Why and how knowledge discovery can be useful for solving problems with CBR

Amedeo Napoli

EPI Orpailleur
LORIA – CNRS – INRIA Nancy Grand Est – Nancy Université
BP 70239, 54506 Vandœuvre-lès-Nancy
Email: Amedeo.Napoli@loria.fr

ICCBR 2010 Conference
July, 21st 2010, Alessandria, Italia

Summary

The Orpailleur Team at LORIA

- Knowledge discovery in databases: symbolic and numerical techniques, text mining, knowledge mining (second-order mining).
- Knowledge representation and reasoning: semantic web technologies, description logics, classification and case-based reasoning, ontology engineering,
- Applications in agronomy, biology, chemistry, cooking (Taaable), medicine, pharmacogenomics...

The basics of Knowledge Discovery in Databases (KDD)

- The goal of knowledge discovery in databases –KDD– is to extract from large databases patterns that are significant and reusable.
- These patterns can be take different forms, e.g. classes of individuals, itemsets, association rules, functional dependencies.
- The KDD process is iterative and interactive and is (usually) guided by a domain expert –the analyst– on the basis of his/her knowledge and experience.
- The extracted patterns are interpreted and in sequence represented as knowledge units within a knowledge representation formalism to be reused in problem-solving activities.

The KDD process

The KDD process is iterative, interactive, and guided by an analyst.

Data
↓ selection and preparation of data
↓ cleaning and formatting the data
Prepared data
↓ data mining operations
↓ numerical and symbolic methods
Discovered patterns
↓ interpretation / evaluation
↓ representation of discovered patterns
Knowledge units
↓
Knowledge systems (problem-solving, ontologies)

Symbolic and numeric methods for knowledge discovery

- Symbolic methods for KDD:
  - Formal Concept Analysis (FCA, design of concept lattices) and extensions of FCA, i.e. Relational Concept Analysis (RCA), FCA guided by similarity, pattern structures.
  - Itemset search (frequent and rare itemsets) and extraction of association rules.
  - Extraction of sequential patterns, graph mining, skylines...
- Numerical methods for KDD:
  - Hidden Markov Models of order 1 and 2 (HMM),
  - K-means, decision trees, neural networks, SVM, statistics, data analysis, etc.

What can say a binary table?

- A. Napoli, A smooth introduction to symbolic methods for knowledge discovery, in Handbook of Categorization in Cognitive Science, H. Cohen and C. Lefebvre editors, Elsevier, pages 913–933, 2005.

What can say a binary table?

Extracting itemsets from a 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	

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 support is a monotonously decreasing function.

The principles of the APriori algorithm

- The search begins with the frequent itemsets of length 1.
- The frequent itemsets are recorded and combined together to form the candidate itemset of greater length, and the process continues in the same way.
- Two fundamental and dual principles:
  $\longrightarrow$ Every sub-itemset of a frequent itemset is a frequent itemset.
  $\longrightarrow$ Every super-itemset of a non frequent itemset is non frequent.

What can say a binary table?

Extracting association rules (AR) from itemsets

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	

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) ...

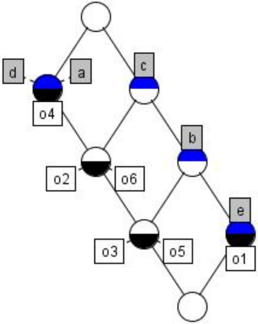
The Coron Platform (http://coron.loria.fr)

- The Coron platform includes a collection of standard and specific algorithms for itemset search:
- Standard algorithms: Apriori, Close, Titanic, Charm, Eclat... Specific algorithms: Zart, Touch, Talky-G, Snow, rare itemset mining (BTB)...
- For association rule extraction and AR basis design: informative rules, generic basis, informative basis, rare rules.
- Modules for data preparation of data and result interpretation: filtering data and results by object, attribute, support, confidence, lift, etc.

- L. Szathmary, P. Valtchev, A. Napoli, and R. Godin, Constructing Iceberg Lattices from Frequent Closures Using Generators, in Proceedings of Discovery Science (DS 2008), Springer LNCS 5255, pages 136-147, 2008.
- L. Szathmary, P. Valtchev, A. Napoli and R. Godin. Efficient Vertical Mining of Frequent Closures and Generators. In Proceedings of the 8th International Symposium on Intelligent Data Analysis (IDA-2009), Springer LNCS 5772, 2009.

Introducing Formal Concept Analysis

Designing a concept lattice from a 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	

Formal context and derivation operators

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	

- Formal context: $\mathbb{K} = (\mathtt{G}, \mathtt{M}, \mathtt{I})$ where $\mathtt{G}$ is a set of objects, $\mathtt{M}$ is a set of attributes,

$\mathtt{I} \subseteq \mathtt{G} \times \mathtt{M}$ is a binary relation.

- Two derivation operators:

$\mathtt{A}' = \{\mathtt{m} \in \mathtt{M} / \forall \mathtt{g} \in \mathtt{A}, \mathtt{gIm}\}$

$\{\mathtt{o}_2, \mathtt{o}_3\}' = \{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$

$\mathtt{B}' = \{\mathtt{g} \in \mathtt{G} / \forall \mathtt{m} \in \mathtt{B}, \mathtt{gIm}\}$

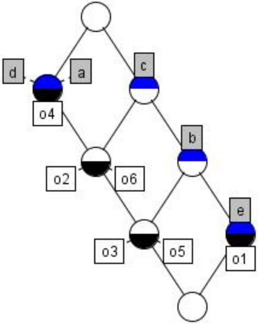
$\{\mathtt{a}, \mathtt{c}, \mathtt{d}\}' = \{\mathtt{o}_2, \mathtt{o}_3, \mathtt{o}_5, \mathtt{o}_6\}$

- Galois connection:

$.'' : \wp(\mathtt{G}) \longrightarrow \wp(\mathtt{M})$

$.'' : \wp(\mathtt{M}) \longrightarrow \wp(\mathtt{G})$

Formal concept and concept lattice



- Formal concept:

  $(\mathtt{A}, \mathtt{B})$: $\mathtt{A}' = \mathtt{B}$ and $\mathtt{B}' = \mathtt{A}$

  $(\{o_2, o_3, o_5, o_6\}, \{a, c, d\})$

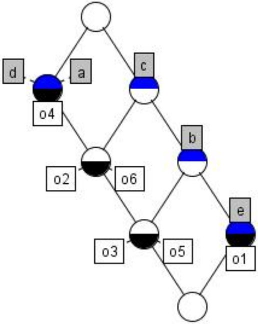
- $\mathfrak{B}(\mathbb{K})$, set of all concepts from $\mathbb{K}$ is associated with a subsumption relation $\sqsubseteq$:

  $(\mathtt{A1}, \mathtt{B1}) \sqsubseteq (\mathtt{A2}, \mathtt{B2})$ iff $\mathtt{A1} \subseteq \mathtt{A2}$ (dually $\mathtt{B2} \subseteq \mathtt{B1}$)

  $(\{o_2, o_3, o_5, o_6\}, \{a, c, d\}) \sqsubseteq (\{o_2, o_3, o_4, o_5, o_6\}, \{a, d\})$

- Concept Lattice: $(\mathfrak{B}(\mathbb{K}), \sqsubseteq)$

Rules can be extracted from the concept lattice



- Rules between local attributes: local attributes are equivalent.
  - $\{a\} \longleftrightarrow \{d\}$
- Hierarchical rules: local attributes imply inherited attributes.
  - $\{e\} \longrightarrow \{b\}$
  - $\{b\} \longrightarrow \{c\}$

Knowledge discovery and knowledge representation are intertwined

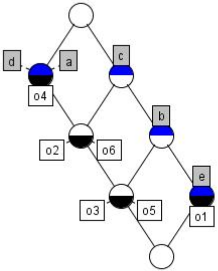
- A. Coulet, M. Smaïl-Tabbone, P. Benlian, A. Napoli, and M.-D. Devignes. Ontology-guided data preparation for discovering genotype-phenotype relationships, in BMC Bioinformatics, 9(S4):S3, 2008.
- J. Lieber, A. Napoli, L. Szathmary, and Y. Toussaint, First Elements on Knowledge Discovery guided by Domain Knowledge (KDDK), in Concept Lattices and Their Applications, S. Ben Yahia, E. Mephu Nguifo, and R. Belohlavek editors, Springer LNCS 4923, pages 22-41, 2008.
- F. Pennerath, G. Niel, P. Vismara, P. Jauffret, C. Laurenço, and A. Napoli. A graph-mining method for the evaluation of bond formability, in ACS Journal of Chemical Information and Modeling, 50(2):221–239, 2010.

A concept lattice can support an ontology schema

An ontology schema $\mathcal{O}$ consists of:

- Recall that:
  The extracted patterns are interpreted and in sequence represented as knowledge units within a knowledge representation formalism to be reused for problem-solving.
- A concept lattice can support an ontology schema which consists of:
  - A set $\mathtt{S_C}$ of concepts organized within a hierarchy $\mathtt{H}$,
  - concepts are ordered in $\mathtt{H}$ by a subsumption relation $\mathtt{C_1} \sqsubseteq \mathtt{C_2}$ (reflexive, transitive, and without cycles),
  - a set $\mathtt{S_R}$ of binary relations specified by pairs $(\mathtt{D}, \mathtt{R})$ of concept domains and ranges, and organized within a relation hierarchy.

From a concept lattice to an ontology schema



Concepts have to be represented within a KR formalism (e.g. DLs):

- $\mathtt{C}_1 \equiv \exists\mathtt{hasAwR.D} \sqcap \exists\mathtt{hasAwR.A}$

where $\mathtt{hasAwR}$ stands for "$\mathtt{has\ Attribute\ with\ Range}$"

- $\mathtt{C}_2 \equiv \exists\mathtt{hasAwR.C}$
- $\mathtt{C}_3 \sqsubseteq \mathtt{C}_1 \sqcap \mathtt{C}_2$

  $\mathtt{C}_3 \equiv \exists\mathtt{hasAwR.D} \sqcap \exists\mathtt{hasAwR.A} \sqcap \exists\mathtt{hasAwR.C}$
- $\mathtt{C}_4 \equiv \mathtt{C}_2 \sqcap \exists\mathtt{hasAwR.B}$

  $\mathtt{C}_4 \sqsubseteq \mathtt{C}_2$
- $\mathtt{C}_5 \sqsubseteq \mathtt{C}_3 \sqcap \mathtt{C}_4$
- $\mathtt{C}_6 \equiv \mathtt{C}_4 \sqcap \exists\mathtt{hasAwR.E}$

  $\mathtt{C}_6 \sqsubseteq \mathtt{C}_4$
- $\mathtt{hasAwR}(\mathtt{x}_2, \mathtt{x}_3), \mathtt{C}_2(\mathtt{x}_2) \models \mathtt{C}(\mathtt{x}_3)$

Knowledge Discovery guided by Domain Knowledge

- No knowledge discovery without domain knowledge!
- At each step of the KDD process, domain knowledge can be used for improving the KDD process.
- An objective of the Orpailleur team is to extract complex knowledge units from complex data being guided by domain knowledge for achieving knowledge discovery guided by domain knowledge (KDDK).
- Domain knowledge can be embedded within general ontologies (e.g. upper ontologies) and/or specific ontologies (e.g. relative to data).

Knowledge Discovery guided by Domain Knowledge

	Data
domain-based	↓ selection and preparation of data
transformations	↓ cleaning and formatting data
	Prepared data
models	↓ data mining operations
similarity, thresholds	↓ numeric and symbolic methods
	Discovered patterns
expertise	↓ interpretation / evaluation
	↓ representation of discovered patterns
	Knowledge units
reasoning	↓
	Knowledge systems (problem-solving, ontologies)

Mining and Analyzing Complex Data

- Extensions of Formal Concept Analysis (FCA) for analyzing complex data with: multi-valued attributes, relational data, graphs, textual documents...
- FCA guided by similarity
- Pattern Structures
- Relational Concept Analysis (RCA) and ontology engineering.

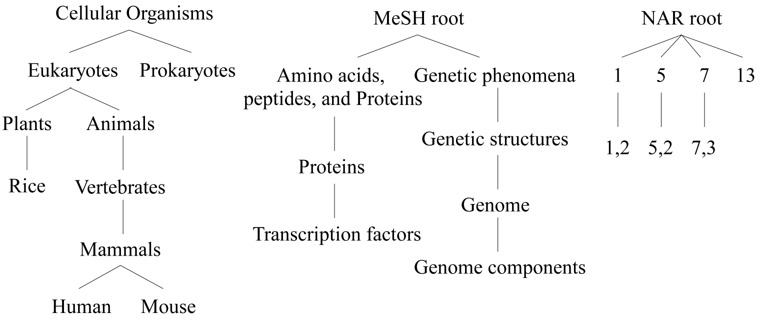
FCA guided by similarity

- N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone. Many-valued concept lattices for conceptual clustering and information retrieval, in Proceedings of 18th European Conference on Artificial Intelligence (ECAI-08), IOS Press, pages 127–131, 2008.
- N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone, Using Domain Knowledge to Guide Lattice-based Complex Data Exploration, in Proceedings of 19th European Conference on Artificial Intelligence (ECAI-10), 2010.

Information retrieval guided by FCA

- Information Retrieval for accessing biological resources on the Web (e.g. databases, thesauri, sets of documents) is a daily task of first importance for *in silico* biologists.
- As for Web services, the BioRegistry catalog includes a collection of annotated resources for providing a guided access to these biological resources.
- The annotation process follows the DCMI model and proposes annotations from semantic resources such as:
  - Taxonomies or hierarchical classifications: NCBI and NAR
  - Thesaurus: MeSH
- The role of Formal Concept Analysis in information retrieval is well known, but FCA has to be adapted in this special case for allowing retrieval of resources guided by domain knowledge.

Semantic Resources associated with BioRegistry



(from NCBI Taxonomy) ***(from MeSH thesaurus)*** ***(from NAR hierarchy of categories)***

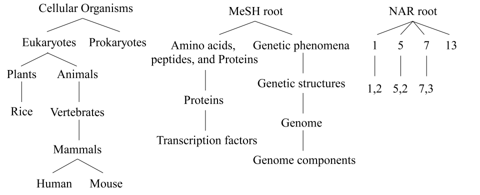
Excerpt of the BioRegistry Catalog

- Multivalued context $(\mathtt{G}, \mathtt{M}, \mathtt{W}, \mathtt{I})$:
  $\mathtt{I} \subseteq \mathtt{G} \times \mathtt{M} \times \mathfrak{P}(\mathtt{W})$: $\mathtt{g}\ \mathtt{I}\ \mathtt{m}$ or $\mathtt{m}(\mathtt{g}) \subseteq \mathfrak{P}(\mathtt{W})$

BD \ Metadata	organism of interest	content quality	semantic resources
BD_1	Amphibians, Fishes	Complete	NCBI
BD_2	Amphibians, Fishes	Complete, Updated	NCBI
BD_3	Amphibians, Mammals		NCBI
BD_4	Birds, Mammals	Updated	
BD_5	Amphibians, Mammals	Complete, Updated	
BD_6	Birds, Mammals	Complete	GO
BD_7	Birds, Mammals	Complete, Updated	GO, NCBI
BD_8	Birds, Mammals		NCBI

Annotation of resources

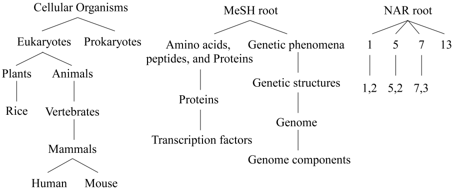
BD \ Metadata	Subject (MeSH)	Organism of interest (NCBI)	Category (NAR)
ExInt	Genome components	Eukaryotes	1.2
HSD	Proteins	Human	7.3
rRNDB	Genomics	Prokaryotes	5.2
SpliceDB	Genome components	Mammals	1.2
CropNet		Plants	13
GOLD	Genomics		5.2
INE		Rice	13
TRANSCompel	Transcription factors	Vertebrates	1.2



(from NCBI Taxonomy) *(from MeSH thesaurus)* *(from NAR hierarchy of categories)*

Computing similarity in a hierarchy

- Similarity can be computed w.r.t. domain knowledge.
- Jaccard similarity in a taxonomy (hierarchy of terms):
  $\mathtt{sim}(\mathtt{v_i}, \mathtt{v_j}) = \frac{|\mathtt{Ancestors}(\mathtt{v_i}) \cap \mathtt{Ancestors}(\mathtt{v_j})|}{|\mathtt{Ancestors}(\mathtt{v_i}) \cup \mathtt{Ancestors}(\mathtt{v_j})|}$
- A threshold θ is set on for controlling similarity:
  $\mathtt{v_i} \simeq \mathtt{v_j}$ iff $\mathtt{sim}(\mathtt{v_i}, \mathtt{v_j}) \geq \theta$
- $\mathtt{Human} \simeq \mathtt{Mammals}$ when $\theta = 0.5$ in NCBI taxonomy.



(from NCBI Taxonomy) *(from MeSH thesaurus)* *(from NAR hierarchy of categories)*

Sharing multi-valued attributes

- $\mathtt{g}_1$ and $\mathtt{g}_2$ are sharing attribute $\mathtt{m}$ iff $\mathtt{m}(\mathtt{g}_1) \simeq \mathtt{m}(\mathtt{g}_2)$:
  When $\theta = 0.5$:
- HSD and SpliceDB share $(\mathtt{NCBI}, \{\mathtt{Human}, \mathtt{Mammals}\})$
- HSD, SpliceDB, and TRANSCompel share $(\mathtt{NCBI}, \{\mathtt{Human}, \mathtt{Mammals}, \mathtt{Vertebrates}\})$

BD \ Metadata	Subject (MeSH)	Organism of interest (NCBI)	Category (NAR)
ExInt	Genome components	Eukaryotes	1.2
HSD	Proteins	Human	7.3
rRNDB	Genomics	Prokaryotes	5.2
SpliceDB	Genome components	Mammals	1.2
CropNet		Plants	13
GOLD	Genomics		5.2
INE		Rice	13
TRANSCompel	Transcription factors	Vertebrates	1.2

Partial ordering on multi-valued attributes

- Inclusion based on similarity: $\subseteq_\theta$
  - $\mathtt{B}_1 \subseteq_\theta \mathtt{B}_2$ iff $\forall(\mathtt{m}, \mathtt{W}_1) \in \mathtt{B}_1, \exists(\mathtt{m}, \mathtt{W}_2) \in \mathtt{B}_2$ $\mathtt{s.t.}$ $\mathtt{W}_2 \subseteq \mathtt{W}_1$
  - With $\theta = 0.5$
    $\{(\mathtt{NCBI}, \{\mathtt{Vertebrates}, \mathtt{Mammals}\}), (\mathtt{NAR}, \{1.2\})\} \subseteq_\theta$
    $\{(\mathtt{NCBI}, \{\mathtt{Mammals}\}), (\mathtt{NAR}, \{1.2\}), (\mathtt{MeSH}, \{\mathtt{Gene\ Comp.}\})\}$
- $(\mathtt{M} \times \mathfrak{P}(\mathtt{W}), \subseteq_\mathtt{S})$ is a partially ordered set.

Maximal sets of objects and multi-valued attributes

- Maximal sets of objects:
  Given $\mathtt{A} \subseteq \mathtt{G}, \mathtt{B} \subseteq \mathtt{M}, \mathtt{m} \in \mathtt{M}, \mathtt{W}_{\mathtt{m}}^{\mathtt{A}} = \{\mathtt{m}(\mathtt{g}) \in \mathtt{W},\ \mathtt{g} \in \mathtt{A}\} \subseteq \mathtt{W}$:
  $\mathtt{A}$ shares $(\mathtt{m}, \mathtt{W}_{\mathtt{m}}^{\mathtt{A}})$ iff $\forall \mathtt{g}_{\mathtt{i}}, \mathtt{g}_{\mathtt{j}} \in \mathtt{A},\ \mathtt{m}(\mathtt{g}_{\mathtt{i}}) \simeq \mathtt{m}(\mathtt{g}_{\mathtt{j}})$
- Maximal set of similar objects in $\mathtt{A}$ for $\mathtt{m}$:
  $\mathfrak{R}(\mathtt{A}, \mathtt{m}) = \{\mathtt{g}_{\mathtt{i}} \in \mathtt{G} \mid \mathtt{m}(\mathtt{g}_{\mathtt{i}}) \simeq \mathtt{m}(\mathtt{g}),\ \forall \mathtt{g} \in \mathtt{A}\}$
  $\mathfrak{R}(\mathtt{A}, \mathtt{B}) = \bigcap_{\mathtt{m} \in \mathtt{B}} \mathfrak{R}(\mathtt{A}, \mathtt{m})$
- Maximal set of objects including $\mathtt{A}$ and sharing $\mathtt{m}$:
  $\mathfrak{R}_{\mathtt{v}}(\mathtt{A}, \mathtt{m}) = \mathfrak{R}(\mathtt{A}, \mathtt{m}) \setminus \{\mathtt{g}_{\mathtt{i}} \in \mathfrak{R}(\mathtt{A}, \mathtt{m}) \mid \exists \mathtt{g}_{\mathtt{j}} \in \mathfrak{R}(\mathtt{A}, \mathtt{m}) \text{ and } \mathtt{m}(\mathtt{g}_{\mathtt{i}}) \not\simeq \mathtt{m}(\mathtt{g}_{\mathtt{j}})\}$
  $\mathfrak{R}_{\mathtt{v}}(\mathtt{A}, \mathtt{B}) = \bigcap_{\mathtt{m} \in \mathtt{B}} \mathfrak{R}_{\mathtt{v}}(\mathtt{A}, \mathtt{m})$
- Maximal sets of values of similar attributes:
  $\gamma(\mathtt{A}, \mathtt{m}) = \{\mathtt{m}(\mathtt{g}) \in \mathtt{W},\ \mathtt{g}_{\mathtt{i}} \in \mathfrak{R}_{\mathtt{v}}(\mathtt{A}, \mathtt{m})\}$
  When objects in $\mathtt{A}$ share $\mathtt{m}$ then they share $(\mathtt{m}, \gamma(\mathtt{A}, \mathtt{m}))$.

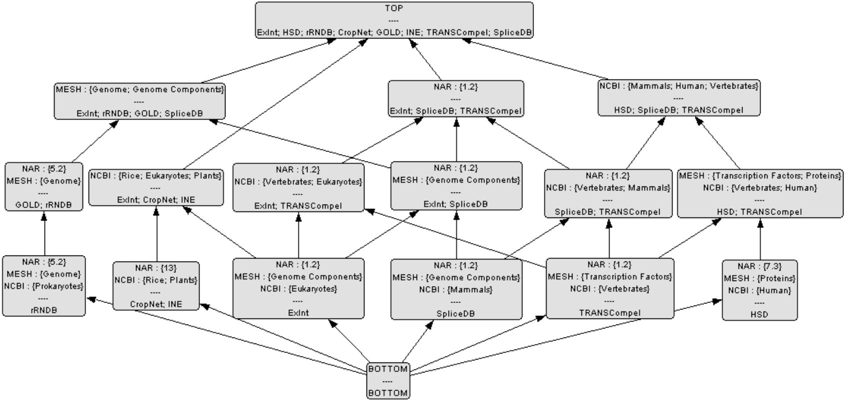
Derivation operators and Galois connection

- $\mathtt{A}^{\uparrow} = \{(\mathtt{m}, \gamma(\mathtt{A}, \mathtt{m})) \in \mathtt{M} \times \mathfrak{P}(\mathtt{W}) \mid \gamma(\mathtt{A}, \mathtt{m}) \neq \emptyset\}$
- With $\theta = 0.5$:
  $\{\mathtt{INE}, \mathtt{CropNet}\}^{\uparrow} = \{(\mathtt{NCBI}, \{\mathtt{Plants}, \mathtt{Rice}\}), (\mathtt{NAR}, \{\mathtt{13}\})\}$
- $\mathtt{B}^{\downarrow} = \mathfrak{R}_{\mathtt{v}}(\{\mathtt{g} \in \mathtt{G} \mid \forall\ (\mathtt{m}, \mathtt{V_m}) \in \mathtt{B},\ \mathtt{m}(\mathtt{g}) \simeq \mathtt{w},\ \forall \mathtt{w} \in \mathtt{V_m}\},\ \mathtt{B})$
- With $\theta = 0.5$:
  $\{(\mathtt{NCBI}, \{\mathtt{Plants}, \mathtt{Rice}\}), (\mathtt{NAR}, \{\mathtt{13}\})\}^{\downarrow} = \{\mathtt{INE}, \mathtt{CropNet}\}$
- The pair $(^{\uparrow}, ^{\downarrow})$ define a Galois connection between $(\mathfrak{P}(\mathtt{G}), \subseteq)$ and $(\mathfrak{P}(\mathtt{M} \times \mathfrak{P}(\mathtt{W})), \subseteq_{\theta})$

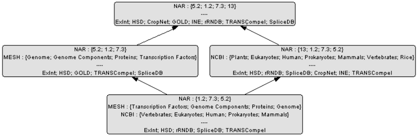
Multi-valued concept lattice

- Multi-valued concept:
  $(\mathtt{A}, \mathtt{B}), \mathtt{A} \subseteq \mathtt{G}, \mathtt{B} \subseteq \mathtt{M} \times \mathfrak{P}(\mathtt{W})$ such as $\mathtt{A}^{\uparrow} = \mathtt{B}$ and $\mathtt{B}^{\downarrow} = \mathtt{A}$
- With $\theta = 0.5$:
  $(\{\mathtt{INE}, \mathtt{CropNet}\}, \{(\mathtt{NCBI}, \{\mathtt{Plants}, \mathtt{Rice}\}), (\mathtt{NAR}, \{13\})\})$
- Partial ordering $\sqsubseteq_\theta$:
  $(\mathtt{A}_1, \mathtt{B}_1) \sqsubseteq_\theta (\mathtt{A}_2, \mathtt{B}_2)$ if $\mathtt{A}_1 \subseteq \mathtt{A}_2$ (dually $\mathtt{B}_2 \sqsubseteq_\theta \mathtt{B}_1$).
  $(\{\mathtt{SpliceDB}, \mathtt{TRANSCompel}\}, \{(\mathtt{NCBI}, \{\mathtt{Mammals}, \mathtt{Vertebrates}\}), (\mathtt{NAR}, \{1.2\})\})$
  $\sqsubseteq_\theta$
  $(\{\mathtt{HSD}, \mathtt{SpliceDB}, \mathtt{TRANSCompel}\}, \{(\mathtt{NCBI}, \{\mathtt{Human}, \mathtt{Mammals}, \mathtt{Vertebrates}\})\})$
- Lattice of multi-valued concepts:
  $(\mathfrak{B}(\mathtt{G}, \mathtt{M}, \mathtt{I}, \mathtt{W}), \sqsubseteq_\theta)$

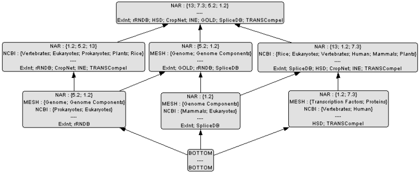
Multi-valued concept lattice



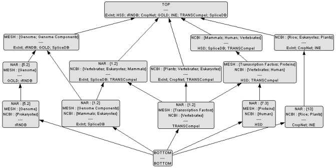
$(\mathfrak{B}(\mathtt{G}, \mathtt{M}, \mathtt{I}, \mathtt{W}), \sqsubseteq_\theta)$ with $\theta = 0.5$



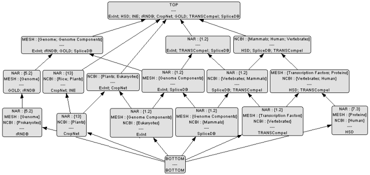
$\theta = 0$ (4 concepts)



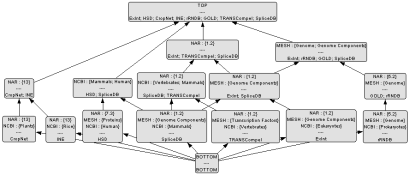
$\theta = 0.2$ (8 concepts)



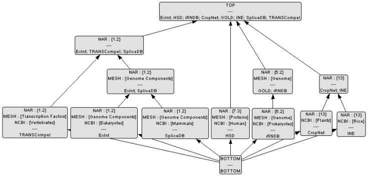
$\theta = 0.4$ (14 concepts)



$\theta = 0.6$ (17 concepts)

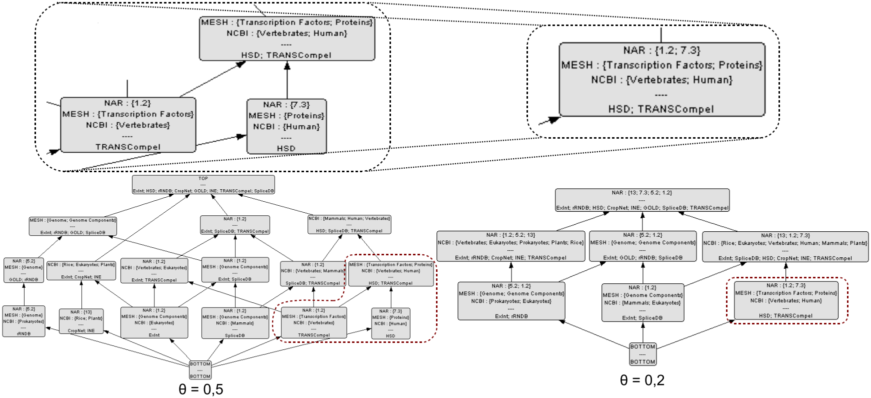


$\theta = 0.8$ (16 concepts)



$\theta = 1$ (13 concepts)

Static and dynamic navigation in a multivalued lattice



Pattern Structures

- B. Ganter and S.O. Kuznetsov. Pattern Structures and Their Projections, in Proceedings of ICCS 2001, LNCS 2120, Springer, pages 129–142, 2001.
- M. Kaytoue, Z. Assaghir, N. Messai, and A. Napoli. Two Complementary Classication Methods for Designing a Concept Lattice from Interval Data, in Proceedings of FoIKS, LNCS 5956, Springer, pages 345–362, 2010.
- M. Kaytoue, S. Duplessis, S. Kuznetsov, and A. Napoli. Two FCA-Based Methods for Mining Gene Expression Data, in Proceedings of ICFCA 2009, LNAI 5548, Springer, pages 251–266, 2009.
- Z. Assaghir, M. Kaytoue, A. Napoli, and H. Prade Managing Information Fusion with Formal Concept Analysis, Proceedings of 7th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2010), LNCS, Springer, to be published.
- M. Kaytoue, Z. Assaghir, S. Kuznetsov, and A. Napoli Embedding tolerance relations in Formal Concept Analysis – An application in information fusion, in Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010), to be published.

Similarity and tolerance relations

- Similarity of documents $\mathtt{x}$ and $\mathtt{y}$ can be defined by non emptiness of the set of their common attributes: $\mathtt{x}' \cap \mathtt{y}' \neq \emptyset$.
- The relation of similarity, being naturally reflexive and symmetric, should not be transitive: e.g., children are often similar to both their parents, the latter being very different.
- Defined in this way similarity is reflexive and symmetric, i.e., similarity is a tolerance relation on the set of objects (documents).
- S.O. Kuznetsov. Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research , in B. Ganter et al.(Eds.): Formal Concept Analysis, LNAI 3626, pp. 196-225, 2005.

- For a set $\mathtt{G}$, a binary relation $\mathtt{T} \subseteq \mathtt{G} \times \mathtt{G}$ is called tolerance if:
  (1) $\forall \mathtt{x} \in \mathtt{G}, \mathtt{xTx}$ (reflexivity),
  (2) $\forall \mathtt{x}, \mathtt{y} \in \mathtt{G}, \mathtt{xTy} \longrightarrow \mathtt{yTx}$ (symmetry).
  A set $\mathtt{G}$ with tolerance $\mathtt{T}$ is called the space of tolerance and denoted by $\mathtt{G}_\mathtt{T}$.
- A subset $\mathtt{K} \subseteq \mathtt{G}$ is called a class of tolerance if:
  (1) $\forall \mathtt{x}, \mathtt{y} \in \mathtt{K}, \mathtt{xTy}$,
  (2) $\forall \mathtt{z} \notin \mathtt{K}, \exists \mathtt{u} \in \mathtt{K}, \neg(\mathtt{zTu})$.
  An arbitrary subset of a class of tolerance is called a preclass.

Pattern structures

$(G, (D, \sqcap), \delta)$

- G is a set of *objects*
- $(D, \sqcap)$ is a meet-semilattice of descriptions or patterns
- $\delta : G \rightarrow D$ is a mapping that associates with each object $g \in G$ its description $\delta(g) \in D$

Subsumption Relation in $(D, \sqcap)$:

$$c \sqsubseteq d \iff c \sqcap d = c \qquad \forall c, d \in D$$

The infimum $\sqcap$ is a *similarity operator* returning a description which represents the similarity of its arguments.
For FCA, the corresponding description representing similarity is:

$$\{a, b\} \cap \{a, d\} = \{a\}$$

Intervals are patterns

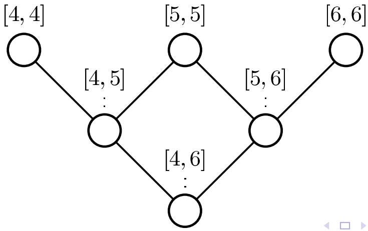
Given two intervals $[a_1, b_1]$ and $[a_2, b_2]$, their infimum is:

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

The partial ordering is given by:

$$\begin{array}{rcrcl} [a_1, b_1] \sqsubseteq [a_2, b_2] & \Longleftrightarrow & [a_1, b_1] \sqcap [a_2, b_2] & = & [a_1, b_1] \\ [4, 5] \sqsubseteq [5, 5] & \Longleftrightarrow & [4, 5] \sqcap [5, 5] & = & [4, 5] \end{array}$$

Inf.-semi-lattice of intervals $(D, \sqcap)$ or $(D, \sqsubseteq)$:



Vectors of intervals are patterns

- Each object is described by an interval vector of n dimensions: $\delta(g_1) = \langle [5,5], [7,7], [6,6] \rangle$
- Each dimension corresponds to an attribute.
- A canonical order of vector dimensions is assumed.
- The meet of two interval vectors:

$$e = \langle [a_i, b_i] \rangle_{i \in [1,p]} \quad \text{and} \quad f = \langle [c_i, d_i] \rangle_{i \in [1,p]}$$

$$e \sqcap f = \langle [a_i, b_i] \sqcap [c_i, d_i] \rangle_{i \in [1,p]}$$

$\langle [4,4], [3,4] \rangle \sqcap \langle [2,3], [2,6] \rangle = \langle [2,4], [2,6] \rangle$

- The ordering is satisfied for each dimension:
$\langle [2,4], [2,6] \rangle \sqsubseteq \langle [4,4], [3,4] \rangle$ s.t. $[2,4] \sqsubseteq [4,4]$ and $[2,6] \sqsubseteq [3,4]$

	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

A Galois connection for pattern structures

Two Derivation operators:

- The first operator returns the description representing the similarity of a set of objects:

$$A^{\square} = \bigsqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G$$

- The second operator returns the maximal set of objects whose similarity is represented by a given description:

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$$

Example

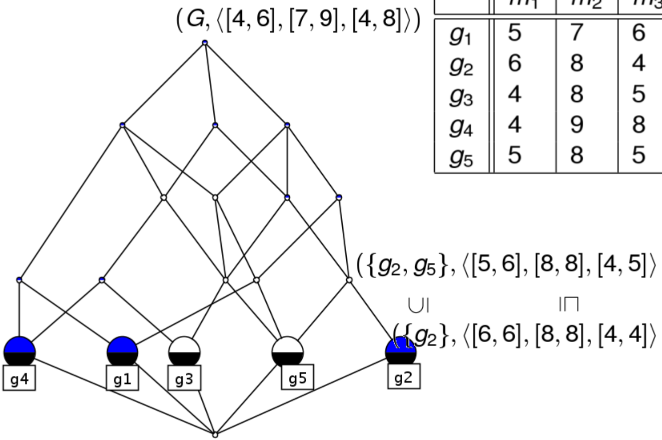
	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

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

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

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

The 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

Introducing a similarity with pattern structures

A similarity can be defined for determining and classifying similar objects:

Similarity on numerical (attribute) values

$$\begin{aligned} a \simeq_\theta b &\Leftrightarrow |a-b| \leq \theta \\ [a_1,b_1] \simeq_\theta [a_2,b_2] &\Leftrightarrow max(b_1,b_2) - min(a_1,a_2) \leq \theta \end{aligned}$$

Examples

$$2 \simeq_2 4, \quad 2 \not\simeq_3 7, \quad [4,5] \simeq_2 [5,6], \quad [4,5] \not\simeq_4 [5,9]$$

- The element $* \in (D, \sqcap)$ denotes dissimilarity.

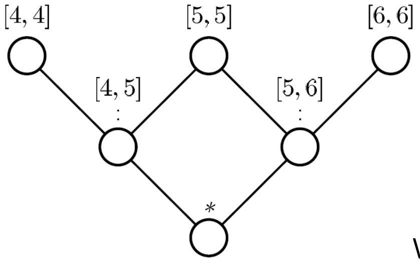
$$c \sqcap d \neq * \iff \text{c and d are similar}$$

$$c \sqcap d = * \iff \text{c and d are dissimilar}$$

- The similarity operator for intervals can be constrained by θ:

$$[a, b] \sqcap_\theta [c, d] = [min(a, c), max(b, d)] \text{ if } max(b, d) - min(a, c) \leq \theta$$

$$[a, b] \sqcap_\theta [c, d] = * \text{ otherwise}$$



With $\theta = 1$

An application

Information fusion and organization of agricultural sources

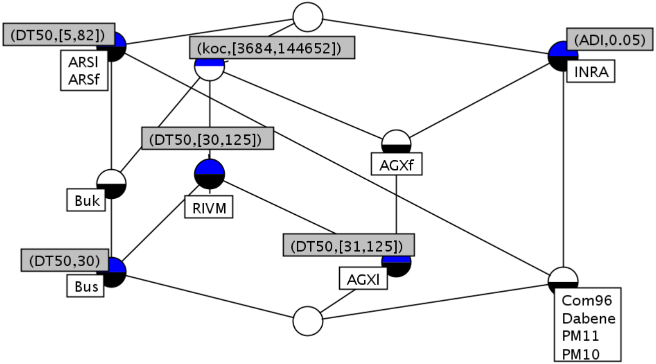
The characteristics of cypermethrin pesticide as given by different information sources:

	DT50 days	koc L/kg	ADI mg/kg.day
BUS	30	10000	*
PM10	5	*	0.05
PM11	5	*	0.05
INRA	*	*	0.05
Dabene	[7,82]	[2000,160000]	0.05
ARSf	[7,82]	[5800,160000]	*
ARSl	[6,60]	[5800,160000]	*
Com96	[7,82]	[2000,160000]	0.05
RIVM	[61,119]	3684	*
BUK	[7,70]	19433	*
AGXf	[14,199]	[26492,144652]	0.05
AGXl	[31,125]	[26492,144652]	0.05

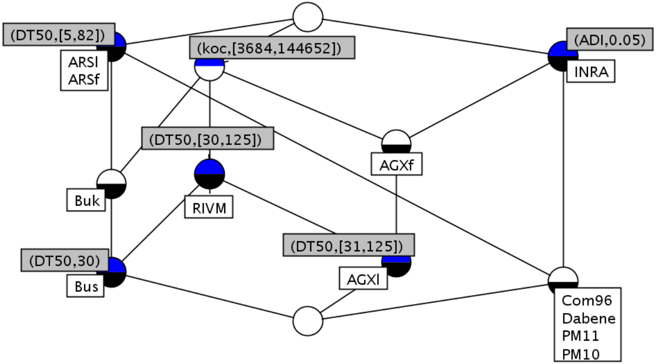
Preparing the data for information fusion and concept lattice design

- Replace missing values by $*$.
- Set the different thresholds θ for each characteristics: here θ is equal respectively to 100, 150000, and 0.
- Compute the concept descriptions w.r.t. the Galois connection for pattern structures.
- Build the concept lattice.

A concept lattice for information



A concept lattice for information



Information fusion: 39 concepts without θ, 28 concepts with θ, and here 11 concepts with a pairwise similarity.

Relational Concept Analysis

- M. Rouane-Hacene, M. Huchard, A. Napoli, and P. Valtchev. A proposal for combining Formal Concept Analysis and description Logics for mining relational data, in Proceedings of ICFCA-2007, LNAI 4390, Springer, pages 51–65, 2007.
- R. Bendaoud, A. Napoli, and Y. Toussaint. Formal Concept Analysis: A unified framework for building and refining ontologies, in Proceedings of the EKAW 2008, LNCS 5268, pages 156–171, 2008.
- M. Rouane-Hacene, A. Napoli, P. Valtchev, Y. Toussaint, and R. Bendaoud. Ontology Learning from Text using Relational Concept Analysis, in International Conference on eTechnologies (MCETECH 08), Montréal, IEEE Computer Society, pages 154–163, 2008.

Introducing Relational Concept Analysis (RCA)

- The objective of RCA is to extend the purpose of FCA for taking into account relations between objects.
- The RCA process relies on the following main points:
  - 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:

- **K** is a set of contexts $\mathcal{K}_i = (\mathtt{G_i}, \mathtt{M_i}, \mathtt{I_i})$
- **R** is a set of relations $\mathtt{r_k} \subseteq \mathtt{O_i} \times \mathtt{O_j}$, 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 $\mathtt{r} \subseteq \mathtt{O_i} \times \mathtt{O_j}$ has a domain and a range:

- $\mathbf{G} = \{\mathtt{G_i} | \mathtt{G_i} \in \mathcal{K}_\mathtt{i} = (\mathtt{G_i}, \mathtt{M_i}, \mathtt{I_i}), \mathcal{K}_\mathtt{i} \in \mathbf{K}\}$
- $\mathtt{dom} : \mathbf{R} \longrightarrow \mathbf{G}$ and $\mathtt{dom}(\mathtt{r}) = \mathtt{G_i}$,
- $\mathtt{ran} : \mathbf{R} \longrightarrow \mathbf{G}$ and $\mathtt{ran}(\mathtt{r}) = \mathtt{G_j}$,
- $\mathtt{rel} : \mathbf{K} \longrightarrow 2^{\mathbf{R}}$ and $\mathtt{rel}(\mathcal{K}_\mathtt{i}) = \{\mathtt{r_k} | \mathtt{dom}(\mathtt{r_k}) = \mathtt{G_i}\}$

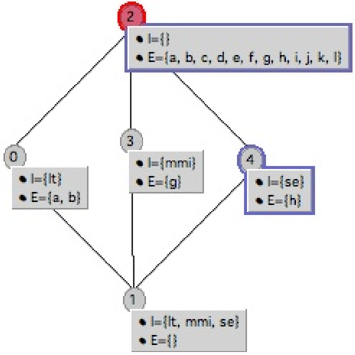
A relational context about paper citations and development

- Context: $\texttt{Kpapers} = \texttt{Papers} \times \texttt{Topics}$
- Relations: $\texttt{cites} : \texttt{Papers} \times \texttt{Papers}$ and $\texttt{develops} : \texttt{Papers} \times \texttt{Papers}$

	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

The Kpapers concept lattice

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



Details on relational scaling

- A relation $\mathtt{r}$ can be considered as a function whose range is a set of objects
  $\mathtt{r} : \mathtt{G_i} \longrightarrow 2^{\mathtt{G_j}}$ where $\mathtt{r}(\mathtt{o_i}) = \{\mathtt{o_{j1}}, \mathtt{o_{j2}}, \ldots, \mathtt{o_{jn}}\}$
- Scaling and notation.
  Suppose that o_{jk} is in relation through r with object o_i, and that o_{jk} is in the extent of concept $\mathtt{C_m}$ of the current lattice, then the relational attribute $\exists\mathtt{r{:}Cm}$ is associated with object o_i.
- Then a new context taking into account relational attributes is built and, accordingly, a new lattice is designed including the new relational attributes.

Details on relational scaling

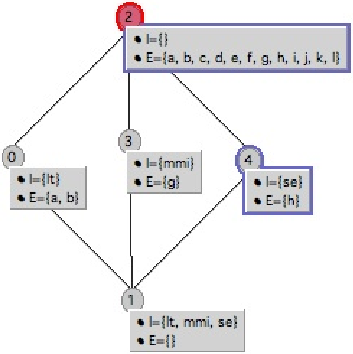
There are two main ways of considering relations:

- universal scaling ($\forall$): $\mathtt{r(o)} \subseteq \mathtt{Extent(C)}$
- existential scaling ($\exists$): $\mathtt{r(o)} \cap \mathtt{Extent(C)} \neq \emptyset$

Some properties of relational scaling:

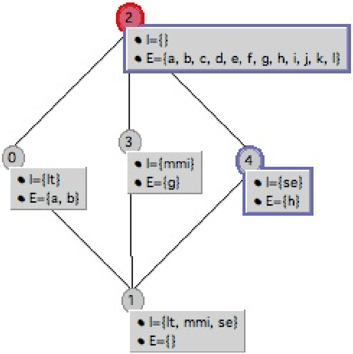
- The homogeneity of concept descriptions is kept: all attributes are considered as binary (even relational attributes).
- Standard algorithms for building concept lattices can be straightforwardly reused.

Relations and relational scaling



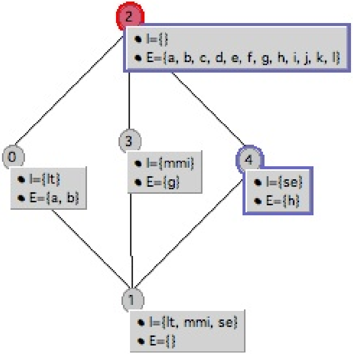
- Object `i` is in relation with object a through relation `cites`.
- Object a is in the extent of concepts C_0 and C_2 of the initial lattice.
- Thus, object `i` is given two new relational attributes, namely $\exists$`cites`:C_0 and $\exists$`cites`:C_2.

Relations and relational scaling



- 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, while object h is in the extent of concepts C_2 and C_4 of the initial lattice.
- 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 and relational lattice design

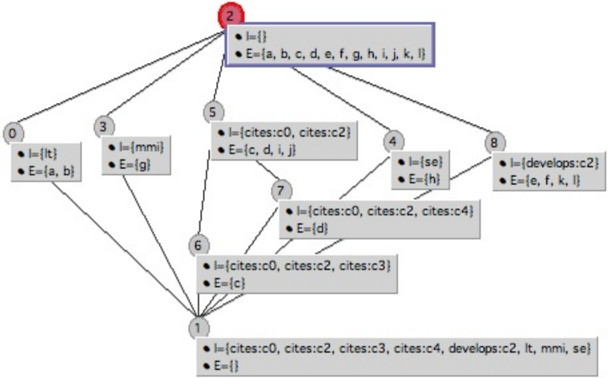


- The same process is applied to `develops`:
- `e` is in relation with `c`, `f` with `d`, `k` with `i`, and `l` with `j`.
- The four objects `e`, `f`, `k`, and `l`, are given the relational attribute $\exists$`develops`:C_2.

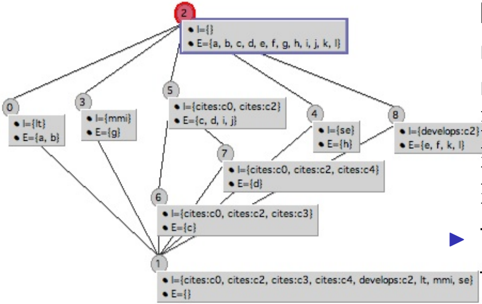
The extended context after the first step

	lt	mmi	se	2	0	3	4	2
a	x							
b	x							
c				x	x	x		
d				x	x		x	
e								x
f								x
g		x						
h			x					
i				x				
j					x			
k								x
l								x

The concept lattice after the first step



Following the construction of the concept lattice

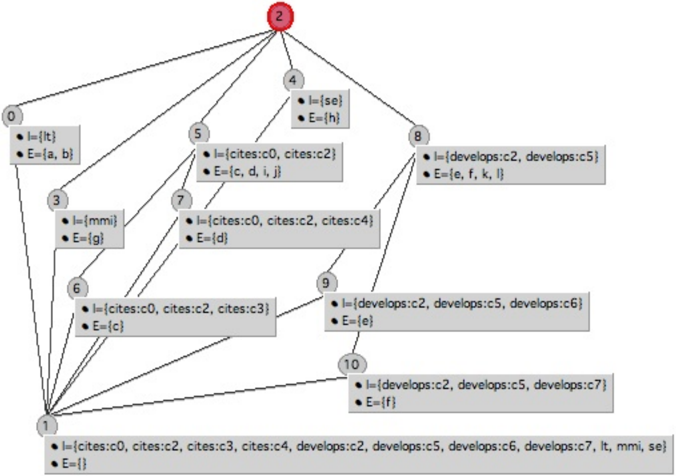


- The process is applied a second time for relations `cites` and `develops`, leading to the creation of new relational attributes, namely $\exists$`develops`:C_2, $\exists$`develops`:C_5, $\exists$`develops`:C_6, and $\exists$`develops`:C_7.
- The object `e` is in relation through `develops` with `c` which is in the extent of concepts C_2, C_5, and C_6.
- For `f`, we have concepts C_2, C_5, and C_7.

The extended context after the second step

	lt	mmi	se	2	0	3	4	2	5	6	7
a	x										
b	x										
c				x	x	x					
d				x	x		x				
e								x	x	x	
f								x	x		x
g		x									
h			x								
i				x	x						
j				x	x						
k								x	x		
l								x	x		

The concept lattice after the second step



From a relational concept lattice to an ontology schema

- The final context and the associated concept lattice are obtained after the second step:
  i.e. relational scaling applied for `cites` and `develops` does not lead to any modification, meaning that the fixpoint of the RCA process is reached and that the final concept lattice can be designed.
- The concepts of this final concept lattice can be represented within a DL-like formalism for designing an ontology schema supported by the lattice.
  It can be noticed that some representational choices have to me made there.

An interpretation of the RCA results within DLs

- Atomic concepts and concept instantiation.
  AboutLatticeTheory: AboutLT(a), AboutLT(b)
  AboutManMachineInterface: AboutMMI(g)
  AboutSoftwareEngineering: AboutSE(h)
- Roles and role instantiation: cites and develops.
- cites(c, a), cites(c, g), cites(d, b), cites(d, h), cites(i, a), cites(j, b)
- develops(e, c), develops(f, d), develops(k, i), develops(l, j).

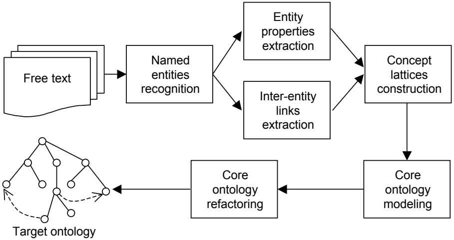
An interpretation of the RCA results within DL)

Defined concepts.

- $\mathtt{C}_5 \equiv \exists\mathtt{cites.AboutLT}$ (cites atleast one paper about LT)
- $\mathtt{C}_6 \equiv \exists\mathtt{cites.AboutLT} \sqcap \exists\mathtt{cites.AboutMMI}$
  (cites at least one paper about LT and at least one paper about MMI)
- $\mathtt{C}_7 \equiv \exists\mathtt{cites.AboutLT} \sqcap \exists\mathtt{cites.AboutSE}$
  (cites at least one paper about LT and at least one paper about SE)
- $\mathtt{C}_8 \equiv \exists\mathtt{develops.}(\exists\mathtt{cites.AboutLT})$
  (develops at least one paper citing at least one paper about LT)
- $\mathtt{C}_9 \equiv \exists\mathtt{develops.}(\exists\mathtt{cites.AboutLT} \sqcap \exists\mathtt{cites.AboutMMI})$
  (develops at least one paper citing at least one paper about LT and MMI)
- $\mathtt{C}_{10} \equiv \exists\mathtt{develops.}(\exists\mathtt{cites.AboutLT} \sqcap \exists\mathtt{cites.AboutSE})$
  (develops at least one paper citing at least one paper about LT and SE)

Ontology engineering guided by RCA

- Input data: free text (scientific abstracts, etc) or structured data (UML, ontology, relational database, etc.) $\Longrightarrow$ contexts and inter-context relations (RCF).
- Output: family of relational concept lattices.
- Post-processing: translation modules, e.g. DL2KB generator, Ontology builder, UML class designer, etc.



A local conclusion on RCA

- RCA is an extension of FCA taking into account relational data within lattice theory.
- FCA builds a concept lattice from binary data and provides efficient algorithms for lattice design.
- So does RCA from binary and relational data, by reusing FCA algorithms.
- A relational concept lattice can be represented as an ontology schema within a Description Logic framework, providing a guideline for ontology engineering from complex (binary and relational) data.

Facets of KDD and FCA in CBR

- M. d'Aquin, F. Badra, S. Lafrogne, J. Lieber, A. Napoli, and L. Szathmary. Case Base Mining for Adaptation Knowledge Acquisition, Proceedings of IJCAI 2007, M.M. Veloso editor, pages 750–755, 2007.
- V. Dufour-Lucier, J. Lieber, E. Nauer, and Y. Toussaint. Text Adaptation Using Formal Concept Analysis, in Proceedings of ICCBR 2010, LNAI 6176, Springer, pages 96–110, 2010.

The CBR process

$$\begin{tikzcd}
\texttt{source} \arrow[d] & & \texttt{target} \arrow[ll, "\textit{retrieval}"'] \arrow[d] \\
\texttt{Sol(source)} \arrow[rr, "\textit{adaptation}"'] & & \texttt{Sol(target)}
\end{tikzcd}$$

- Retrieval: given a target problem `target`, search for a "similar source problems" in the case base, evaluate similarity and rank retrieved source problems.
- Adaptation: adapt the solution of the selected source problem(s) to build the solution of `target`.
- Learning: when interest, retain the new pair `(target,sol(target))` for possible reuse.

A CBR system is also a knowledge-based system

- Each case in the case base can be associated with knowledge units such as annotations, index hierarchy, procedural knowledge, etc.
- A knowledge base –ontological component– may be used for guiding retrieval and adaptation.
- The system can be able to learn new cases and extend its case base.

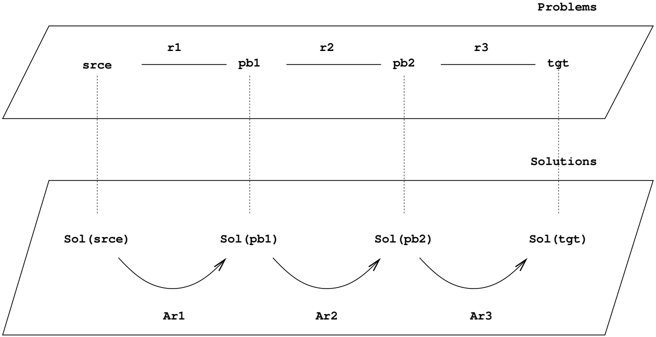
What's in a case?

- A case can be considered as a pair (problem,solution).
- The problem part is a set of problem statements to be fulfilled (preconditions).
- The solution part is a set of solution elements that are satisfied for leading to a solution expression (goals).
- There are semantic relations inside a case (intra-case relations) between some problem statements and some solution elements, e.g. influences, dependency.

What's in a case base?

- There are semantic relations between cases (inter-case relations), e.g. e.g. cases can be be indexed and organized within a case hierarchy.
- Semantic relations between cases: subsumption (specialization, generalization), transformations such as extension, projection, dependence or reference, substitution, alternative or disjunction, causality, etc.

A view of similarity and adaptation based on paths



- Similarity can be reified as a similarity path in the "problem space".
- In a dual way, adaptation can be reified as a adaptation path in the "solution space".

An example of planning problem

- A case `C = (Pres,Goals)` is composed of a set `Pres` of preconditions and a set `Goals` of goals.
- Preconditions are described by properties needed to be fulfilled for applying the case solution.
- The set `Goals` gives the list of goals satisfied after applying the case solution.
- In the query, the preconditions describe the current situation allowing to decide which are the applicable cases.
  The goals are those that have to be satisfied.

From B. Diaz-Agudo and P.A. Gonzales-Calero, Classification Based retrieval using formal concept analysis, in ICCBR-2001, LNAI 2080, pages 173–188, 2001.

An example of planning problem

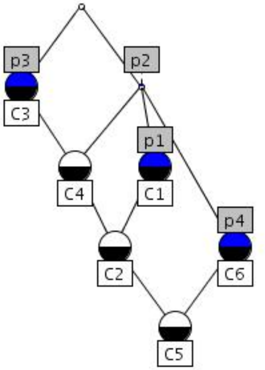
The case base:

- C1: (Pres1: p1, p2) (Goals1: g1, g2, g3)
- C2: (Pres2: p1, p2, p3) (Goals2: g1, g3)
- C3: (Pres1: p3) (Goals3: g1, g2, g4, g5)
- C4: (Pres4: p2, p3) (Goals4: g1, g2)
- C5: (Pres5: p1, p2, p3, p4) (Goals5: g4, g5)
- C6: (Pres6: p2 p4) (Goals6: g1, g3)

Target Problem:
(Pres: p1, p2, p3) (Goals: g1, g2, g3)

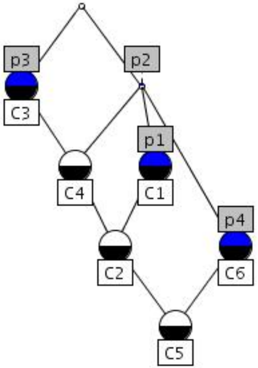
The concept lattice of preconditions

Case / Pres	p1	p2	p3	p4
C1 (Pres1)	x	x		
C2 (Pres2)	x	x	x	
C3 (Pres3)			x	
C4 (Pres4)		x	x	
C5 (Pres5)	x	x	x	x
C6 (Pres6)		x		x



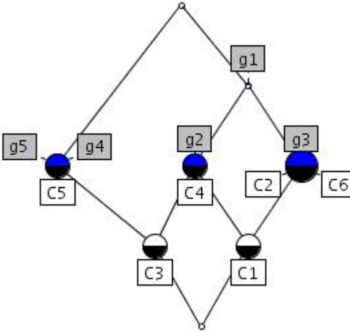
The concept lattice of preconditions

- Relations between cases:
- C2 $\sqsubseteq$ C1 $\sqcap$ C4
- C4 $\sqsubseteq$ C3
- C5 $\sqsubseteq$ C2 $\sqcap$ C6
- Rules: p1 $\longrightarrow$ p2 and p4 $\longrightarrow$ p2
- Ranking of cases w.r.t. the concept lattice of preconditions: C2, C1 or C4, C3, as C5 and C6 cannot be applied.



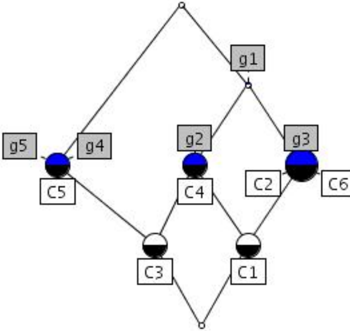
The concept lattice of goals

Case / Goals	g1	g2	g3	g4	g5
C1 (Goals1)	x	x	x		
C2 (Goals2)	x		x		
C3 (Goals3)	x	x		x	x
C4 (Goals4)	x	x			
C5 (Goals5)				x	x
C6 (Goals6)	x		x		

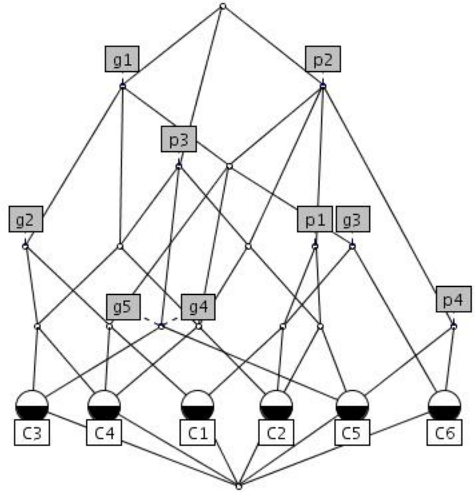


The concept lattice of goals

- Relations between cases:
- $\mathtt{C1} \sqsubseteq \mathtt{C4} \sqcap \mathtt{C2}$ and $\mathtt{C2} \equiv \mathtt{C6}$
- $\mathtt{C3} \sqsubseteq \mathtt{C4} \sqcap \mathtt{C5}$
- Rules: $g4 \longleftrightarrow g5$,
  $g2 \longrightarrow g1$ and $g3 \longrightarrow g1$
- Ranking of cases w.r.t. the concept lattice of goals: C1, C2 or C4 or C6, C3.



The complete lattice including preconditions and goals



Similarity as "proximity" in the lattice

- The lattice provide a view of the structures and dependencies, i.e. specialization and generalization, between cases.
- Rules extracted from the concept lattice can be used to complete user queries in an interactive mode.
- Concepts provide maximal groupings of cases w.r.t. maximal sets of properties.
- In this way, the concept lattice as a case organization structure corresponds to a representational approach of CBR where proximity means similarity...

Another point of view on similarity between cases

- A case $\mathtt{C_i} = (\mathtt{Pres}(\mathtt{C_i}), \mathtt{Goals}(\mathtt{C_i}))$ is similar to a case $\mathtt{C_j} = (\mathtt{Pres}(\mathtt{C_j}), \mathtt{Goals}(\mathtt{C_j}))$ as soon as the Jaccard index is greater than a threshold σ:
- $\mathtt{sim}(\mathtt{C_i}, \mathtt{C_j}) = \frac{|\mathtt{Pres}(\mathtt{C_i}) \cap \mathtt{Pres}(\mathtt{C_j})|}{|\mathtt{Pres}(\mathtt{C_i}) \cup \mathtt{Pres}(\mathtt{C_j})|}$
- $\mathtt{C_i} \simeq \mathtt{C_j}$ iff $\mathtt{sim}(\mathtt{C_i}, \mathtt{C_j}) \geq \sigma$
- Similarity can be computed using either FCA guided by similarity or pattern structures.

Similarity between cases is a tolerance relation

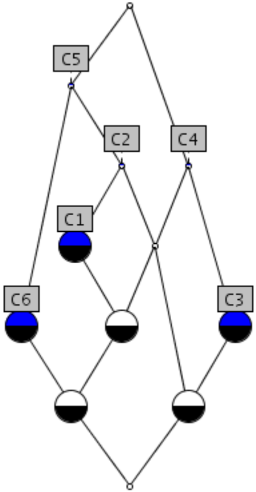
With $\sigma = 1/2$:

Cases	C1	C2	C3	C4	C5	C6
C1	x	x			x	
C2	x	x		x	x	
C3			x	x		
C4		x	x	x	x	
C5	x	x		x	x	x
C6					x	x

The "similarity concept lattice"

With $\sigma = 1/2$:

Cases	C1	C2	C3	C4	C5	C6
C1	x	x			x	
C2	x	x		x	x	
C3			x	x		
C4		x	x	x	x	
C5	x	x		x	x	x
C6					x	x



The corresponding point of view for adaptation

- A case $\mathtt{C_i} = (\mathtt{Pres}(\mathtt{C_i}), \mathtt{Goals}(\mathtt{C_i}))$ is adaptable into a case $\mathtt{C_j} = (\mathtt{Pres}(\mathtt{C_j}), \mathtt{Goals}(\mathtt{C_j}))$ as soon as the cardinal of the diffset of goals is no more than a threshold σ:
- $\mathtt{adapt}(\mathtt{C_i}, \mathtt{C_j}) = |\mathtt{Goals}(\mathtt{C_i}) \cup \mathtt{Goals}(\mathtt{C_j})| - |\mathtt{Goals}(\mathtt{C_i}) \cap \mathtt{Goals}(\mathtt{C_j})| \leq \sigma$
- $\mathtt{C_i}$ is adaptable into $\mathtt{C_j}$ iff $\mathtt{adapt}(\mathtt{C_i}, \mathtt{C_j}) \leq \sigma$
- Adaptability can be computed using either FCA guided by similarity or pattern structures.

Adaptability between cases

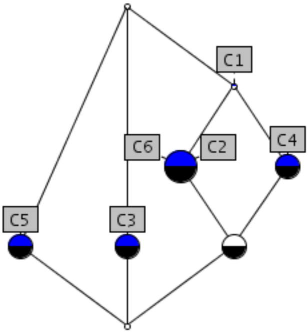
Adaptability is still a tolerance relation, i.e. reflexive and symmetric, but not (necessarily) transitive.
With $\sigma = 1$:

Cases	C1	C2	C3	C4	C5	C6
C1	x	x		x		x
C2	x	x				x
C3			x			
C4	x			x		
C5					x	
C6	x	x				x

The "adaptation concept lattice"

With $\sigma = 1$:

Cases	C1	C2	C3	C4	C5	C6
C1	x	x		x		x
C2	x	x				x
C3			x			
C4	x			x		
C5					x	
C6	x	x				x



Concluding remarks

- Organization of cases: FCA and extensions can be used for organizing a case base w.r.t. case content, i.e. problem and solution statements.
- Similarity and adaptation, and other complex and semantic relations between cases can be reified as paths in a concept lattice.
- Navigation in such a concept lattice can be used for problem-solving.
- Concept lattices are mathematically well-founded and provide efficient support to problem-solving.
- Alternatives knowledge discovery methods are available and can be used for still guiding and improving CBR problem-solving.
- Grazie per la vostra attenzione !

Many thanks to all members of Orpailleur team at LORIA and many other colleagues from France, Québec, Russia,...

- Permanent members: Adrien Coulet (MdC, UHP), Emmanuelle Deschamps (assistante), Marie-Dominique Devignes (CR CNRS), Nicolas Jay (MdC, UHP), Jean Lieber (MdC, UHP), Bernard Maigret (DR CNRS), Jean-François Mari (Professeur, Nancy 2), Amedeo Napoli (DR CNRS, responsable scientifique), Emmanuel Nauer (MdC, Metz), Chedy Raissi (CR INRIA), Dave Ritchie (DR INRIA), Malika Smaïl (MdC, UHP), Yannick Toussaint (CR INRIA).
- Associate permanent members: Florence Le Ber (Professeure, ENGEES Strasbourg), Frédéric Pennerath (MdC, Supelec Metz).
- PhD Students: Zainab Assaghir, Yasmine Assess, Sid-Ahmed Benabderrahmane, Israel Wakwoya Bayissa, Emmanuel Bresso, Julien Cojan, Valmi Dufour-Lucier, Anisha Ghoora, Mehdi Kaytoue, Thomas Meilender.
- Engineers and post-docs: Thoms Bourquard (p-d), Léo Ghemtio (p-d), Inaki Hernandez (e), Lazaros Mavridis (p-d), Violeta Perez-Nueno (p-d, MC Grant), Lian Shi (p-d), Vishwesh Venkatraman (p-d), Jean Villerd (p-d). ...
- plus roughly 3 PhDs, 2 engineers, and 1 post-doc coming soon...