **support**, $\mathrm{supp}(A \rightarrow B)$.

Covers of association rules

What is a minimal representation of the set of association rules, from which one can obtain all association rules of a context using "admissible transformations"?

Consider an association rule $A \to_{c,s} B$. Under fixed confidence $c = \frac{|(A \cup B)'|}{|A'|}$ and support $s = \frac{|(A \cup B)'|}{|G|}$ we try to reduce its premise and increase its conclusion.

1. **Decreasing premise.** For fixed c and s one can decrease premise from A to a certain subset $D \subseteq A$ such that $(D \cup B)' = (A \cup B)' = A' \cap B' = D' \cap B'$, that is $D' = A' = A'''$. Thus, minimal D is by definition a minimal generator of A'', i.e. $D \in \text{mingen}(A'')$.

Recall that a subset of attributes $D \subseteq M$ is a **generator** of a closed subset of attributes $B \subseteq M$, $B'' = B$ if $D \subseteq B$, $D'' = B = B''$. A subset $D \subseteq M$ is a **minimal generator** if for any $E \subset D$ one has $E'' \neq D'' = B''$.

Covers of association rules

2. **Increasing conclusion.** Conclusion B can be increased by a set Δ such that $(A \cup B)' = (A \cup B \cup \Delta)' = (A \cup B)' \cap \Delta'$, which is possible only when $(A \cup B)' \subseteq \Delta'$, which is equivalent to $A \cup B \to \Delta$ and to $\Delta \subseteq (A \cup B)''$. Thus, conclusion of the association rule can be increased up to $(A \cup B)''$.
Thus, the rules from the set

$$CP(K) = \{D \to (A \cup B)'' \mid D \in \text{ mingen } (A'')\}$$

make a cover of the set of all association rules. We can obtain all other rules by admissible transformations – increasing premises and decreasing conclusions (these operations do not decrease confidence and support) – of rules from $CP(K)$. In terms of these admissible transformations, $CP(K)$ makes a cover of association rules.

Base of association rules

Consider an association rule of the form $D \to (A \cup B)''$, where $D \in \text{mingen}(A'')$. In the concept lattice diagram this rule corresponds to a path from the concept (A', A'') to the concept $((A \cup B)', (A \cup B)'')$. If $(A', A'') \not\succ ((A \cup B)', (A \cup B)'')$, i.e., if the vertex of the diagram corresponding to the concept (A', A'') is not an upper neighbor of the vertex corresponding to $((A \cup B)', (A \cup B)'')$, then there is a concept (E', E'') such that $(A', A'') \succ (E', E'') > ((A \cup B)', (A \cup B)'')$.
Consider $D \to E''$, where $D \in \text{mingen}(A'')$ and $F \to (A \cup B)''$, where $F \in \text{mingen}(E'')$. The confidence of the first rule is $c_1 = \frac{|E'|}{|A'|}$, and the confidence of the second rule is $c_2 = \frac{|(A \cup B)'|}{|E'|}$. The confidence of the rule $D \to (A \cup B)''$, where $D \in \text{mingen}(A'')$ is

$$c = \frac{|(A \cup B)'|}{|A'|} = \frac{|E'|}{|A'|} \cdot \frac{|(A \cup B)'|}{|E'|} = c_1 \cdot c_2.$$

Base of association rules

Hence, the cover of the set of association rules can be made even smaller by restricting to the set of rules

$$\{F \to ('' \setminus F'') \mid F \subseteq M, F \in \text{ mingen } (F''), (F', F'') \succ (E', E'')\},$$

which correspond to the arcs of the diagram. Supports and confidence of other rules from the cover can be obtained by multiplying supports along the respective paths in the diagram.

To minimize the cover of association rules, making it a basis, one can retain only those rules from $CP(K)$ that correspond to edges from a spanning tree of the lattice diagram.

General task of finding association rules

Find all "frequent" (with support greater than a threshold) association rules with confidence greater than a threshold.

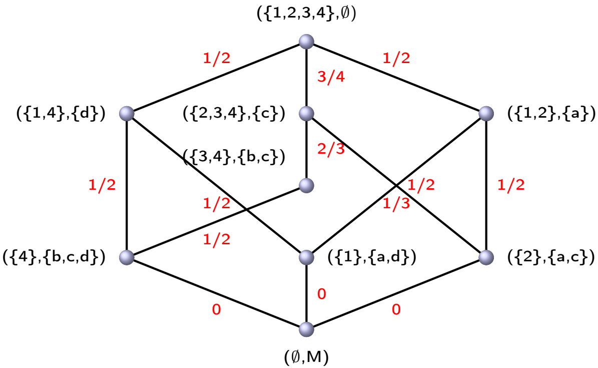
Solution stages

- Find all frequent "closed itemsets" (frequent intents)
- For each frequent intent B find all its maximal subintents $A_1, \ldots, A_n$
- Retain only those A_i for which $\text{conf}(A_i \to B) \geq \theta$, where θ is confidence threshold
- Find minimal generators of the remaining A_i, compose rules of the form $mingen(A_i) \to B$.

Luxenburger basis

- Spanning tree of the concept lattice diagram
- Duquenne-Guigues implication base

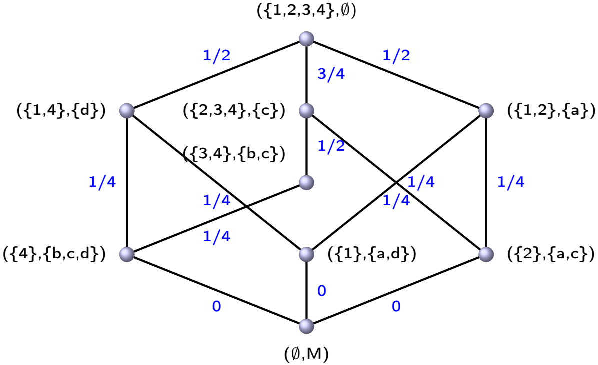
Example. Confidence of association rules



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Good rules with *supp* $\geq 1/2$ and *minconf* $\geq 3/4$
1. $\emptyset \to c$, sup($\emptyset \to c$) = conf($\emptyset \to c$) = 3/4;
2. $c \to b$, sup($c \to b$)= 1/2, conf($c \to b$) = 2/3.

Example. Support of association rules



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Good rules with *supp* $\geq 1/2$ and *minconf* $\geq 3/4$
1. $\emptyset \to c$, $\quad$ sup($\emptyset \to c$) = conf($\emptyset \to c$) = 3/4;
2. $c \to b$, $\quad$ sup($c \to b$)= 1/2, conf($c \to b$) = 2/3.

Pattern Structures. Relational Knowledge Discovery

Learning with labeled graphs: A motivation

- Structure-Activity Relationship problems for chemicals given by molecular graphs
- Learning semantics from graph-based (XML, syntactic tree) text representation

Starting point

To proceed with graphs like it was done for objects described by binary sets of attributes (i.e., for contexts), one should define for graphs an operation $\sqcap$ similar to that of set-theoretic $\cap$ (since then a closure operator $''$ can be defined).

The first natural attempt to do this, like introducing an operation "take the largest common subgraph of two graphs" fails, since there can be several subgraphs of this type.

Perhaps operation should be defined not for graphs, but for sets of graphs?

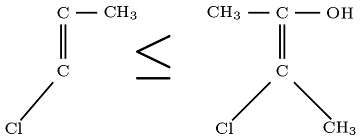
The attempt even fails if we take all largest (in the number of vertices) common subgraphs of two graphs.

Order on labeled graphs

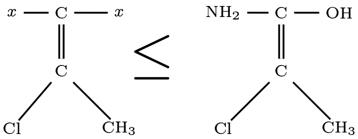
Let $(\mathcal{L}, \preceq)$ be an ordered set of vertex labels.
$\Gamma_1 := ((V_1, l_1), E_1)$ **dominates** $\Gamma_2 := ((V_2, l_2), E_2)$ or $\Gamma_2 \leq \Gamma_1$
if there exists a one-to-one mapping $\varphi \colon V_2 \to V_1$ such that

- respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $l_2(v) \preceq l_1(\varphi(v))$.

Example: $\mathcal{L} = \{x, NH_2, Cl, CH_3, C, OH\}$



vertex labels are unordered



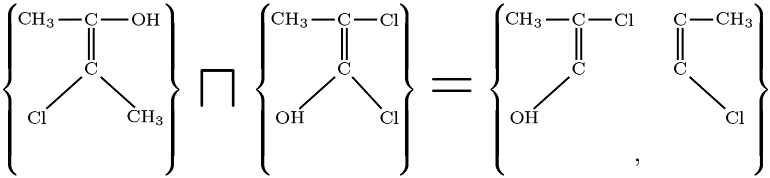
$x \preceq A$ for any vertex label $A \in \mathcal{L}$

Semilattice on graph sets

$\{X\} \sqcap \{Y\} := \{Z \mid Z \leq X, Y, \quad \forall Z_* \leq X, Y \quad Z_* \not< Z\}$

$=$ The set of all maximal common subgraphs of X and Y.

Example:



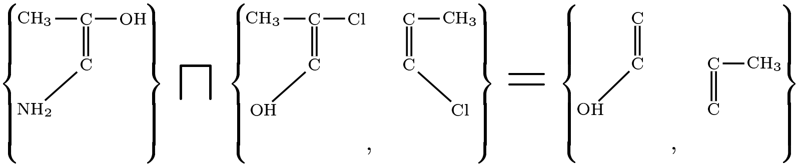
Meet of graph sets

For sets of graphs
$\mathcal{X} = \{X_1, \ldots, X_k\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$

$\mathcal{X} \sqcap \mathcal{Y} :=$ MAX $_{\leq}(\bigcup_{i,j}(\{X_i\} \sqcap \{Y_j\}))$

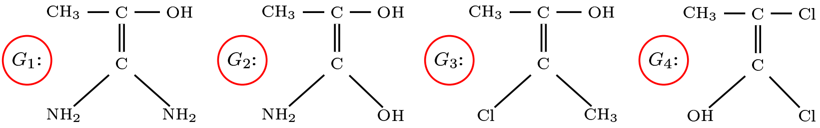
$\sqcap$ is idempotent, commutative, and associative.

Example:

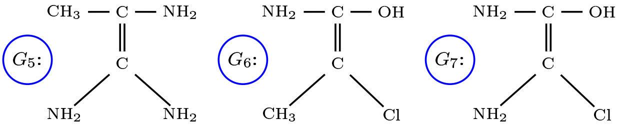


Examples

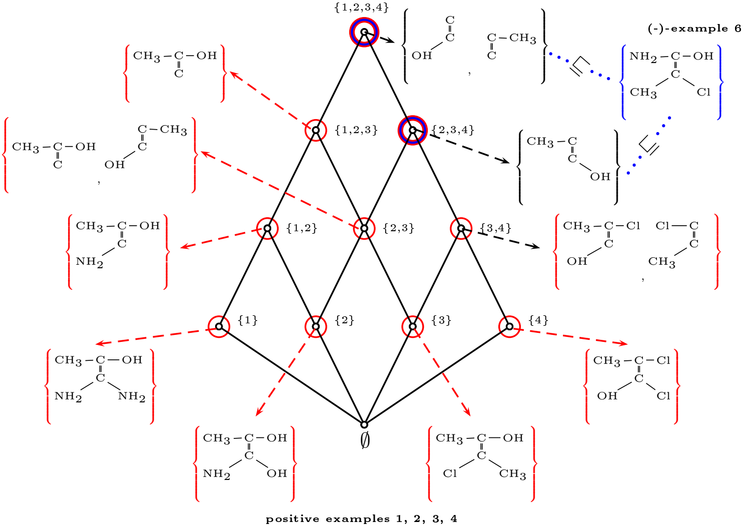
Positive examples:



Negative examples:



Positive lattice



Pattern Structures

[Ganter, Kuznetsov 2001]

$(G, \underline{D}, \delta)$ is a **pattern structure** if

- G is a set ("set of objects");
- $\underline{D} = (D, \sqcap)$ is a meet-semilattice;
- $\delta : G \to D$ is a mapping;
- the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$.

Possible origin of $\sqcap$ operation:

- A set of objects G, each with description from P;
- Partially ordered set $(P, \leq)$ of "descriptions" ($\leq$ is a "more general than" relation);
- The (distributive) lattice of order ideals of the ordered set $(P, \leq)$.

Pattern Structures

Let $(G, (D, \sqcap), \delta)$ be a **pattern structure**, then

the subsumption order is defined as $c \sqsubseteq d :\ = c \sqcap d = c$.

Derivation operators:

$$A^\diamond := \sqcap_{g \in A} \delta(g) \text{ for } A \subseteq G$$

$$c^\diamond := \{g \in G \mid c \sqsubseteq \delta(g)\} \text{ for } c \in C.$$

A pair (A, c) is a **pattern concept** of $(G, (C, \sqcap), \delta)$ if

$$A \subseteq G, c \in C, A^\diamond = c, c^\diamond = A$$

A is **extent** and c is **pattern intent**.
$A \subseteq G$ is **closed** if $A^{\diamond\diamond} = A$.
$d \in D$ is **closed** if $d^{\diamond\diamond} = d$.

Reinventing the closure: Data mining

Late 1990s: a wave of interest in graph mining, with application in chemistry, protein analysis, analysis of XML documents, etc. First Apriori-like algorithm gSpan was fairly efficient.

X. Yan and J. Han, gSpan: Graph-Based Substructure Pattern Mining, Proc. IEEE Int. Conf. on Data Mining, ICDM'02, 2002, pp.721–724, IEEE Computer Society

However, it was outperformed by CloseGraph from

X. Yan and J. Han, CloseGraph: mining closed frequent graph patterns, *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD'03*

where **closed graphs** are defined in terms of "counting inference":
Given a labeled graph dataset D and a graph $g \in D$
support(g) is a set (or number) of graphs in D, in which g is a subgraph.
A graph g is called **closed** if no supergraph f of g (i.e., a graph such that g is isomorphic to its subgraph) has the same support.

Closed graphs and closed sets of graphs

Remark: Unlike closed subsets of attributes, closed graphs, ordered by subgraph isomorphism, do not make a lattice (there can be multiple sups and infs).

However, closed graphs are related to closed sets of graphs (i.e., sets $\mathcal{G}$ such that $\mathcal{G}^{\diamond\diamond} = \mathcal{G}$) as follows:

Proposition. Let a labeled graph dataset D be given, then

1. For a closed graph g there is a closed set of graphs $\mathcal{G}$ such that $g \in \mathcal{G}$.
2. For a closed set of graphs $\mathcal{G}$ and an arbitrary $g \in \mathcal{G}$, graph g is closed.

Projections as Approximation Tool

Motivation: Complexity of computations in $(G, \underline{D}, \delta)$, e.g.,
testing SUBGRAPH ISOMORPHISM, i.e., relation $\leq$ for graphs, is NP-complete.

ψ is **projection** on an ordered set $(D, \sqsubseteq)$ if ψ is

monotone: if $x \sqsubseteq y$, then $\psi(x) \sqsubseteq \psi(y)$,

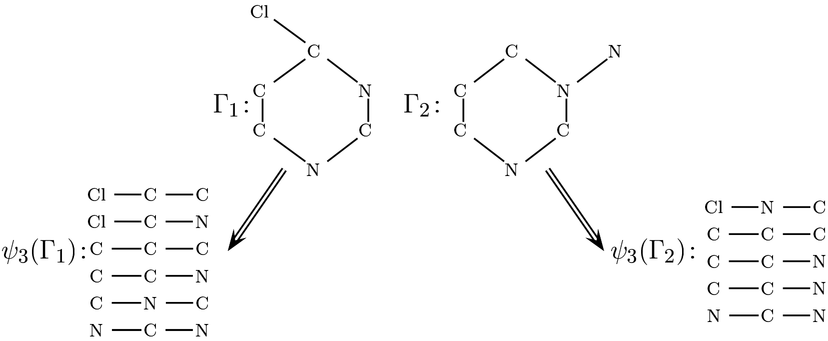
contractive:

a. $\psi(x) \sqsubseteq x$
b. $\forall x, \forall y, \exists z$: $y \sqsubseteq \psi(x) \Rightarrow y = \psi(z)$

idempotent: $\psi(\psi(x)) = \psi(x)$.

Projections as Approximation Tool

Example. Projection $\psi_3(\Gamma)$ takes Γ_1 and Γ_2 to the sets of their connected 3-vertex subgraphs.



Property of projections

Any projection of a complete semilattice $(D, \sqcap)$ *is* $\sqcap$*-preserving, i.e., for any* $X, Y \in D$

$$\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y).$$

Example. A projection $\psi_n(\Gamma)$ takes Γ to the set of its n-chains not dominated by other n-chains. Here $n = 3$, the label x is smaller than other labels, other labels are pairwise incomparable.

$\psi_3(\Gamma_1)$:

Cl — C — C
Cl — C — N
C — C — C
C — C — N
C — N — C
N — C — N

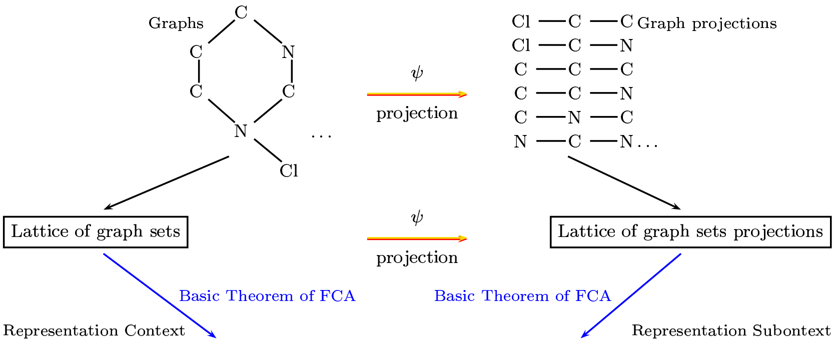
$\psi_3(\Gamma_2)$:

Cl — N — C
C — C — C
C — C — N
C — N — C
N — C — N

$\psi_3(\Gamma_1) \sqcap \psi_3(\Gamma_2)$:

Cl — x — C
C — C — C
C — C — N
C — N — C
N — C — N

Projections and Representation Context



Representation Context

G \ M	a	b	c	d	e	f	**goal**
1	×	×	×		×		+
2	×	×	×	×	×	×	+
3	×	×		×			+
4	×	×		×		×	+
5	×		×		×		−
6	×	×	×	×		×	−
7		×	×	×	×	×	−

ψ

projection

Representation Subontext

G \ M	a	b	c	d	e	f	**goal**
1	×	×	×		×		+
2	×	×	×	×	×	×	+
3	×	×		×			+
4	×	×		×		×	+
5	×		×		×		−
6	×	×	×	×		×	−
7		×	×	×	×	×	−

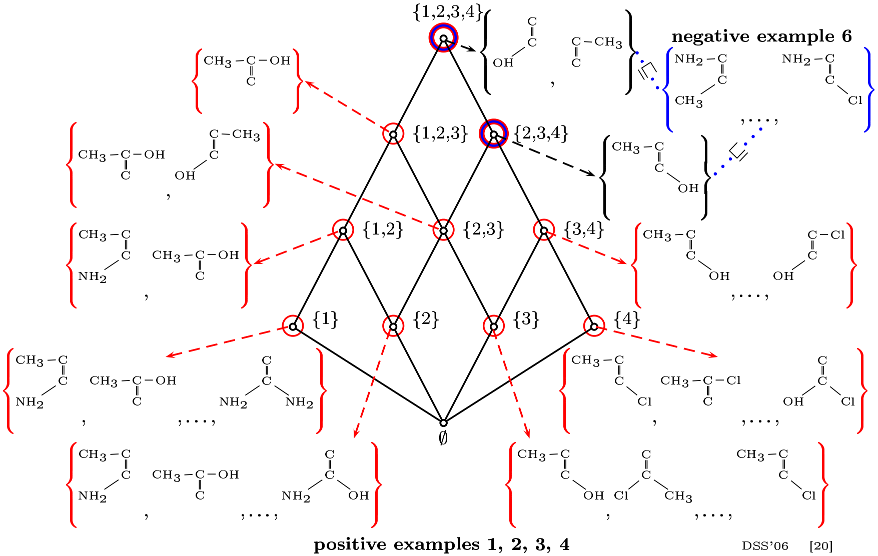
Projection types used in chemical applications

We used several types of projections of labeled graph sets that are natural in chemical applications:

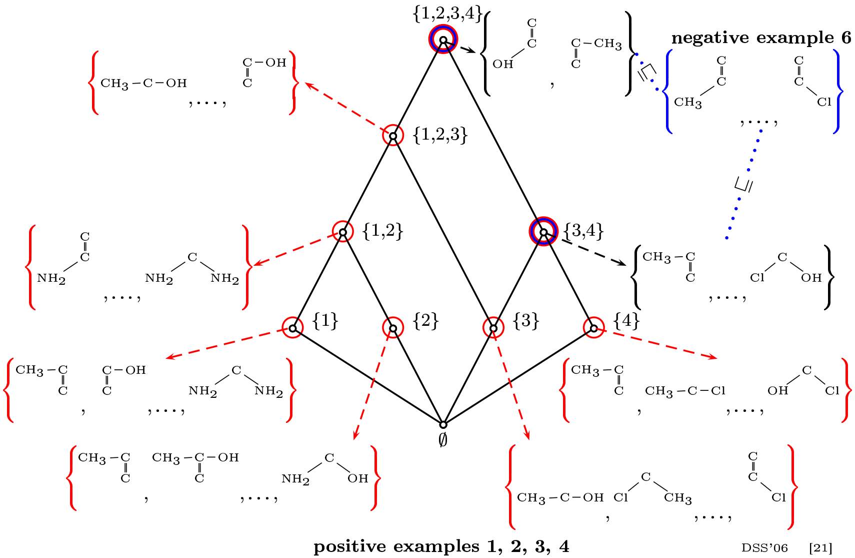
- *k-chain* projection: a set of graphs X is taken to the set of all chains with k vertices that are subgraphs of at least one graph of the set X;
- *k-vertex* projection: a set of graphs X is taken to the set of all subgraphs with k vertices that are subgraphs of at least one graph of the set X;
- *k-cycles* projection: a set of graphs X is taken to the set of all subgraphs consisting of k adjacent cycles of a minimal cyclic basis of at least one graph of the set X.

Mixed projections (with same algebraical properties of simple projections) are also possible.

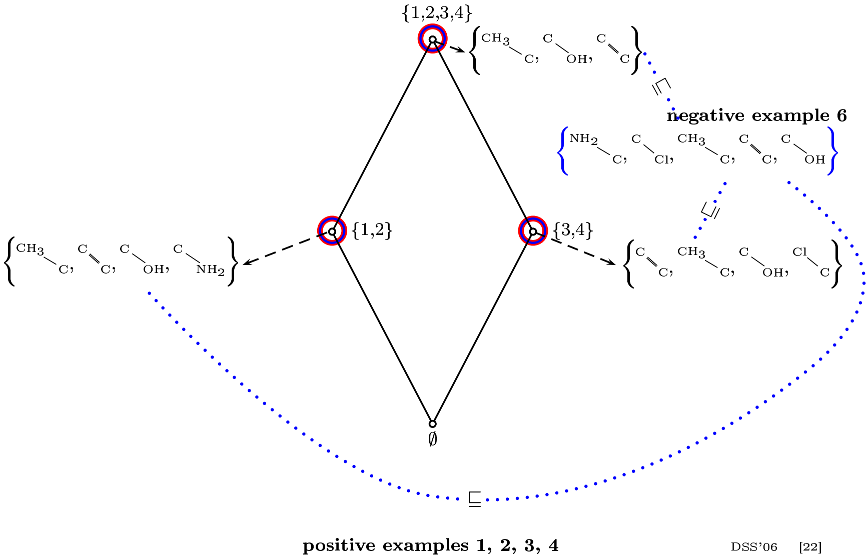
4-Projections



3-Projections



2-Projections



Pattern Structure with Description Logic

Description Logic $\mathcal{EL}^{\perp}$. Syntax

Description Logics: a well-founded logical means for efficient knowledge representation and reasoning [Baader, Calvanese, Horrocks, McGuiness,...].
Plenty of DLs: $\mathcal{EL}$, $\mathcal{EL}^{++}$ (OWL 2 EL), (OWL 2 DL), ...

Basic notions of $\mathcal{EL}^{\perp}$.
A **signature** is a pair (N_C, N_R):
N_C, a set of **concept names**,
N_R, a set of **role names**.
Concept descriptions: each concept name $A \in N_C$ is a concept description;
the **bottom concept** $\perp$ and the **top concept** $\top$ are concept descriptions;
if C and D are concept descriptions, then their **conjunction** $C \sqcap D$ is a concept description;
if $r \in N_R$ is a role name, and C is a concept description, then the **existential restriction** $\exists r.C$ is a concept description.

The set of all $\mathcal{EL}^{\perp}$-concept descriptions (w.r.t. signature (N_C, N_R)) is symbolized as $\mathcal{EL}^{\perp}(N_C, N_R)$.

Description Logic $\mathcal{EL}^{\perp}$. Semantics

An **interpretation** is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set (**domain**), and $(\cdot)^{\mathcal{I}}$ is an **extension function** that maps concept names $A \in N_C$ to subsets $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and role names $r \in N_R$ to binary relations $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The extension function is canonically extended to all concept descriptions as follows:

$$
\begin{aligned}
\bot^{\mathcal{I}} &:= \emptyset \\
\top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} &:= \{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d, e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}
\end{aligned}
$$

Description logic. Implication (GCI)

A **general concept inclusion (GCI)** is of the form $C \sqsubseteq D$ where C and D are concept descriptions. A GCI $C \sqsubseteq D$ is **valid** in an interpretation Iif $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We shall denote this by $\mathcal{I} \models C \sqsubseteq D$, and call Ia **model** of $C \sqsubseteq D$. A **TBox** is a set of GCIs, and a model of a TBox is a model of all its GCIs. Furthermore, a concept description C is **subsumed** by a concept description D if it is valid in all interpretations. We shall denote this by $C \sqsubseteq D$, and call C a **subsumee** of D, and D a **subsumer** of C. It is easily verified that $\sqsubseteq$ is a quasi-order on $\mathcal{EL}^{\perp}(N_C, N_R)$.

Description logic. LCS and MMSC

Definition (Least Common Subsumer)

Let C and D be two concept descriptions. Then a concept description E is called **least common subsumer (lcs)** of C and D if it satisfies the following properties:

1. E is a common subsumer of C and D, i.e., $C \sqsubseteq E$ and $D \sqsubseteq E$, and
2. for all concept descriptions F, if $C \sqsubseteq F$ and $D \sqsubseteq F$, then $E \sqsubseteq F$.

Definition (Model-Based Most-Specific Concept Description)

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$ be an interpretation, and $X \subseteq \Delta^{\mathcal{I}}$ be a subset of its domain. Then a concept description C is called *model-based most-specific concept description (mmsc)* of X in $\mathcal{I}$ if it satisfies the following conditions:

1. $X \subseteq C^{\mathcal{I}}$, and
2. for all concept descriptions D, if $X \subseteq D^{\mathcal{I}}$, then $C \sqsubseteq D$.

Pattern structure with a description logic ($\mathcal{EL}^{\perp}$)

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation. Then the following triple is a pattern structure:

$$(\Delta^{\mathcal{I}}, (\mathcal{EL}^{\perp}(N_C, N_R), \sqcap), (\cdot)^{\mathcal{I}}),$$

where $\sqcap \colon \mathcal{EL}^{\perp}(N_C, N_R) \times \mathcal{EL}^{\perp}(N_C, N_R) \to \mathcal{EL}^{\perp}(N_C, N_R)$ is the least common subsumer operation,
$(\cdot)^{\mathcal{I}} \colon \wp(\Delta^{\mathcal{I}}) \to \mathcal{EL}^{\perp}(N_C, N_R)$ is the model-based most specific concept (mmsc) description mapping.

Interpretation operator and mmsc define a Galois connection between the powerset of objects and the set of "descriptions", i.e., $\mathcal{EL}^{\perp}$-formulas.

Having this, one has closure operator, (pattern) concepts, taxonomy (given by the lattice of pattern concepts), implications, hypotheses, association rules, ...

Conclusions

FCA and pattern structures give convenient tools for

1. construction and visualization of taxonomies of subject domains
2. representation of implicative dependencies: implications, association rules, hypotheses
3. knowledge discovery (taxonomies, implications, association rules, hypotheses) with relational data

Conclusions

FCA and pattern structures give convenient tools for

1. construction and visualization of taxonomies of subject domains
2. representation of implicative dependencies: implications, association rules, hypotheses
3. knowledge discovery (taxonomies, implications, association rules, hypotheses) with relational data

Thank you!

References

- S.O.Kuznetsov, *On Computing the Size of a Lattice and Related Decision Problems.* Order, vol. 18, no. 4, pp. 313-321 (2001).
- B.Ganter, S.O.Kuznetsov, *Pattern Structures and Their Projections* Proc. ICCS (2001).
- B.Ganter, S.O.Kuznetsov, *Hypotheses and Version Spaces* Proc. ICFCA (2003).
- S.O.Kuznetsov, *Complexity of Learning in Concept Lattices from Positive and Negative Examples.* Discr. Appl. Math., vol. 142, pp. 111–125 (2004).
- M.A.Babin, S.O.Kuznetsov, *Computing premises of a minimal cover of functional dependencies is intractable.* Discr. Appl. Math., vol. 161(6), pp. 742-749 (2013).