Formal Concept Anlysis: Themes and Variations for Knowledge Discovery

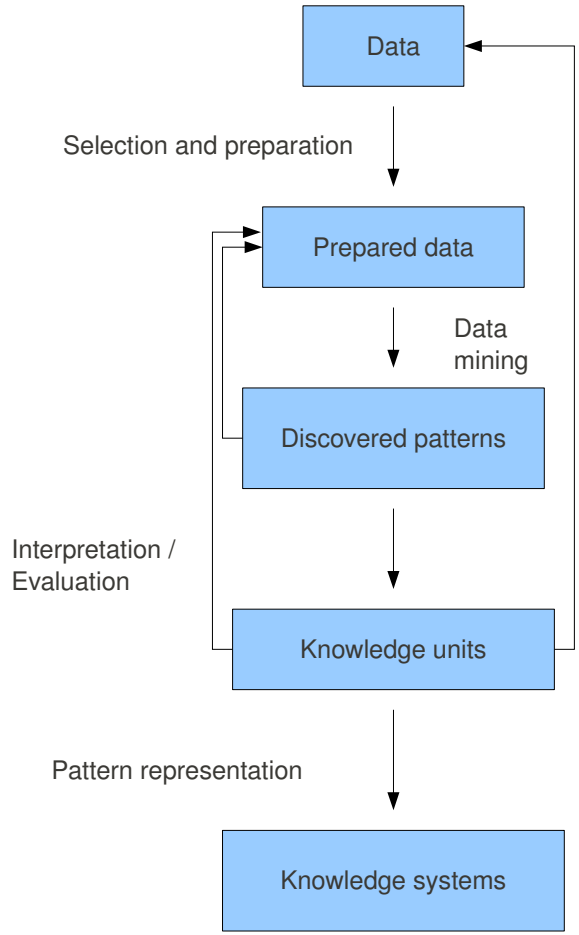
Amedeo Napoli
Orpailleur@LORIA
CNRS – INRIA Nancy Grand Est – Université de Lorraine
B.P. 239, 54506 Vandoeuvre les Nancy
Email: Amedeo.Napoli@loria.fr
http://orpailleur.loria.fr/

Séminaire L3i
La Rochelle, March 15, 2012

Summary of the presentation

Knowledge Discovery guided by Domain Knowledge (1)

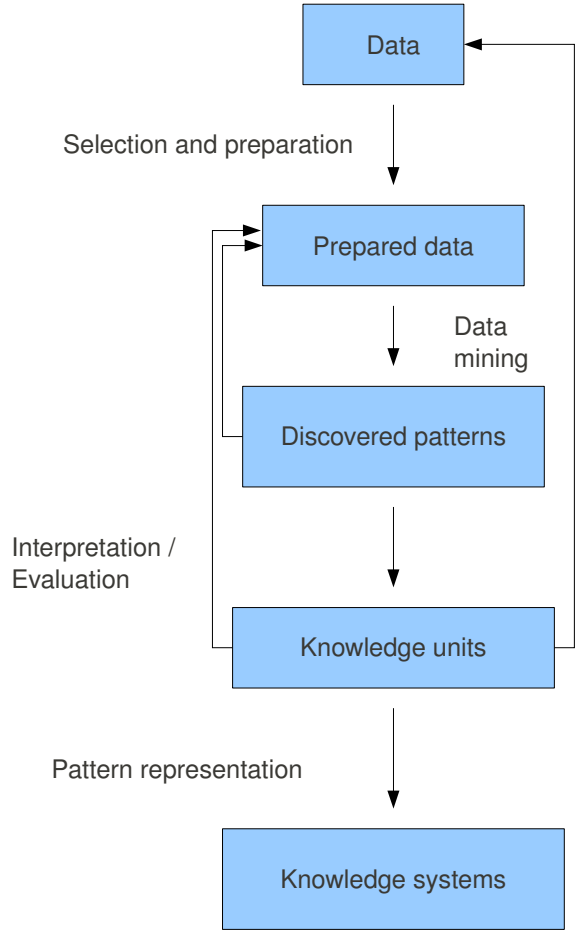
- The process of Knowledge Discovery guided by Domain Knowledge (KDDK) is applied on large volumes of data for extracting information units which are useful, significant, and reusable.
- KDDK is based on four main operations: data preparation, data mining, interpretation and representation of the extracted units.
- KDDK is iterative and interactive, guided by an analyst, and by domain knowledge.



KDDK is an interactive and iterative process that can be replayed.

Knowledge Discovery guided by Domain Knowledge (2)

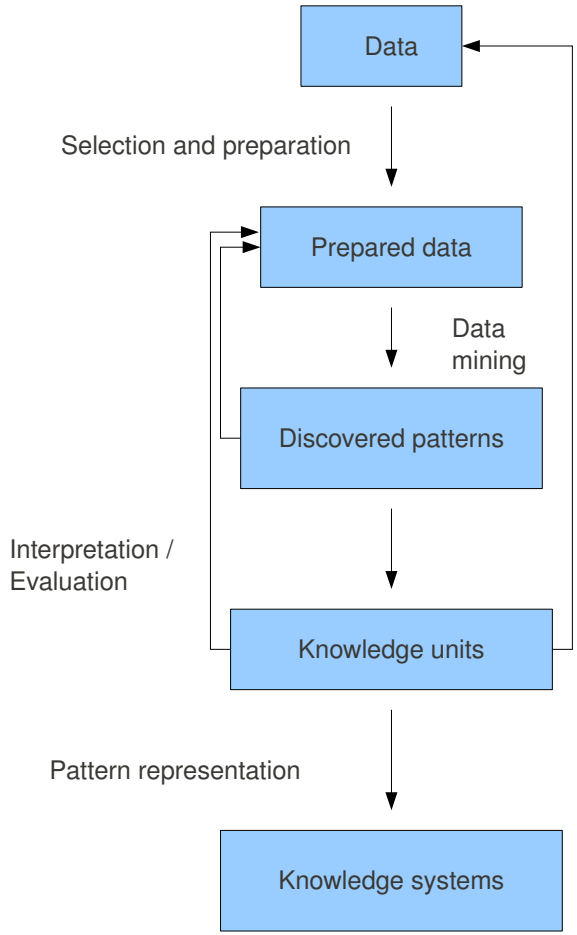
- One the core idea of KDDK is classification, which is involved in all tasks of data and knowledge processing:
- mining: Formal Concept Analysis (FCA), pattern mining, HMM...
- modeling: hierarchy of concepts and relations,
- representing: concepts and relations as knowledge units,
- reasoning and problem solving: classification-based and case-based reasoning.



KDDK is an interactive and iterative process that can be replayed.

Knowledge Discovery guided by Domain Knowledge (3)

- KDDK is used for knowledge engineering and problem-solving activities in some application domains:
  - agronomy
  - astronomy
  - biology
  - chemistry
  - cooking
  - medicine



Four research dimensions in Orpailleur

- Knowledge Discovery guided by Domain Knowledge (KDDK)
  $\longrightarrow$ Themes and variations in FCA: pattern structures, RCA, triadic concept analysis
  $\longrightarrow$ Pattern mining and association rule extraction.
  $\longrightarrow$ Text mining
- Knowledge systems and Semantic Web
  $\longrightarrow$ CBR and textual adaptation
- Implementing KDDK in Life sciences
  $\longrightarrow$ Data processing and knowledge mining
- Structural Systems Biology
  $\longrightarrow$ Docking, structural similarity and 3-D classification

Mining graphs and other complex data

- Graph Pattern Structures: generalization of interval pattern structures.
- Text mining on a graph-based representation of texts for allowing a deeper analysis of texts (e.g. multidimensional, linguistic, temporal).
- Retrieval, annotation and indexing of complex data (e.g. movie scenes, news) guided by domain knowledge and "environment" (subject to dynamic changes).

Software

- The Coron Platform: a KDDK toolkit for pattern and rule mining (`http://coron.loria.fr`).
- The Carottage system: a second-order HMM system for spatio-temporal classification (`http://www.loria.fr/~jfmari/App/`)
- Kasimir and CabamakA: decision support in oncology (`http://katexowl.loria.fr`).
- Taaable: CBR system for the cooking domain, with recipe adaptation (`http://taaable.fr`).
- BioRegistry repository: content metadata of biological resources (`http://bioregistry.loria.fr`).
- Hex: spherical polar Fourier docking program (`http://hex.loria.fr`) and HexServer, an interface to the GPU-powered Hex (`http://hexserver.loria.fr`).
- 3D-Blast: clustering and classifying protein folds (`http://threedblast.loria.fr`).

FCA: Themes

- A formal context is a triple (G, M, I) where G is a set of objects, M a set of attributes, and I a binary relation such as $(g, m) \in I$ means that "object g owns attribute m".
- A Galois connection characterizes formal concepts:

$$A' = \{m \in M \mid \forall g \in A \subseteq G : (g, m) \in I\}$$

$$B' = \{g \in G \mid \forall m \in B \subseteq M : (g, m) \in I\}$$

- (A, B) is a formal concept with extent $A = B'$ and intent $B = A'$, e.g. $(\{g_3, g_4, g_5\}, \{m_2, m_3\})$.

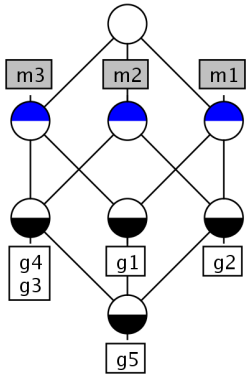
	m_1	m_2	m_3
g_1	×		×
g_2	×	×	
g_3		×	×
g_4		×	×
g_5	×	×	×

M. Barbut and B. Monjardet. Ordre et classification. Hachette, 1970.
B. Ganter and R. Wille. Formal Concept Analysis. Springer, 1999.

FCA: Themes

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \ (\Leftrightarrow B_2 \subseteq B_1)$$

$$(\{g_1, g_5\}, \{m_1, m_3\}) \leq (\{g_1, g_2, g_5\}, \{m_1\})$$



- A concept lattice is an ordered set of concepts with interesting properties:
- Concepts are pairs of maximal sets of objects and corresponding sets of attributes.
- The lattice provides a synthetic representation of data without loss of information and interpretation capabilities for knowledge discovery purposes.

FCA: first variation with Pattern structures

A pattern structure $(G, (D, \sqcap), \delta)$ is based on:

- a set G of objects
- a semi-lattice $(D, \sqcap)$ of descriptions or patterns
- a mapping δ associating an object g with its description $\delta(g) \in D$
- a Galois connection:

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

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

FCA: second variation with Triadic Concept Analysis (TCA)

- Given a numerical dataset (G, M, W, I), a bicluster is a pair (A, B) with $A \subseteq G$ and $B \subseteq M$.
- G a set of objects (rows)
- M a set of attributes (columns)
- W a set of values
- $I \subseteq G \times M \times W$ a relation s.t. $(g, m, w) \in I$, written $m(g) = w$, means that object g takes the value w for attribute m

	m_1	m_2	m_3	m_4	m_5
g_1	1	2	2	1	6
g_2	2	1	1	0	6
g_3	2	2	1	7	6
g_4	8	9	2	6	7

FCA: third variation with 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 relational model based on the entity-relationship model,
  - a conceptual scaling process allowing to represent relations between objects as relational attributes,
  - an iterative process for designing a concept lattice where concept intents include binary and relational attributes.
- The RCA process provides "relational structures" that can be represented as ontology concepts within a knowledge representation formalism such as description logics (DLs).

Pattern Structures

- Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli and Sébastien Duplessis. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis, Information Science, 181(10):1989–2001, 2011.
- Mehdi Kaytoue, Sergei O. Kuznetsov and Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis, in Proceedings of 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), Barcelona, Spain, 2011.
- Zainab Assaghir, Mehdi Kaytoue, and Amedeo Napoli and Henri Prade. Managing Information Fusion with Formal Concept Analysis, in Proceedings of 7th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2010), LNCS 6408, Springer, pages 104–115, 2010.

Handling numerical data with FCA?

Conceptual scaling (discretization or binarization)

An object has an attribute if its value lies in a predefined interval

	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

	$m_1, [4, 5]$	$m_2, [4, 7]$	$m_3, [5, 6]$
g_1	×	×	×
g_2			
g_3	×		×
g_4	×		
g_5	×		×

Different scalings: different interpretations of the data

General problem

How to directly build a concept lattice from numerical data?

How to handle complex descriptions

An intersection as a similarity operator

- $\cap$ behaves as *similarity operator*

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

- $\cap$ induces an ordering relation $\subseteq$

$$N \cap O = N \iff N \subseteq O$$
$$\{m_1\} \cap \{m_1, m_2\} = \{m_1\} \iff \{m_1\} \subseteq \{m_1, m_2\}$$

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

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

Pattern structure

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

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

A Galois connection

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

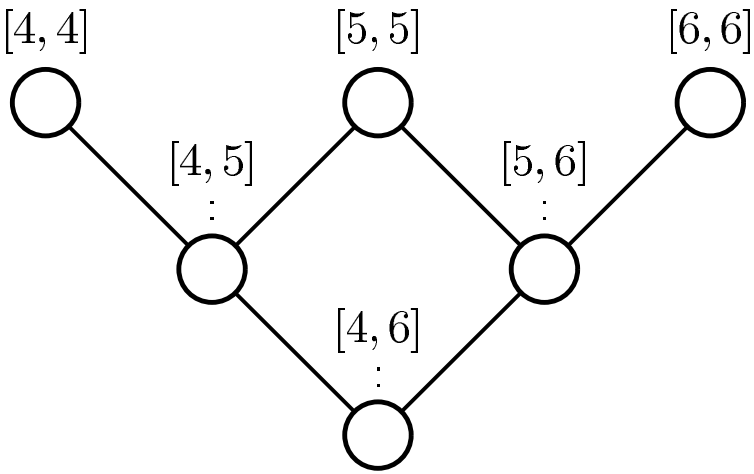
$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$$

Interval Pattern Structure

- A meet-semi-lattice for intervals $(\mathtt{D}, \sqcap)$ where $\mathtt{D}$ is a set of intervals,
- a possible choice for the meet operator is the convexification of intervals:

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

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



Interval Pattern Structure

- An interval pattern p is an n-dimensional vector of intervals: $\mathtt{p} = \langle [\mathtt{a_i}, \mathtt{b_i}] \rangle_{\mathtt{i} \in [1,\mathtt{n}]}$
- Operation $\sqcap$ and order of interval patterns:
  Given interval patterns $\mathtt{p} = \langle [\mathtt{a_i}, \mathtt{b_i}] \rangle_{\mathtt{i} \in [1,\mathtt{n}]}$ and $\mathtt{q} = \langle [\mathtt{c_i}, \mathtt{d_i}] \rangle_{\mathtt{i} \in [1,\mathtt{n}]}$:
- $$p \sqcap q = \langle [a_i, b_i] \rangle_{i \in [1,n]} \sqcap \langle [c_i, d_i] \rangle_{i \in [1,n]}$$
  $$p \sqcap q = \langle [a_i, b_i] \sqcap [c_i, d_i] \rangle_{i \in [1,n]}$$
- $$p \sqcap q = p \Leftrightarrow p \sqsubseteq q$$
  $$p \sqsubseteq q \Leftrightarrow [a_i, b_i] \sqsubseteq [c_i, d_i], \forall i \in [1, n]$$

Interval pattern structures based on convexification

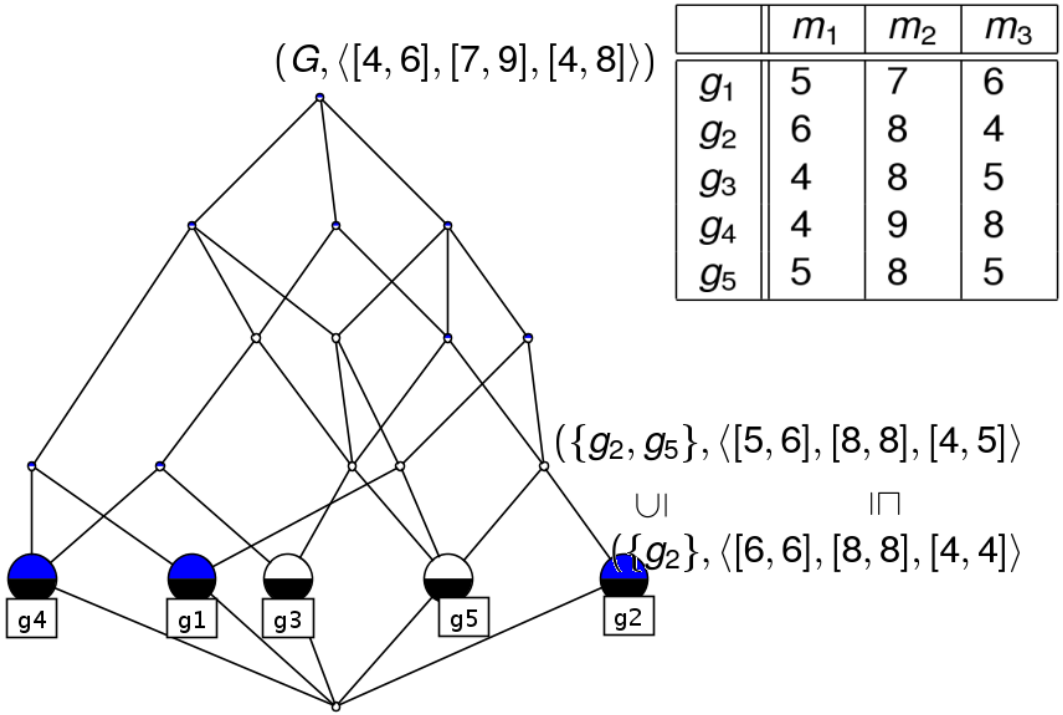
	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

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

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

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

Interval pattern concept lattice



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

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

Links with conceptual scaling

Interordinal scaling [Ganter & Wille]

- A scale to encode intervals of attribute values

	$m_1 \leq 4$	$m_1 \leq 5$	$m_1 \leq 6$	$m_1 \geq 4$	$m_1 \geq 5$	$m_1 \geq 6$
4	×	×	×	×		
5		×	×	×	×	
6			×	×	×	×

- Equivalent concept lattice
- Example

$$(\{g_1, g_2, g_5\}, \quad \{m_1 \leq 6, m_1 \geq 4, m_1 \geq 5, \ \dots\ , \ \dots\ \})$$
$$(\{g_1, g_2, g_5\}, \quad \langle [5, 6]\ , \ \dots\ , \ \dots\ \rangle)$$

Why should we use pattern structures as we have scaling?

Processing a pattern structure is more efficient

Interval pattern search space

Counting all possible interval patterns with interordinal scaling

$$\langle [a_{m_1}, b_{m_1}], [a_{m_2}, b_{m_2}], \ldots \rangle$$
where $a_{m_i}, b_{m_i} \in W_{m_i}$

	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

$$\prod_{i \in \{1,\ldots,|M|\}} \frac{|W_{m_i}| \times (|W_{m_i}| + 1)}{2}$$

360 possible interval patterns in our small example

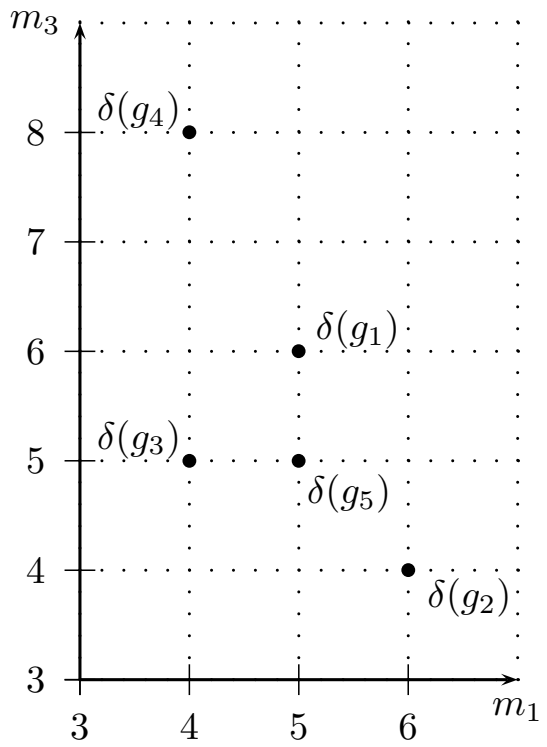
Questions on interval pattern mining

- What are the links between numerical pattern structures and pattern mining?
- How can we reuse (good) ideas from pattern mining, i.e. closed patterns, generators and equivalence classes, in the framework of pattern structures?

Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

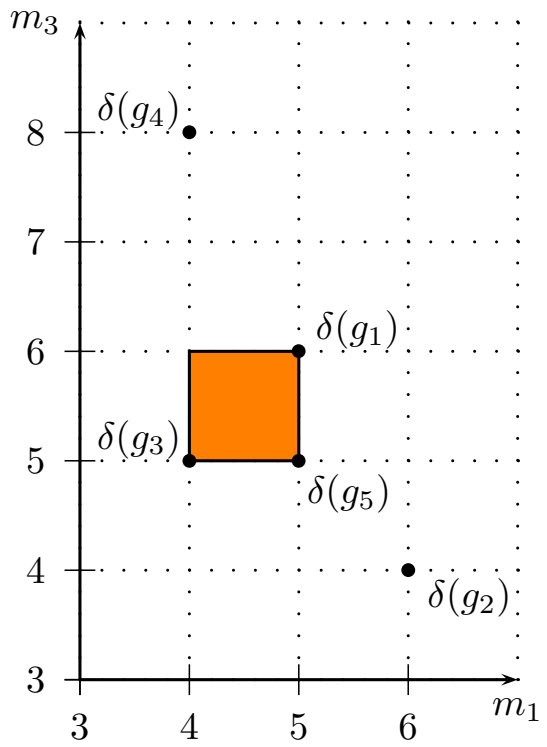


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$$\langle [4,5],[5,6] \rangle^{\square} = \{g_1, g_3, g_5\}$$

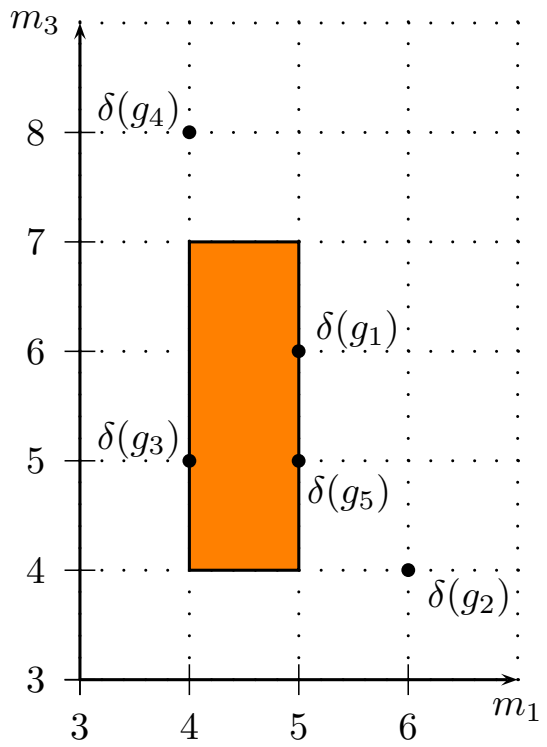


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$\langle [4,5],[5,6] \rangle^{\square} = \{g_1, g_3, g_5\}$
$\langle [4,5],[4,7] \rangle^{\square} = \{g_1, g_3, g_5\}$

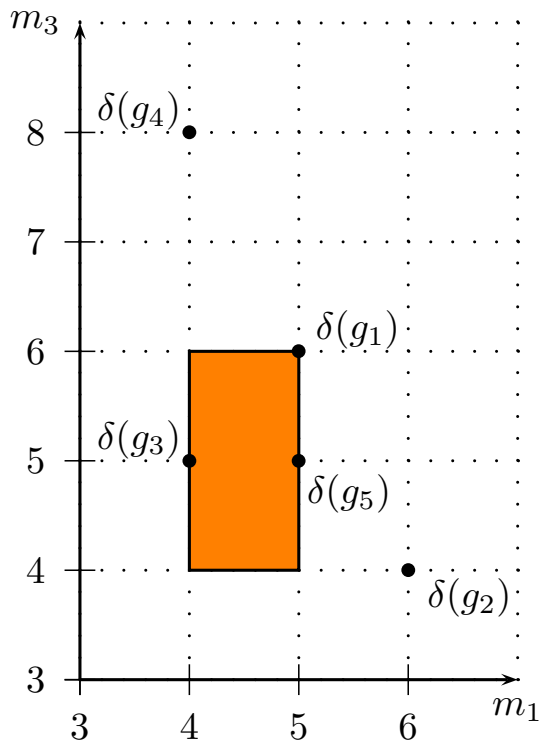


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$$\langle [4,5],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$

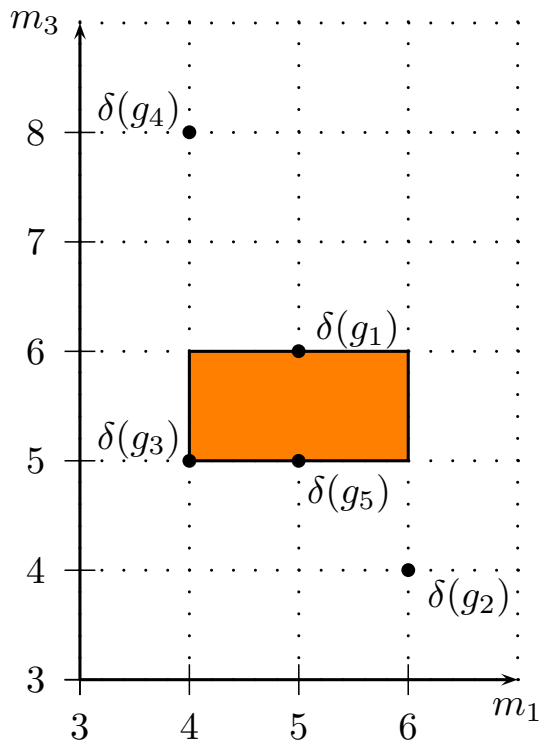


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$$\langle [4,5],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,6],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$

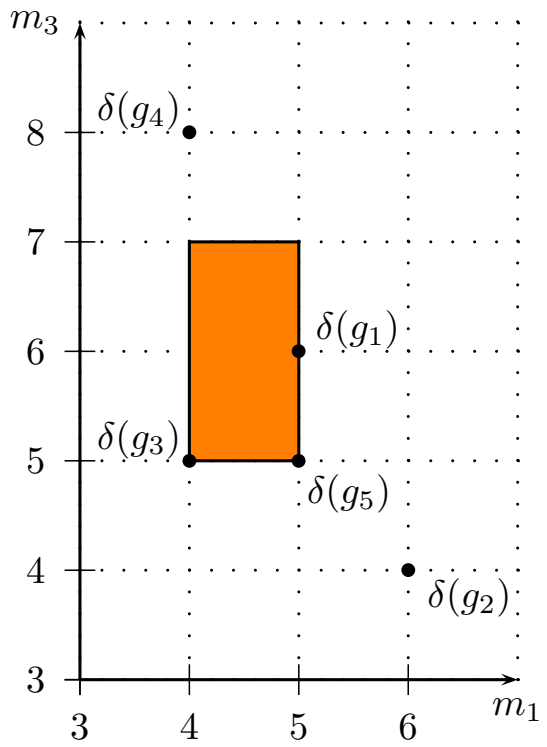


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$$\langle [4,5],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,6],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[5,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$

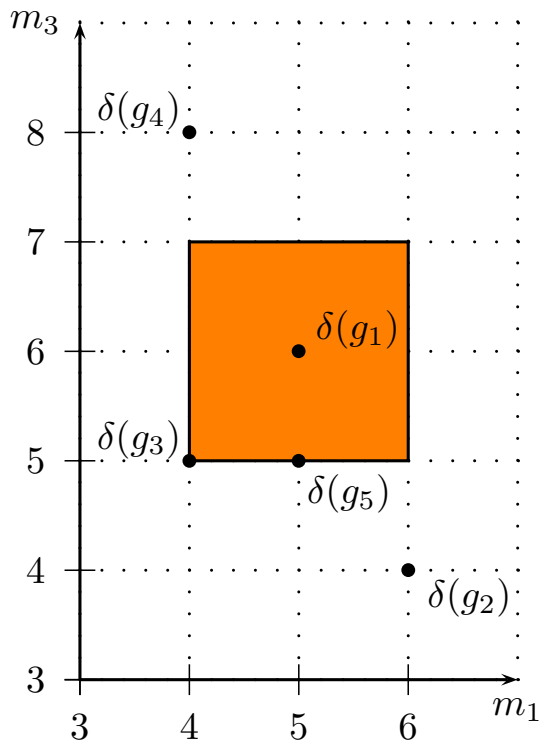


Semantics for interval patterns

Interval patterns as (hyper) rectangles

	m_1		m_3
g_1	5		6
g_2	6		4
g_3	4		5
g_4	4		8
g_5	5		5

$$\langle [4,5],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[4,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,6],[5,6]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,5],[5,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$
$$\langle [4,6],[5,7]\rangle^{\square} = \{g_1, g_3, g_5\}$$



A condensed representation

Equivalence classes of interval patterns

Two interval patterns with same image are said to be equivalent

$$c \cong d \iff c^{\square} = d^{\square}$$

Equivalence class of a pattern d

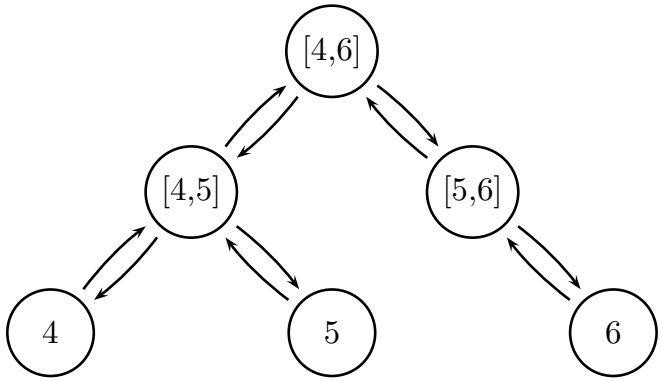
$$[d] = \{c | c \cong d\}$$

- with a unique closed pattern: the smallest rectangle
- and one or several generators: the largest rectangles

In the example: 360 patterns ; 18 closed patterns ; 44 generators

Algorithms & experiments

Algorithms: MintIntChange, MinIntChangeG[t|h]

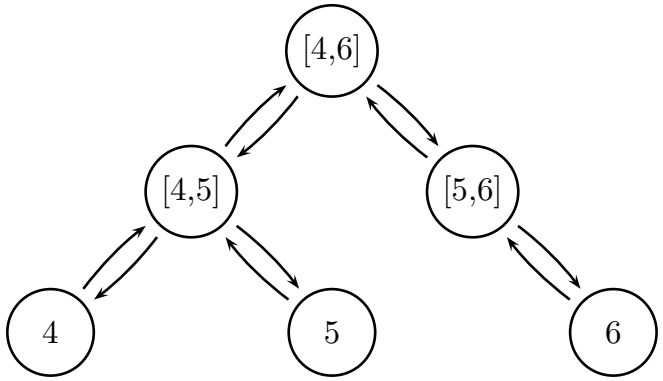


Principle with an example

1. Start from the most general interval pattern: $\langle [4,6], [7,9], [4,8] \rangle$
2. Apply next minimal change following a canonical order $c = \langle [4,5], [7,9], [4,8] \rangle$
3. Apply closure operator $c^{\square\square} = \langle [4,5], [7,9], [5,8] \rangle$
4. If canonicity test fails: backtrack (in the depth first traversal)
5. Otherwise go to 2. with $c^{\square\square} = \langle [4,5], [7,9], [5,8] \rangle$

Algorithms & experiments

Algorithms: MintIntChange, MinIntChangeG[t|h]



Experiments

- Mining several datasets from Bilkent University Repository
- Compression rate varies between 10^7 and 10^9
- Interordinal scaling
  - not efficient even with best algorithms (e.g. LCMv2)
  - redundancy problem discarding its use for generator extraction

Discussion

- Potential applications:
  - data privacy and k-anonymisation
  - k-box problem in computational geometry
  - quantitative association rule mining
  - data summarization
- Extension: focus on generator extraction
- Problems:
  - compression is not enough when considering very large data set
  - numerical data are noisy: this calls for fault-tolerant condensed representations

Triadic Concept Analysis and Biclustering

- Mehdi Kaytoue, Sergi O. Kuznetsov and Amedeo Napoli. Biclustering Numerical Data in Formal Concept Analysis, in Proceedings of 9th International Conference on Formal Concept Analysis (ICFCA 2011), LNCS 6628, Springer, pages 135–150, 2011.
- Mehdi Kaytoue, Sergei O. Kuznetsov, Juraj Macko, Wagner Meira and Amedeo Napoli. Mining Biclusters of Similar Values with Triadic Concept Analysis, in Proceedings of the Eighth International Conference on Concept Lattices and their Applications - CLA 2011, Amedeo Napoli and Vilem Vychodil editors, INRIA Nancy Grand Est - LORIA, pages 175-190, 2011.

Triadic FCA (TCA)

Triadic context

- (K_1, K_2, K_3, Y), where K_1, K_2, and K_3 are respectively called sets of objects, attributes, conditions, and $Y \subseteq K_1 \times K_2 \times K_3$.
- The fact $(a_1, a_2, a_3) \in Y$ is interpreted as the statement object a_1 has the attribute a_2 under condition a_3.

Triadic concept

- A triadic concept of (K_1, K_2, K_3, Y) is a triple (A_1, A_2, A_3) with $A_1 \subseteq K_1$, $A_2 \subseteq K_2$ and $A_3 \subseteq K_3$ satisfying the two following statements:
- (i) $A_1 \times A_2 \times A_3 \subseteq Y$, $X_1 \times X_2 \times X_3 \subseteq Y$
- (ii) $A_1 \subseteq X_1$, $A_2 \subseteq X_2$ and $A_3 \subseteq X_3$ implies $A_1 = X_1$, $A_2 = X_2$ and $A_3 = X_3$.

Triadic FCA (TCA)

- A triadic context $\mathbb{K} = (K_1, K_2, K_3, Y)$ gives rise to the three dyadic contexts:
  $\mathbb{K}^{(1)} = (K_1, K_2 \times K_3, Y^{(1)})$
  $\mathbb{K}^{(2)} = (K_2, K_1 \times K_3, Y^{(2)})$
  $\mathbb{K}^{(3)} = (K_3, K_1 \times K_2, Y^{(3)})$
  $gY^{(1)}(m, b) \Leftrightarrow mY^{(2)}(g, b) \Leftrightarrow bY^{(3)}(g, m) \Leftrightarrow (g, m, b) \in Y$
- A triadic concept of $\mathbb{K}$ is defined as a triple (A_1, A_2, A_3) where: $A_1 = (A_2 \times A_3)^{(1)} \subseteq K_1$ is the extent,
  $A_2 = (A_1 \times A_3)^{(2)} \subseteq K_2$ is the intent,
  $A_3 = (A_1 \times A_2)^{(3)} \subseteq K_3$ is the modus.

Triadic FCA (TCA)

Triconcept forming operators - outer closure

$$\Phi : X \to X^{(i)} : \{(a_j, a_k) \in K_j \times K_k \mid (a_i, a_j, a_k) \in Y \text{ for all } a_i \in X\}$$
$$\Phi' : Z \to Z^{(i)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in Z\}$$

Triconcept forming operators - inner (dyadic) closure

$$\Psi : X_i \to X_i^{(i,j,A_k)} : \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_i, a_k) \in X_i \times A_k\}$$
$$\Psi' : X_j \to X_j^{(i,j,A_k)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in X_j \times A_k\}$$

Existing algorithms for TCA

Trias for extracting frequent triadic concepts.

Data-Peeler for extracting frequent polyadic concepts

Biclustering of numerical data

- Given a numerical dataset (G, M, W, I), a bicluster is a pair (A, B) with $A \subseteq G$ and $B \subseteq M$.
- G a set of objects (rows)
- M a set of attributes (columns)
- W a set of values
- $I \subseteq G \times M \times W$ a relation s.t. $(g, m, w) \in I$, written $m(g) = w$, means that object g takes the value w for attribute m
- $(\{g_2, g_3, g_4\}, \{m_3, m_4\})$ is a bicluster.

	m_1	m_2	m_3	m_4	m_5
g_1	1	2	2	1	6
g_2	2	1	**1**	**0**	6
g_3	2	2	**1**	**7**	6
g_4	8	9	**2**	**6**	7

Biclustering of numerical data

A Bicluster should reflect

- a **local** phenomena in the data: "rectangles of values"
- **connectedness** of values: e.g. "similar values"
- **overlapping**: objects/attributes may belong to several biclusters
- a partial **order**, e.g. for algorithmic issues
- **maximality** of rectangles w.r.t. connectedness and ordering

Some types of biclusters

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

1.0	1.0	1.0	0.0
2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0
4.0	4.0	40	40

1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

Biclustering of numerical data

Applications

- Collaborative filtering and recommender systems
- Finding web communities
- Discovery of association rules in databases
- Gene expression analysis, ...

Algorithms

- Iterative Row and Column Clustering Combination
- Divide and Conquer / Distribution Parameter Identification
- Greedy Iterative Search / Exhaustive Bicluster Enumeration

S. C. Madeira and A. L. Oliveira Biclustering Algorithms for Biological Data Analysis: a survey. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2004.

Formal Concept Analysis

Can we use FCA for biclustering?

The interesting properties of concepts

- Maximality of concepts as rectangles
- Overlapping of concepts
- Specialization/generalisation hierarchy
- Synthetic representation of the data without loss of information

This is exactly what we need for biclustering!

An example of numerical biclusters

(A, B) is a numerical bicluster of equal values if:

$$m_i(g_j) = m_k(g_l), \forall g_j, g_l \in A, \forall m_i, m_k \in B$$

	m_1	m_2	m_3	m_4	m_5
g_1	1	2	2	1	**6**
g_2	2	1	1	0	**6**
g_3	2	2	1	7	**6**
g_4	8	9	2	6	7

The bicluster (A, B) is maximal if either:

- $(A \cup g, B)$, $g \in G \backslash A$ is not a bicluster of equal values
- $(A, B \cup m)$, $m \in M \backslash B$ is not a bicluster of equal values

A scale for biclusters of equal values

Nominal scaling restricted to each $w \in W$

$w \in W$	$\mathbb{K}_w$	$\mathfrak{B}_w$	Bicluster corresponding to first concept on left list
1	$\begin{array}{c\|ccccc} & m_1 & m_2 & m_3 & m_4 & m_5 \\ \hline g_1 & \times & & & \times & \\ g_2 & & \times & \times & & \\ g_3 & & & \times & & \\ g_4 & & & & & \end{array}$	$(\{g_2, g_3\}, \{m_3\})$ $(\{g_2\}, \{m_2, m_3\})$ $(\{g_1\}, \{m_1, m_4\})$	$\begin{array}{c\|ccccc} & m_1 & m_2 & m_3 & m_4 & m_5 \\ \hline g_1 & 1 & 2 & 2 & 1 & 6 \\ g_2 & 2 & 1 & \mathbf{1} & 0 & 6 \\ g_3 & 2 & 2 & \mathbf{1} & 7 & 6 \\ g_4 & 8 & 9 & 2 & 6 & 7 \end{array}$
6	$\begin{array}{c\|ccccc} & m_1 & m_2 & m_3 & m_4 & m_5 \\ \hline g_1 & & & & & \times \\ g_2 & & & & & \times \\ g_3 & & & & & \times \\ g_4 & & & & \times & \end{array}$	$(\{g_1, g_2, g_3\}, \{m_5\})$ $(\{g_4\}, \{m_4\})$	$\begin{array}{c\|ccccc} & m_1 & m_2 & m_3 & m_4 & m_5 \\ \hline g_1 & 1 & 2 & 2 & 1 & \mathbf{6} \\ g_2 & 2 & 1 & 1 & 0 & \mathbf{6} \\ g_3 & 2 & 2 & 1 & 7 & \mathbf{6} \\ g_4 & 8 & 9 & 2 & 6 & 7 \end{array}$
...	...	...	...

Biclusters of "numerical similar values"

A similarity relation on numerical values

$$w_1 \simeq_\theta w_2 \iff |w_1 - w_2| \leq \theta \text{ with } \theta \in \mathbb{R}, w_1, w_2 \in W$$

A rectangle (A, B) is a bicluster of similar values if:

$$m_i(g_j) \simeq_\theta m_k(g_l), \forall g_j, g_l \in A, \forall m_i, m_k \in B$$

	m_1	m_2	m_3	m_4	m_5
g_1	**1**	**2**	**2**	1	6
g_2	**2**	**1**	**1**	0	6
g_3	**2**	**2**	**1**	7	6
g_4	8	9	2	6	7

The bicluster (A, B) is maximal if no object/attribute can be added.

J. Besson, C. Robardet, L. De Raedt, J.-F. Boulicaut. Mining Bi-sets in Numerical Data, In *KDID 2006: 11-23*.

TCA - framework for biclustering

Biclustering in TCA

A dyadic context with objects and attributes
+
an interordinal scaling of numerical values in K_3
=
A scaled triadic context

Proposition for biclustering with TCA

(A_1, A_2, A_3) is a triadic concept
iff
(A_1, A_2) is a maximal bicluster of similar values for some $\theta \geq 0$.

TCA - framework for biclustering

The scale associated with interordinal scaling

J	$t_1 = [0,0]$	$t_2 = [0,1]$	$t_3 = [0,2]$	$t_4 = [0,6]$	$t_5 = [0,7]$	$t_6 = [0,8]$	$t_7 = [0,9]$	$t_8 = [1,9]$	$t_9 = [2,9]$	$t_{10} = [6,9]$	$t_{11} = [7,9]$	$t_{12} = [8,9]$	$t_{13} = [9,9]$
0	×	×	×	×	×	×	×						
1		×	×	×	×	×	×	×					
2			×	×	×	×	×	×	×				
6				×	×	×	×	×	×	×			
7					×	×	×	×	×	×	×		
8						×	×	×	×	×	×	×	
9							×	×	×	×	×	×	×

TCA - framework for biclustering

The triadic context with interordinal scaling

	$t_1 = [0,0]$					$t_2 = [0,1]$					$t_3 = [0,2]$					$t_4 = [0,6]$					$t_5 = [0,7]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1						×			×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_2				×			×	×	×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_3								×			×	×	×			×	×	×		×	×	×	×	×	×
g_4													×					×	×				×	×	×

	$t_6 = [0,8]$					$t_7 = [0,9]$					$t_8 = [1,9]$					$t_9 = [2,9]$					$t_{10} = [6,9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×		×					×
g_2	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×				×					×
g_3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×				×	×
g_4	×		×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×

	$t_{11} = [7,9]$					$t_{12} = [8,9]$					$t_{13} = [9,9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1															
g_2															
g_3				×											
g_4	×	×			×	×	×					×			

Scaling based on a tolerance relation

An alternative to interordinal scaling

- A tolerance relation $\simeq_\theta$ is reflexive, symmetric, but not transitive.
- Blocks of tolerance of a set of values W are defined as maximal sets of pairwise similar values.

$\simeq_1$	0	1	2	6	7	8	9
0	×	×					
1	×	×	×				
2		×	×				
6				×	×		
7				×	×	×	
8					×	×	×
9						×	×

Blocks of tolerance
$\{0,1\}$
$\{1,2\}$
$\{6,7\}$
$\{7,8\}$
$\{8,9\}$

Renamed classes
$[0,1]$
$[1,2]$
$[6,7]$
$[7,8]$
$[8,9]$

S. O. Kuznetsov Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research, in Formal Concept Analysis, Foundations and Applications, 2005.

Scaling based on a tolerance relation

- The scaling relation is a binary relation $J \subseteq W \times C$, where C is the set of blocks of tolerance over W renamed as their convex hulls. Then, $(w, c) \in J$ iff $w \in c$.
- Trimax is an algrithm based on triadic analysis for extracting maximal biclusters from a scaled triadic context.
- Let $Y \subseteq G \times M \times C$ be a ternary relation.
  Then $(g, m, c) \in Y$ iff $(m(g), c) \in J$, or simply $m(g) \in c$, where J is the scale relation.
  (G, M, C, Y) is called the TriMax triadic scaled context.

Scaling based on a tolerance relation

The triadic context scaled wrt a tolerance relation

	label 1					label 2					label 3					label 4					label 5				
	[0, 1]					[1, 2]					[6, 7]					[7, 8]					[8, 9]				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1	×			×		×	×	×	×						×										
g_2		×	×	×		×	×	×							×										
g_3			×			×	×	×						×	×				×						
g_4								×						×	×	×				×	×	×			

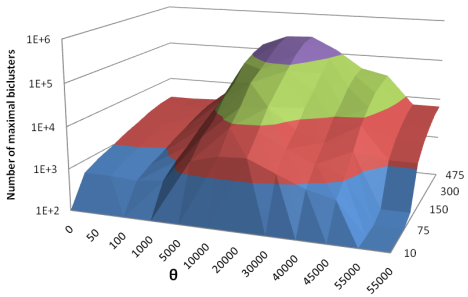
Trimax Algorithm

Trimax algorithm

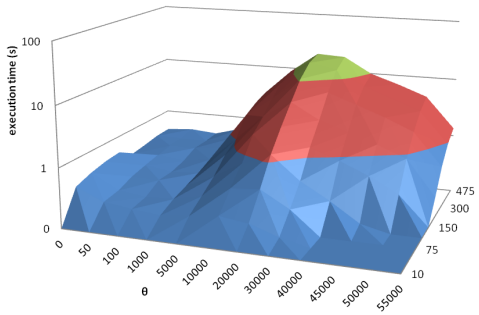
- Each dyadic context corresponds to a block of tolerance and there is no need to compute intersections of dyadic contexts.
- Each dyadic context is processed separately by a dyadic FCA algorithm.
- A dyadic concept in a dyadic context necessarily represents a bicluster of similar values, but we cannot be sure it is maximal.
- We need to check whether a concept is still a concept in other dyadic contexts (when overlapping) corresponding to other blocks of tolerance.

Experiments

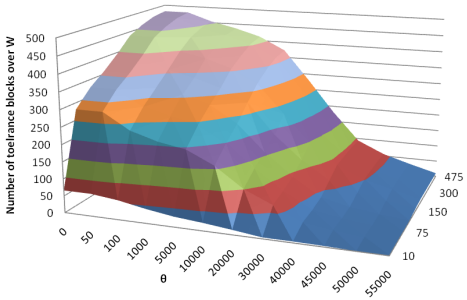
Trimax - experimental results



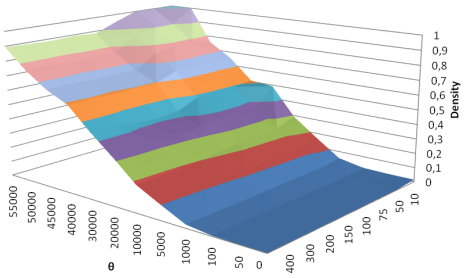
Nr. of max. biclusters



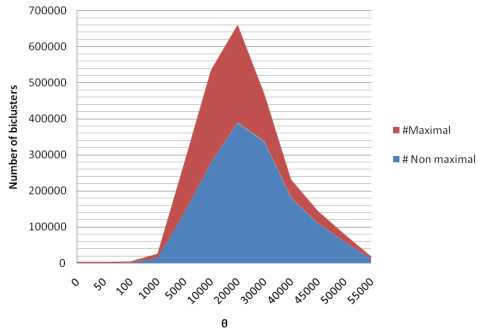
Execution times in sec.



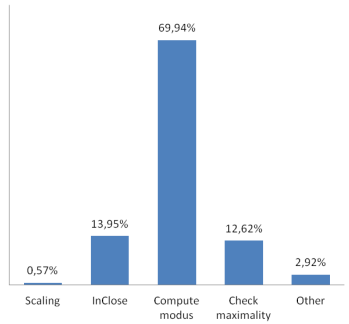
Nr. of blocks of toler.



Density of 3-adic cont.



Nr. generated of biclusters



Execution time

Experiments

Trimax - Elementary comparison

- *Numerical Biset Miner* (NBS-Miner) - not scalable
- *Interval Pattern Structures* (IPS) - less efficient than TriMax

Conclusion on Triadic Analysis

- Triadic FCA - a formal framework for biclustering, i.e. finding maximal biclusters with similar values.
- TriMax is efficient for computing maximal biclusters of similar values for a given similarity relation (with tolerance blocks).
- TriMax is a correct, complete and non-redundant algorithm.

Future research

- A deeper comparison of TriMax with other existing biclustering algorithms.
- A possible parallization of TriMax.

Conclusion

- FCA is a well-founded mathematical theory equipped with efficient algorithmic tools.
- FCA is a polymorphic process and addresses problems ranging from knowledge discovery to knowledge representation and reasoning, and pattern recognition as well.
- Times are there for various variations: pattern structures e.g. intervals and graphs, RCA, triadic analysis.
- There is room for many improvments and especially:
  $\longrightarrow$ in taking into account domain knowledge,
  $\longrightarrow$ dealing with documents (trees) and graphs,
  $\longrightarrow$ combining FCA with numerical processes (for data mining and reasoning).

The members of the team (March 2012)

- Permanent members: Adrien Coulet (MdC, UHP), Emmanuelle Deschamps (assistante), Marie-Dominique Devignes (CR CNRS), Nicolas Jay (MdC, UHP), Florence Le Ber (Professeure, ENGEES Strasbourg), Jean Lieber (MdC, UHP), Bernard Maigret (DR CNRS émérite), 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).
- PhD Students: Mehwish Alam, Emmanuel Bresso, Aleksey Buzmakov, Victor Codocedo, Sébastien da Silva, Valmi Dufour-Lussier, Elias Egho, Anisah Ghoorah, Thomas Meilender, Julien Stévenot, My Thao Tang.
- Engineers and post-docs: Yasmine Assess (p-d), Thomas Bourquard (p-d), Renaud Grisoni (e), Laura Infante-Blanco (e), Jean-François Kneib (e), Ioanna Lykourentzou (p-d), Felipe Melo (e), Violeta Perez-Nueno (p-d).
- Students (Masters and visitors: Quynh Do, Emmanuelle Gaillard, Laura Handojo, Ghania Khensous.