Embedding Tolerance Relations in Concept Lattices - An application in Information Fusion

Mehdi Kaytoue, Sergei O. Kuznetsov, Zainab Assaghir, Amedeo Napoli

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Embedding Tolerance Relations in Concept Lattices
An application in Information Fusion

Mehdi Kaytoue — Sergei O. Kuznetsov — Zainab Assaghir — Amedeo Napoli

N° 7353

August 2010

Knowledge and Data Representation and Management

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7353--FR+ENG

Embedding Tolerance Relations in Concept Lattices An application in Information Fusion

Mehdi Kaytoue, Sergei O. Kuznetsov, Zainab Assaghir, Amedeo Napoli

Theme : Knowledge and Data Representation and Management
Équipes-Projets Orpailleur

Rapport de recherche n° 7353 — August 2010 — 23 pages

Abstract: Formal Concept Analysis (FCA) is a well founded mathematical framework used for conceptual classification and knowledge management. Given a binary table describing a relation between objects and attributes, FCA consists in building a set of concepts organized by a subsumption relation within a concept lattice. Accordingly, FCA requires to transform complex data, e.g. numbers, intervals, graphs, into binary data leading to loss of information and poor interpretability of object classes. In this paper, we propose a pre-processing method producing binary data from complex data taking advantage of similarity between objects. As a result, the concept lattice is composed of classes being maximal sets of pairwise similar objects. This method is based on FCA and on a formalization of similarity as a tolerance relation (reflexive and symmetric). It applies to complex object descriptions and especially here to interval data. Moreover, it can be applied to any kind of structured data for which a similarity can be defined (sequences, graphs, etc.). Finally, an application highlights that the resulting concept lattice plays an important role in information fusion problem, as illustrated with a real-world example in agronomy.

Key-words: Formal concept analysis, similarity, tolerance, information fusion

Sergei O. Kuznetsov is a Professor at the State University Higher School of Economics (HSE) – Kirpichnaya 33/5 – 125219 Moscow, Russia

This research report is an extended version of the paper [11]. It includes supplementary materials and mathematical proofs.

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Introduire une relation de tolérance dans un treillis de concepts pour la fusion d'information

Résumé : L'analyse formelle de concepts (AFC) est un formalisme mathémati--que bien établi, utilisé pour la classification conceptuelle et l'organisation des connaissances. A partir d'une table binaire décrivant une relation entre des objets et leurs attributs, l'AFC permet de construire un ensemble de concepts organisés par une relation de subsomption, au sein d'un treillis de concepts. Mais l'AFC a besoin de transformer les données complexes, par exemple composées de nombres, intervalles, graphes, en données binaires. Cela peut aboutir à une perte d'information et une pauvre interprétabilité des classes d'objets. Dans ce papier, nous proposons une méthode de pré-traitement qui produit une table binaire à partir de données complexes tout en bénéficiant d'une similarité entre objets. De cette manière, le treillis de concepts est composée de classes representant des ensembles maximaux d'objets similaires deux à deux. Cette méthode est basée sur l'AFC et sur une formalisation de la similarité par une relation de tolerance (refléxive et symétrique). Cette méthode s'applique à des descriptions d'objets complexes et particulièrement dans ce papier, à des données intervalles. De plus, elle peut être appliquée à tout type de données structurées pour lesquelles une similarité peut être définie (séquences, graphes, etc.). Pour finir, une application argumente l'utilité d'un tel treillis pour des problèmes de fusion d'information, et s'illustre sur un exemple réel en agronomie.

Mots-clés : Analyse formelle de concepts, similarité, tolérance, fusion d'information

1 Introduction

Classification methods are currently used in artificial intelligence for various tasks, e.g. knowledge discovery, knowledge representation, and reasoning. One main objective of a good classification method is to work on complex data and to extract classes having a maximal internal cohesion and a maximal external separation with other classes. Moreover, classes have to be understandable and interpretable for being embedded in knowledge systems for problem solving purposes, e.g. decision support.

In this paper, we are interested in knowledge discovery and especially in using classification methods for analyzing complex real-world data, e.g., genes, agronomic and medical measures, weather forecast, etc. Such data, are most of the time described by objects whose attributes are valued by numbers, intervals, lists of symbols, or graphs. Data can be analyzed or mined using knowledge discovery methods among which Formal Context Analysis (FCA) [9]. FCA is a mathematically well founded classification framework allowing to derive implicit relationships from a set of objects and their attributes. The main structure which is built is a concept lattice, that can be represented by a diagram where classes of objects and ordering relations between classes can be drawn and interpreted.

FCA requires binary data but complex real-world data can be processed to some extent after a discretization of data. This transformation is based on a scaling process leading to the division of attributes and their ranges into a set of binary attributes, i.e. a scale. A scaling process implies arbitrary choices leading to a different concept lattice and thus to a different interpretation (no "universal scale" is existing). In addition, scaling is not always consistent with real-world knowledge: the same problem arises when working with crisp or fuzzy values in problem solving and decision support.

In the following, we propose a classification approach based on FCA that can be applied to real-world objects described by numerical attributes and taking into account similarity between attribute values. A scaling procedure is properly defined. In addition, the main characteristics of this approach are:

- Maximal classes of similar objects with maximal set of similar attributes are computed just as FCA concepts are based on maximal sets of objects sharing maximal sets of attributes.
- Classes are made of pairwise similar objects: two objects in a class are similar with respect to all attribute values shared by the two objects (and by every other object in the extent).
- The FCA machinery, i.e. algorithms and mathematical theory, can be reused with slight modifications.

In this paper, the mathematical formalization of similarity relies on a *tolerance relation* which is reflexive and symmetric. A tolerance relation can be used for building tolerance classes of similar objects, and an associated concept lattice. Tolerance classes are reused to properly define a scaling for initial numerical data allowing FCA to be applied.

Besides scaling, it is possible to directly process complex data using the so-called pattern structure approach. This extension of the basic FCA formalism

Table 1: A numerical dataset.

	m_1	m_2	m_3
g_1	6	0	$[1,2]$
g_2	8	4	$[2,5]$
g_3	11	8	$[4,5]$
g_4	16	8	$[6,9]$
g_5	17	12	$[7,10]$

Table 2: A formal context.

	$m_1 \geq 10$	$m_2 \leq 6$	$m_3 \leq 5$
g_1		×	×
g_2		×	×
g_3	×		×
g_4	×		
g_5	×		

is introduced here and applied to intervals. In addition, we show that tolerance relations can be embedded from complex data in pattern structures for deriving lattices with maximal classes of similarity.

Among the contributions of this paper, we show how to efficiently embed similarity within concept lattices by defining an appropriate scaling which makes the extent of each concept a maximal set of similar objects. Contrasting the large body of work on discretization for numerical classification methods [21], the present work is one of the few taking place in FCA and applicable to symbolic classification (a first attempt in this direction can be found in [18]). Accordingly, this paper introduces and explains the working relations between standard FCA, scaling, pattern structures, and similarity. This original classification framework is applied to complex data, i.e. intervals and numbers, in the domain of agronomy. The resulting lattice materializes information fusion and can be rather easily interpreted by agronomy experts for analyzing distributions of agricultural practices and agreement between information sources.

The paper is organized as follows. Section 2 presents preliminaries on FCA and scaling procedures. Section 3 introduces tolerance relations, the way how scales can be designed from numerical data, and how numerical concept lattices are built. Section 4 describes pattern structures in FCA and the use of tolerance relations in this framework. Section 5 describes a real-world experiment in agronomy based on this classification proposition. Finally, a discussion and a conclusion end the paper.

2 Formal concept analysis

2.1 Basics

Formal concept analysis starts with a formal context (G, M, I) where G denotes a set of objects, M a set of attributes, and $I \subseteq G \times M$ a binary relation between G and M. The statement $(g, m) \in I$ is interpreted as "the object g has attribute m" (see Table 2). The two derivation operators $(\cdot)'$ define a Galois connection between the powersets $(2^G, \subseteq)$ and $(2^M, \subseteq)$.

$$A' = \{m \in M \mid \forall g \in A : gIm\} \qquad for\ A \subseteq G,$$
$$B' = \{g \in G \mid \forall m \in B : gIm\} \qquad for\ B \subseteq M$$

For $A \subseteq G$, $B \subseteq M$, a pair (A, B), such that $A' = B$ and $B' = A$, is called a *(formal) concept*, e.g. $(\{g_3, g_4, g_5\}, \{m_1 \geq 10\})$. In (A, B), the set A is called the

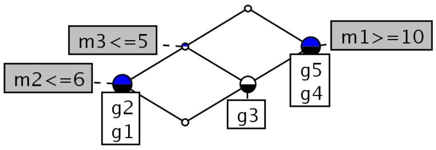

Figure 1: Concept lattice associated with Table 2.

extent and the set B the *intent* of the concept (A, B). Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ ($\Leftrightarrow B_2 \subseteq B_1$), e.g. the concept $(\{g_3\}, \{m_1 \geq 10, m_3 \leq 5\})$ is a sub-concept of $(\{g_3, g_4, g_5\}, \{m_1 \geq 10\})$. With respect to this partial order, the set of all formal concepts forms a complete lattice called the *concept lattice* of the formal context (G, M, I). Figure 1 shows the concept lattice[1] associated with Table 2. On the diagram, each node denotes a concept while a line denotes an order relation between two concepts. Due to *reduced labeling*, the extent of a concept has to be considered as composed of all objects lying in the extents of its sub-concepts. Dually, the intent of a concept is composed of all attributes in the intents of its super-concepts. The top concept ($\top$) is the highest and the bottom concept ($\perp$) is the lowest in the lattice.

The concept lattice provides a classification of objects in a domain. It entails both notions of maximality and generalization/specialization: (i) maximality as a concept corresponds to a maximal set of objects (extent) sharing a common maximal set of attributes (intent), and (ii) the generalization/specialization is given by the partial ordering of concepts. Concept lattices can be used in many application fields (life sciences, semantic-web, ...) for a number of purposes among which knowledge management (formalization, acquisition, extraction), data mining, information retrieval, and visualization, see e.g. [20, 19, 4].

2.2 Conceptual and interordinal scaling

Non binary data are described by a many-valued context (G, M, W, I), where W denotes a set of attribute values, such that $(g, m, w) \in I$, written $m(g) = w$, means that "attribute m takes value w for object g", e.g. the Table 1 is a many-valued context and $m_1(g_1) = 6$. In this example, W is a set of numbers (for m_1 and m_2) or intervals (for m_3).

For processing a context (G, M, W, I), a *conceptual scaling* is needed, where a scale for a given attribute is given by the transformation of attribute-value pairs into a set of binary attributes. For example, Table 1 can be transformed into Table 2 where the scale for m_1 is given by $\{m_1 \geq 10\}$ and the scale for m_2 by $\{m_2 \leq 6\}$. The choice of a scale is arbitrary and usually leads to loss of information (links and closeness between values) and border problems. For Table 3, $\{m_2 = 4\}$ and $\{m_2 = 8\}$ yield two different classes, but could also be considered as close values and classified together. Although scaling directly affects the size and interpretation of the resulting concept lattice, it remains an important technique for binarizing complex data [9].

Nominal scaling transforms a many-valued context (G, M, W, I) into a derived formal context (G, N, J) where each $n \in N$ is a pair (m, w) with $m \in M$

[1] In this paper, lattice diagrams are drawn with the ConExp software, `http://conexp.sourceforge.net/`.

Table 3: Two formal contexts obtained from Table 1, the first with nominal scaling and the second with interordinal scaling for attribute m_2.

	$m_2 = 0$	$m_2 = 4$	$m_2 = 8$	$m_2 = 12$	$m_2 \leq 0$	$m_2 \leq 4$	$m_2 \leq 8$	$m_2 \leq 12$	$m_2 \geq 0$	$m_2 \geq 4$	$m_2 \geq 8$	$m_2 \geq 12$
g_1	×				×	×	×	×	×			
g_2		×				×	×	×	×	×		
g_3			×				×	×	×	×	×	
g_4			×				×	×	×	×	×	
g_5				×				×	×	×	×	×

and $g(m) = w$ (also written "$m = w$") for some $g \in G$. For example, nominal scaling transforms Table 1 into Table 3 (left, where for readability, only attribute m_2 is kept). Knowing that $|N| = |W_m \times M|$ where W_m is the range of the attribute m, when $|W|$ is large, then N is also large, making the derived context harder to process [13].

Interordinal scale yields a formal context capturing for each attribute a set of intervals depending on the original values of the attribute. For example, Table 3 (right) shows the resulting binary context for attribute m_2. As it can be seen, the number of derived attributes grows very rapidly, i.e. $|N| = 2.|W_m \times M|$. Accordingly, the number of concepts of the corresponding lattice will also grow and lead to readability and interpretation problems.

In both scaling approaches and in scaling in general, arbitrary choices must be made, rarely depending on domain knowledge. There does not exist a "universal scale" and two different scales lead to different concept lattices and thus to different interpretations. In the following, we discuss the notion of similarity between objects having complex descriptions, and propose a scaling approach that associates to any object classes of similar objects using a formalization of similarity as a tolerance relation.

3 Formalizing similarity as a tolerance relation

3.1 Introduction and definitions

Similarity has been studied from many points of view in artificial intelligence and pattern recognition [17, 14]. For example, considering documents being described by their attributes, e.g. keywords, similarity of documents x and y can be defined by non-emptiness of the set of their common attributes, $x' \cap y' \neq \emptyset$. The similarity is reflexive and symmetric, but not necessarily transitive. Following this idea, a tolerance relation captures the characteristics of a similarity [12].

Definition 3.1 *For a set G, a binary relation $T \subseteq G \times G$ is called* tolerance *if:*

(i) $\forall x \in G \; xTx$ (reflexivity)
(ii) $\forall x, y \in G \; xTy \rightarrow yTx$ (symmetry)

Let us consider now a set of objects G, a tolerance relation T, and a formal context (G, G, T). First, some objects, say g_1 and g_2, are observed to be pairwise similar, i.e. $g_1 T g_2$. Then pairs of the tolerance relation lead to a class of similar objects or "class of similarity". Moreover, among the classes of similarity, some classes are maximal meaning that the class is not included in any larger class.

Definition 3.2 *Given a set G, a subset $K \subseteq G$, and a tolerance relation T on G, K is a* class of tolerance *if:*
(i) $\forall x, y \in K\ xTy$ (pairwise similarity)
(ii) $\forall z \notin K, \exists u \in K\ \neg(zTu)$ (maximality)
An arbitrary subset of a class of tolerance is a preclass.

Now, let us consider the classes of tolerance associated with the formal context (G, G, T). The class of tolerance of an object g has to be considered along two dimensions: *(i)* the class is defined as the set of all objects which are tolerant with g, *(ii)* the class is maximal in the sense that objects in the class are pairwise similar, and adding any other object in the class results in some pairs of non tolerant objects. A class of tolerance may be given a name which can be further used as an "attribute name" that describes the object. The result is a formal context (G, M, I) where I associates any object in G with its classes of tolerance $m \in M$.

Based on this observation, we show below how to use tolerance relations for designing scales for complex attributes and for building formal concepts whose extent are made of pairwise similar objects.

3.2 A tolerance relation for numerical data

Let us return to objects and numerical attributes of Table 1. Intuitively, two objects g_1 and g_2 are similar for a set of attributes if the values for each attribute are "similar". Similarity (or closeness) of two numerical values can be measured by the difference of these two values: $|m_1(g_1) - m_1(g_2)|$. Then, two numerical values are similar when their difference is lower than a *similarity threshold* θ expressing the maximal variation allowed between two similar values. More precisely, given two numbers $a, b \in \mathbb{R}$ and a similarity threshold θ, a similarity relation $\simeq_\theta$ is defined as:

$$a \simeq_\theta b \quad \Leftrightarrow \quad |a - b| \leq \theta \tag{1}$$

This similarity relation $\simeq_\theta$ is reflexive and symmetric but not necessarily transitive, i.e. $\simeq_\theta$ is a tolerance relation. For example, with $\theta = 2$, $a = 1$, $b = 3$ and $c = 5$, $a \simeq_\theta b$ and $b \simeq_\theta c$ but $a \not\simeq_\theta c$ ($1 \not\simeq_\theta 5$).

3.3 Classes of tolerance for numerical attributes

Let us consider a numerical many-valued context (G, M, W, I) where the range W_m of an attribute m is such that $W_m \subseteq W \subset \mathbb{R}$. Each attribute has a different range and different similarities and thresholds θ have to be defined. However, data can be normalized (e.g. by subtracting the mean, followed by dividing the standard deviation) and use a single threshold can be used for a given context.

Given an attribute $m \in M$, let us consider the formal context $(W_m, W_m, \simeq_\theta)$, and the relation $\simeq_\theta$ (Formula 1). Related objects in W_m are related are similar

m_1	6	8	11	16	17
6	×	×	×		
8	×	×	×		
11	×	×	×	×	
16			×	×	×
17				×	×

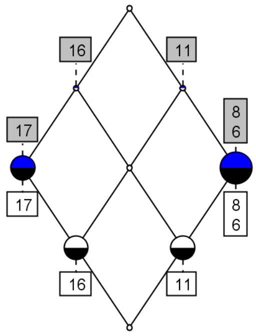

Figure 2: A tolerance relation formalized by a symmetric context (left) and its corresponding concept lattice (right).

w.r.t. $\simeq_\theta$. For example, given $\theta = 5$ and m_1 in Table 1, the formal context $(W_{m_1}, W_{m_1}, \simeq_5)$ can be read in Figure 2 (left). As $\simeq_5$ is symmetric and reflexive, so is $(W_{m_1}, W_{m_1}, \simeq_5)$ and it contains a diagonal of crosses. Furthermore, the associated concept lattice (see Figure 2 (right)) is also symmetric.

Proposition 3.1 *Given a context $(W_m, W_m, \simeq_\theta)$ and the associated lattice, any concept (A, B) is such that either $A \subset B$, $B \subset A$, or $A = B$. Then, for each concept (A, B), there exists a unique concept (B, A).*

Proof. In the context $(W_m, W_m, \simeq_\theta)$, the set of objects is the same as the set of attributes. Then, for a concept (A, B), either $A \subset B$, $B \subset A$, or $A = B$. Since both $A, B \in W_m$ and for any formal concept (A, B), $A' = B$ and $B' = A$. (B, A) is also a formal concept, as verifying $B' = A$ and $A' = B$.

For example, the upper right concept on Figure 2 (right) can be read as $(\{8, 6, 11, 16\}, \{11\})$ and has a corresponding concept $(\{11\}, \{8, 6, 11, 16\})$ lower still on the right. One consequence of the above proposition is that the concept lattice can be separated in two parts w.r.t. the mapping $(A, B) \mapsto (B, A)$. In [9], such a mapping is called a *polarity*, i.e. an order-reversing bijection inverse of itself, and the resulting concept lattice is a *polarity lattice*. Then, we have the notion of axis of polarity:

Definition 3.3 (Axis of polarity) *In a polarity lattice, the set of all concepts (A, B) such that $A = B$ forms an* axis of polarity *of the concept lattice.*

For example, the set of concepts $\{(\{16, 17\}, \{16, 17\}), (\{11, 16\}, \{11, 16\}), (\{6, 8, 11\}, \{6, 8, 11\})\}$ is the axis of polarity of the concept lattice on Figure 2 (right). The set of all concepts (C, D) such that $(A, B) \leq (C, D)$, denoted by U, forms the upper part of the concept lattice. Dually, the set of all concepts (E, F) such that $(E, F) \leq (A, B)$, denoted by L, forms the lower part of the concept lattice. If $(A, B) \in U$ then $(B, A) \in L$ and $B \subset A$. Dually, if $(A, B) \in L$ then $(B, A) \in U$ and $A \subset B$.

Let us now consider the concept $(\{16, 17\}, \{16, 17\})$ of the axis of polarity in the lattice on Figure 2 (right). The values in $\{16, 17\}$ are all similar w.r.t. $\simeq_5$ and $\{16, 17\}$ cannot be extended with any other value without violating the internal similarity, i.e. there does not exist any element that does not belongs to $\{16, 17\}$ and that is similar with all elements in $\{16, 17\}$. This is true for all concepts in the axis of polarity.

This means that the extent or intents of the concepts in the axis of polarity are tolerance classes. Let us now consider the upper left concept $(\{11, 16, 17\}, \{16\})$ in the lattice on Figure 2 (right). This concept is in U and the values in the extent $\{11, 16, 17\}$ are similar to 16. Moreover, the intent $\{16\}$ is contained in the larger intent $\{16, 17\}$ meaning that $\{16\}$ determines a preclass of tolerance. Dually, we have the same interpretation for the symmetric concept $(\{16\}, \{11, 16, 17\}) \in L$.

Proposition 3.2 *Let (A, B) be a concept of the axis of polarity, i.e. $A = B$. Then, A (or B) is a set of maximal pairwise similar values, i.e. A determines a class of tolerance. Let (C, D) a concept in U but not in the axis of polarity, i.e. $D \subset C$. D is a preclass of tolerance and C is the set of all values similar to values in D.*

Proof. Both derivation operators $(\cdot)'$ have same domain and range W_m, and $(\cdot)'$ associates with a subset A of values in W_m the maximal subset of similar values in W_m, i.e. related through $\simeq_\theta$. Then, for a concept (A, B) where $A = B$ and $A' = B$ or $A = B'$, then $A = A'$ or $B = B'$ are maximal and define a same tolerance class. Moreover, the set of all extents A or all intents B from concepts of the axis of polarity covers the set W_m. For a concept (C, D) with $D \subset C$, since $C' = D$, all values in C are similar to values in D. Now, relying on the preceding proposition, as the concept (C, D) does not verify $C = D$ but instead $D \subset C$, it exists a class of tolerance say F such as $D \subset F \subset C$ and thus D is a preclass of tolerance.

The intents of the concepts in the upper part of the lattice –or dually the extents in the lower part– are partially ordered and determine sets of similar values. Among these intents, the intents in the axis of polarity are maximal and are classes of tolerance, and the other intents are only preclasses of tolerance. For example, taking $\theta = 5$ and m_1 in Table 1, there are 5 intents, namely $\{16\}$, $\{11\}$, $\{16, 17\}$, $\{11, 16\}$, and $\{6, 8, 11\}$, where the three last intents are tolerance classes. When there is no ambiguity, we use the term of "class of similarity" for a class or a preclass of tolerance.

These classes of similarity are used to define a scale allowing the application of FCA algorithms to a numerical many-valued context. Classical FCA algorithms can be used to compute classes of similarity and require slight modifications for generating the upper (dually lower) part of the concept lattice only (discussed later).

3.4 From numbers to intervals

The preceding approach is generalized to intervals by defining a similarity $\simeq_\theta$ as follows. Given $a, b, c, d \in \mathbb{R}$ and a threshold $\theta \in \mathbb{R}$:

$$[a, b] \simeq_\theta [c, d] \quad \Leftrightarrow \quad max(b, d) - min(a, c) \leq \theta \tag{2}$$

provided that $|a - b| \leq \theta$ and that $|c - d| \leq \theta$. Then, two numerical intervals are similar if the length of their "convex hull" is not larger than a threshold θ. The similarity $\simeq_\theta$ on intervals is a reflexive and symmetric relation and is

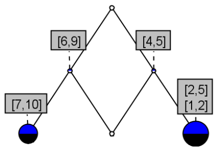

Figure 3: Partial order of classes of similarity.

therefore a tolerance relation. The framework developed for numbers is adapted to intervals for computing classes of similarity. For example, the similarity $\simeq_5$ for attribute m_3 in Table 1 yields the partial concept lattice on Figure 3. This lattice is equivalent for m_3 to the lattice given on Figure 2 for m_1. Classes of similarity can be read with reduced labelling.

The definition of similarity in Formula 2 is discussed now. There are three main choices that can be envisioned: union of intervals, intersection of intervals, and convexification. These three ways of considering similarity of intervals can be interpreted in terms of *consensus*, i.e. a general agreement between intervals and thus between attributes associated with these intervals. Union reflects a minimal consensus as it takes all values of the class of similarity. Intersection returns a maximal agreement but the constraint may be too strong, as intersection may be empty. The convexification of a set of intervals returns the smallest interval containing all its arguments (convex hull) and is used here in formula 2. Convexification can be seen as equivalent to union of intervals when filling the holes between intervals. Then convexification shows the larger consensus as it gives an agreement between all original intervals and the holes between these values. Thus, the choice of convexification allows a greater flexibility and probably offers the best consensus. Accordingly, the convex hull of intervals was used with positive results in [10] for classifying objects with attributes valued by intervals FCA. We go back to the choice of convexification in Section 5, where an application based on an information fusion problem is detailed.

3.5 Building a "numerical concept lattice"

At present, we have made precise how a partially ordered set of classes of similarity can be built from attributes valued by numbers or intervals of numbers in a many-valued context. Now, classes of similarity have to be named before being used as attribute names for scaling the original many-valued context and derive a scaled binary context from which the final concept lattice is built. Actually, the name of the elements of the scale can be related to the content of the corresponding class of similarity and to the name of the original attribute that is scaled. In the present case, an element of the scale is named by a pair associating the name of the original attribute and either the content of the class of similarity in case of numbers, e.g. $\{16, 17\}$ for m_1, or the convex hull in case of intervals, e.g. $[7, 10]$ for m_3.

Let us consider the numerical many-valued context (G, W, M, I) in Table 1. Three sets of classes of similarity, one for each attribute m_1, m_2, and m_3, are computed thanks to three tolerance relations relying on three different similarities $\simeq_\theta$, and extracted from the symmetric concept lattices associated with each tolerance relation. The transformation of the original (G, W, M, I) context into the derived (G, N, J) reads as follows:

Table 4: A formal context obtained from Table 1 handling classes of tolerance of attributes m_1 and m_2.

	$(m_1, 11)$	$(m_1, 16)$	$(m_1, [6, 11])$	$(m_1, [11, 16])$	$(m_1, [16, 17])$	$(m_2, 4)$	$(m_2, 8)$	$(m_2, [0, 4])$	$(m_2, [4, 8])$	$(m_2, [8, 12])$
g_1			×					×		
g_2			×			×		×	×	
g_3	×		×	×			×		×	×
g_4		×		×	×		×		×	×
g_5					×					×

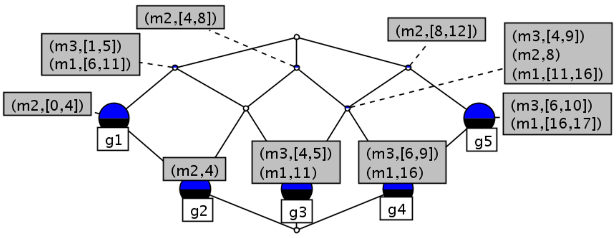

Figure 4: Concept lattice raised from Table 4.

- G is the set of original objects,
- $N = \bigcup_{m \in M}(\{m\} \times C_m)$ with C_m is the set of all classes of similarity of attribute m,
- $(g, (m, C_m)) \in J$ means that the value of object g in the many valued context, i.e. $m(g)$, belongs to class C_m,

For example, the derived binary context associated with Table 1 is given in Table 4 for attributes m_1 and m_2 where the thresholds are $\theta = 5$ for m_1 and $\theta = 4$ for m_2 (and $\theta = 5$ for m_3). Figure 4 shows the resulting concept lattice.

4 Embedding tolerance relations in pattern structures

The preceding formalization of similarity based on a tolerance relation allowed us to design concept lattices whose concept extent are composed of pairwise similar objects. This work takes place in standard FCA and applies to binary formal contexts. One first extension consists in applying this work directly on complex data, i.e. formal contexts where attribute values can be symbolic, numbers, intervals, graphs, etc.

For this purpose, a *pattern structure* is defined as a generalization of a formal context including complex data [8]. First a similarity operation is defined on object descriptions, allowing to organize these descriptions within a partial ordering. Then, as in FCA, a Galois connection between two ordered sets (objects and descriptions) gives rise to a concept lattice.

For summarizing, two important outputs can be considered. Firstly, the use of pattern structures allows to avoid binarization and to keep the same data formalism during the analysis, known to be a critical issue in knowledge representation and reasoning systems with concept lattices [8, 1]. Secondly, the introduction of a tolerance relation in pattern structures allows to obtain concept extents made of pairwise similar objects and obtained from directly from complex data.

4.1 Pattern structures

Formally, let G be a set of objects, let $(D, \sqcap)$ be a meet-semi-lattice of potential object descriptions and let $\delta : G \longrightarrow D$ be a mapping associating an object with its description. Then $(G, (D, \sqcap), \delta)$ is called a *pattern structure*. Elements of D are called *patterns* and are ordered by the subsumption relation $\sqsubseteq$: given $c, d \in D$, $c \sqsubseteq d \Longleftrightarrow c \sqcap d = c$. A pattern structure $(G, (D, \sqcap), \delta)$ gives rise to the following derivation operators $(\cdot)^{\square}$, given $A \subseteq G$ and $d \in (D, \sqcap)$:

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

These operators form a Galois connection between $(2^G, \subseteq)$ and $(D, \sqsubseteq)$. *(Pattern) concepts* of $(G, (D, \sqcap), \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in (D, \sqcap)$, such that $A^{\square} = d$ and $A = d^{\square}$. For a pattern concept (A, d), d is called a *pattern intent* and is a common description of all objects in A, called *pattern extent*. When partially ordered by $(A_1, d_1) \leq (A_2, d_2) \Leftrightarrow A_1 \subseteq A_2 \;\; (\Leftrightarrow d_2 \sqsubseteq d_1)$, the set of all concepts forms a complete lattice called a *(pattern) concept lattice*.

Since pattern structures are defined in full compliance with FCA, i.e. based on a Galois connection between two ordered sets, many FCA algorithms (detailed in [13]) can be used to compute the pattern concept lattice.

4.2 Interval pattern structures

Now we show how to build a concept lattice from numerical data without discretization. We instantiate the general definition of pattern structures for numerical data, called *interval pattern structures* and introduced in [10].

Considering only one numerical attribute. Numbers and intervals of numbers are patterns: they may be ordered within a meet-semi-lattice making them potential object descriptions. A possibility is to define the meet $\sqcap$ of two intervals $[a_1, b_1]$ and $[a_2, b_2]$, with $a_1, b_1, a_2, b_2 \in \mathbb{R}$ as a convexification operator: $[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$, i.e. the smallest interval containing them. Indeed, when c and d are intervals, $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$ holds: $[a_1, b_1] \sqsubseteq [a_2, b_2] \Leftrightarrow [a_1, b_1] \supseteq [a_2, b_2]$.

Figure 5 gives an example of meet-semi-lattice of intervals. The interval labelling a node is the meet of all intervals labelling its ascending nodes, e.g. $[4, 8] = [4, 4] \sqcap [8, 8]$, and is also subsumed by these intervals, e.g. $[4, 8] \sqsubseteq [4, 4]$.

Note that several meet operations can be chosen, e.g. intersection, union, minimum, maximum, etc., inducing partial order of intervals (more details

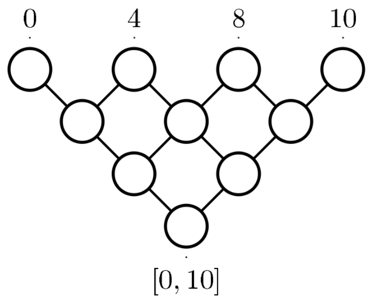

Figure 5: A meet-semi-lattice of intervals. The meet operation consists in a convexification.

in [7]). We use here convexification for being in accordance with the previous section.

Considering a numerical dataset. A numerical dataset is represented by a many-valued context (G, M, W, I). Objects are described by several numbers or intervals, each one standing for a given attribute, and hence *interval vectors* are introduced as patterns. When c and d are interval vectors, we write $c = \langle [a_i, b_i] \rangle_{i \in [1,|M|]}$ and $d = \langle [c_i, d_i] \rangle_{i \in [1,|M|]}$. Interval vectors may be partially ordered within a meet-semi-lattice as follows. Given two interval vectors $c = \langle [a_i, b_i] \rangle_{i \in \{1,\dots,|M|\}}$, and $d = \langle [c_i, d_i] \rangle_{i \in \{1,\dots,|M|\}}$,

$$c \sqcap d = \langle [min(a_i, c_i), max(b_i, d_i)] \rangle_{i \in \{1,\dots,|M|\}}$$

meaning that a convexification of intervals on each vector dimension is operated. The meet operator induces the following subsumption relation $\sqsubseteq$ on interval patterns

$$\langle [a_i, b_i] \rangle \sqsubseteq \langle [c_i, d_i] \rangle \Leftrightarrow [a_i, b_i] \supseteq [c_i, d_i],\ \forall i \in \{1, ..., |M|\}.$$

In this way, a numerical dataset is a pattern structure. In Table 1, description of g_1 is $\delta(g_1) = \langle [6, 6], [0, 0], [1, 2] \rangle$. We have $\delta(g_1) \sqcap \delta(g_2) = \langle [6, 8], [0, 4], [1, 5] \rangle$, and therefore $\langle [6, 8], [0, 4], [1, 5] \rangle \sqsubseteq \langle [6, 6], [0, 0], [1, 2] \rangle$. The Galois connection is illustrated as follows. $\{g_1, g_3\}^\square = \langle [6, 11], [0, 8], [1, 5] \rangle$ and $\langle [6, 11], [0, 8], [1, 5] \rangle^\square = \{g_1, g_2, g_3\}$, making the pair $(\{g_1, g_2, g_3\}, \langle [6, 11], [0, 8], [1, 5] \rangle)$ a pattern concept.

Actually, it was proved that the resulting pattern concept lattice is isomorphic to the one obtained after interordinal scaling in classical FCA [10]. However, pattern structures avoid binarization and are shown to be much more efficient to process [10]. Nevertheless, the problem of combinatorial explosion of number of concepts outlined in Section 2.2 still holds. This is due to the convexification that allows to build too general concept intents. For example, the top concept intent is always composed of intervals of maximal size, i.e. the whole domain of each attribute. It seems then quite natural that the convexifiation should be controlled, i.e. two intervals or numbers can be convexified iff their difference is not "too large". Interestingly, this involves a notion of similarity between complex descriptions as related before. In fact, we show in the following that a tolerance relation can also be directly embedded in a pattern structure and its pattern concept lattice, thus avoiding binarization and controlling convexification as in the previous section.

4.3 Tolerance relation in pattern structures

Basically, pattern structures consider the meet operator $\sqcap$ as a similarity operator [8]. Given two objects g and h, and their respective descriptions $d = \delta(g)$ and $e = \delta(h)$ from a meet-semi-lattice, $d \sqcap e$ gives a description representing similarity between g and h. As a meet-semi-lattice is defined on the existence of a meet for any pair of elements, it follows that any two objects are similar and that their "level" of similarity depends on the level of their meet in the semi-lattice. However, such symbolic notion of similarity is at the heart of lattice-based classification for its ability to distinguish if one description is more general than another.

By contrast, numerical notion of similarity allows to quantify (measure, or approximate) with a continuous value the proximity between two descriptions, for a large spectrum of complex data, see e.g. [15]. However, pairs of descriptions having the same similarity value may correspond to radically different generalization levels. For example, with $\theta = 2$, we have $[30, 30] \simeq_2 [30, 32]$ and $[30, 32] \sqsubseteq [30, 30]$ while $[2, 2] \simeq_2 [4, 4]$ and neither $[2, 2] \sqsubseteq [4, 4]$ nor $[4, 4] \sqsubseteq [2, 2]$.

Now, we are interested in introducing in a pattern concept lattice a notion of similarity relation $\simeq_\theta$, combining calculability and flexibility of the numerical aspect, and the intelligibility of the symbolic aspect. Tolerance relations allow this combination, since they are defined by a binary relation encoding a similarity.

Going back to numerical data, a similarity relation $\simeq_\theta$ was introduced for numbers in Formula 1 and for intervals in Formula 2. Then, we state that two descriptions are similar or not as follows. Given a,b,c,$d \in \mathbb{R}$, a parameter $\theta \in \mathbb{R}$, and an additional pattern denoted by $*$ representing "non similarity",

$$[a, b] \sqcap_\theta [c, d] = \begin{cases} [min(a, c), max(b, d)] \text{ if } [a, b] \simeq_\theta [c, d] \\ * \text{ otherwise,} \end{cases}$$

and

$$* \sqcap_\theta [a, b] \quad = \quad * \Leftrightarrow \; * \sqsubseteq_\theta [a, b].$$

More generally, given $x, y \in D$ two patterns, then x and y are said to be similar iff $x \simeq_\theta y \iff x \sqcap_\theta y \neq *$ where $*$ materializes the pattern that is subsumed by any other pattern. This pattern is added in D and can be interpreted as the pattern denoting "non similarity" between two patterns. In this way, the convexification is controlled and it is not possible to have intervals whose length exceeds θ.

This means that though each interval from a semi-lattice $(D, \sqcap_\theta)$ describes a preclass of tolerance (except $*$), some intervals may not be "maximal", i.e. do not describe classes of tolerance. Below, we show how to replace any interval by its "maximal" interval thanks to a so-called *projection* in a meet-semi-lattice

"Similarity balls" as sets of similar patterns. Firstly, consider the meet-semi-lattice $(D, \sqcap_\theta)$ of interval values for a given attribute. Then, for any interval $x \in D$, we define the ball $B(x, \theta)$ as the set of intervals in D similar to x as follows.

$$B(x, \theta) = \{y \in D \mid y \simeq_\theta x\}$$

This ball of center x and diameter θ contains all intervals y whose meet with x is different of *, meaning that x and y are *similar*.

"Tolerance balls" as classes of tolerance. Now, among this set of intervals, we should remove any pair of intervals that are not pairwise similar, and build an interval with left border (resp. right border) as the minimum (resp. maximum) of all intervals. This can be done by replacing any x of the meet-semi-lattice of intervals by the meet of all intervals y from the ball $B(x,\theta)$ that are not dissimilar with another element y' of this ball, i.e. $y \sqcap_\theta y' \neq *$:

$$\psi(x,\theta) = \textstyle\bigsqcap_{\theta \ e \in B(x,\theta)} y \sqcap_\theta x$$
$$\text{such as } \nexists y' \in B(x,\theta) \text{ with } y \sqcap_\theta y' = *$$

ψ is a mapping that associates to any element a representation of its class of tolerance, i.e. the associated maximal set of pairwise similar elements. For example, with attribute m_3, we have $\psi([2,5],5) = [1,5]$ and $[1,5]$ is the convexification of all elements of the class of tolerance containing $[2,5]$, i.e. the class $\{[1,2],[2,5],[4,5]\}$. When the size of an interval exceeds θ, the ball of similar patterns is empty and ψ returns the element $*$.

Projecting pattern structures. ψ is a mapping that associates to any $x \in D$ an element $\psi(x) \in (D, \sqcap)$ such that $\psi(x) \sqsubseteq x$, as $\psi(x)$ is the meet of x and all intervals similar to x and pairwise similar. The fact $\psi(x) \sqsubseteq x$ means that ψ is contractive. In sense of [8], ψ is a *projection* in the semi-lattice $(D, \sqcap)$ as also monotone and idempotent. Moreover, any projection of a complete semi-lattice $(D, \sqcap)$ is $\sqcap$-preserving, i.e. for any $x, y \in D$, $\psi(x \sqcap y) = \psi(x) \sqcap \psi(y)$ [8].

Thereby, the projection may be computed in advance, replacing each pattern by a "weaker" or "more general" pattern. It comes with a loss of information, e.g in previous example $[2,5]$ replaced by $[1,5]$ which is more imprecise. However, this loss of information is controlled by θ: the projected pattern structure preserves the similarity between descriptions in the original pattern structure, and keeps the same representation formalism while embedding a tolerance relation. We develop in the following an application of concept lattices embedding a tolerance relation, in presence of information fusion problems.

5 An information fusion problem

The problem of information fusion is encountered in various fields of application, e.g sensor fusion, merging multiple sources, etc. Information fusion consists of merging several sources of information for answering questions of interest and make proper decisions [6]. Accordingly, a fusion operator is an operation summarizing information given by sources into a consensual and representative information. In this section, we introduce a real-world information fusion problem in agronomy, concerning pesticide application to fields. Then, we show how this fusion information problem can be solved with a concept lattice involving a tolerance relation. The output is an analysis and an evaluation of agricultural practices w.r.t. pesticide application and subsequent ecological problems.

5.1 Data and problem settings

Agronomists compute indicators for evaluating the impact of agricultural practices on the environment. Questions such as the following are of importance: what are the consequences of the application of a pesticide given the characteristic of this pesticide, the period of application, and the characteristics of the

Table 5: Characteristics of pesticide *glyphosate*.

	$DT50$ day	koc L/kg	ADI g/kg/day
BUS	47	24000	0.3
PM10	[3,60]	[25,68000]	0.3
INRA	[38,60]	167	0.05
Dabene	[38,60]	167	0.05
ARSf	[2,174]	[500,2640]	[0.05,0.3]
ARSl	[2,174]	[500,2640]	[0.05,0.3]
Com96	[2,174]	[25,68000]	0.3
Com98	[38,60]	[500,2640]	0.3
RIVM	[18,66]	[3566,40420]	[0.05,0.3]
BUK	[3,60]	[25,68000]	0.3
AGXf	[8,30]	[301,59000]	0.3
AGXl	[14,111]	[301,59000]	0.3

field? The risk level for a pesticide to reach groundwater is computed by the indicator I_{gro} in [3]. Based on the value of I_{gro}, agronomists try to make a diagnosis of agronomic know-how w.r.t. the use of pesticides. Pesticide characteristics depend on the chemical characteristics of the product while pesticide period application and field characteristics depend on domain knowledge. This knowledge lies in information sources among which books, databases, and expert knowledge in agronomy. Moreover, values for some characteristics may vary w.r.t. information sources.

Here, we are interested in the analysis of practices through the use of *glyphosate* in different countries w.r.t. farmers habits. Glyphosate is a widespread product used by farmers in temperate areas, actually one of the mostly used herbicide in USA[2]. In 2006, IFEN, for French Institute for the Environment, observed that glyphosate is the most encountered substance in French waters, possibly leading to long-term adverse effects in the aquatic environment[3].

Below, three characteristics of glyphosate, namely $DT50$, koc, and ADI, are given in Table 5 (simplified data), according to 12 different information sources.

- $DT50$ represents "half-life" of the pesticide, i.e. time required for the pesticide concentration to decrease of 50% under some conditions. Pesticides with $DT50$ value lower than 100 days can be considered as having a weak impact on groundwater quality in general temperate conditions.
- koc characteristic represents the mobility of the pesticide and depends on pesticide properties and type of soil. Pesticides with high koc values typically stay in upper level of soil and do not reach groundwater. By contrast, pesticides with koc value less than 2200 have good chances to contaminate groundwater.

[2] `http://www.epa.gov/`
[3] `http://www.ifen.fr/`

- ADI (for "Acceptable daily intake") represents toxicity for humans. Glyphosate is considered as having a low toxicity, i.e. no toxic effects were observed for doses of 400 mg/kg/day according to specialized studies.

In Table 5, information sources are not always in agreement. Then, it can be interesting for experts in agronomy to analyse such a table from the point of view of information fusion: which are the sources being in agreement and at which level are they in agreement? This is done using a concept lattice involving a tolerance relation as explained below.

5.2 Methodology and first results

Now, we apply our framework on similarity and scaling to build a concept lattice from Table 5. Three thresholds are defined according to the above observations: $\theta = 100$ for $DT50$, $\theta = 2200$ for koc, and $\theta = 0$ for ADI. Then, for each attribute, classes of similarity and the scale for each attribute are computed and can be read on the lattice in Figure 6.

The lattice shows an interesting classification of information sources w.r.t. information fusion. Each concept in the lattice is composed of an extent with a maximal set of sources in agreement w.r.t. the interval of values in the intent.

The operator used for managing information fusion is convex hull, controlled by a similarity parameter θ, i.e. for two similar intervals the lower bound is the minimum of the two lower bounds and the upper bound is the maximum of the two upper bounds. Let us examine the lattice in detail. The highest concept in the lattice, $\top$, has the intent with the larger intervals (since $*$ is subsumed by any other interval): $[2, 174]$ for $DT50$, $[25, 68000]$ for koc, and $[0.05, 0.3]$ for ADI. The higher a concept is in the lattice, the more information sources in the extent agree on the values to be verified by the attributes. This could be considered as the maximal agreement of all sources but this does not provide any precise information (indeed, the calculation of I_{gro}, which cannot be detailed here, does not allow any recommendation). Moreover, the concepts in the lower levels of the lattice have more restricted intervals. Going further, we can observe that there are four descendants of $\top$ that determine four main parts of the lattice. On the left, there are mainly French and UK information sources, namely AGXf, AGXl, PM10 (French), and BUK and BUS (UK), with com96 denoting an expert committee. In the middle of the lattice, there are mainly French sources, namely RIVM, Dabene, and INRA. Finally, on the right, there are US information sources, namely ARSl, ARSf, and the committee Com98. Interestingly, there is an agreement between European sources as English or French sources share some upper level values such as $[3, 66]$ for $DT50$ or 0.3 for ADI. By contrast, there is no agreement between European and US sources except for the expert committee com98. This shows that practices are actually different and allowed values for pesticide characteristics are not the same w.r.t. the country. Among the possible explanations, it remains very difficult to harvest agronomic data (in cost and time) in every country. For example, the circumstances in which these data are collected are very different w.r.t. climate, soil type, measure devices, etc. In this sense, according to experts in agronomy, the lattice on Figure 6 is a good witness (confirmation) of this diversity of practices and of the agreement degree between sources as given by the lower level concepts.

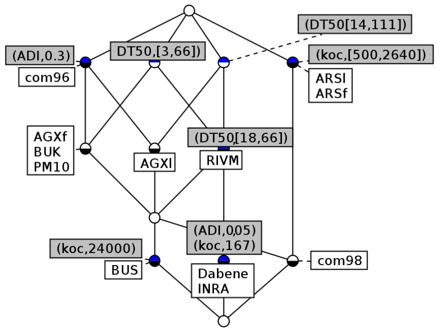

Figure 6: Concept lattice raised from Table 5

6 Discussion

Concept lattices and similarity. Tolerance relations in connection with FCA were studied in several papers [9, 2, 12]. In [12], tolerance relations are introduced from an historical perspective and their role in the formalization of similarity of documents is detailed. In the basic reference [9], it is shown that starting from any complete lattice and a tolerance relation between its elements (from any arbitrary set), there exists a formal context encoding tolerance (pre-)classes. In this work, the statement is used in the opposite way: starting from an arbitrary numerical context, a tolerance relation formalizes the similarity between numerical values and the resulting classes of similarity are then reused for defining scales and a concept lattice encoding the initial numerical context. Other important related work can be found in [2], where fuzzy formal concept analysis introduced. A fuzzy context contains truth values and both attribute and object sets are considered as fuzzy sets. Then a fuzzy concept lattice can be built in the same way as this is done here by grouping pairwise similar objects or attributes with a tolerance-like relation. However, the research work in [2] is much more oriented on the study of mathematical properties of similarity within a concept lattice, contrasting our work on the embedding of constrained tolerance relations in FCA for classifying objects with complex numerical attributes.

Discretization approaches. The scaling procedure proposed in this paper transforms quantitative data into qualitative data. Following [21], this method is: *unsupervised* since class membership of objects is unknown ; *parametric* since a similarity parameter θ has to be given and relies on domain knowledge ; *univariate* as processing each attribute separately ; *ordinal* since taking advantage of the implicit ordering information in quantitative attributes ; and finally and most importantly, *hierarchical* as it builds a partially ordered set (poset) of similarity classes. This poset is actually given by a concept lattice from a formal context encoding a tolerance relation (Section 3) and by a projected meet-semi-lattice of object descriptions (Section 4). This poset is finally used to raise a concept lattice giving a conceptual classification of objects of a domain. Thereby, it can be applied to any kind of structured data for which a similarity can be defined (sequences, graphs, etc.) Cluster-based discretization methods are close to our scaling (see [21]). First, some clusters are searched for, then their intents are used to define intervals for the discretization process. In this paper, we focused on showing how discretization can be automated and controlled (with tolerance relation), with two different approaches, while resulting concept lattices keep the same interpretation.

Processing symmetric contexts. There are many efficient algorithms for generating a concept lattice from a binary context [13]. The efficiency of these algorithms mainly depends on the density of the formal context (G, M, I), i.e. $|I|/|G \times M|$. In the present case, computational complexity is related to the similarity and the range of each attribute. These algorithms may also be used to obtain the partially ordered set of classes of similarity. We propose here two optimizations of FCA algorithms to process symmetric contexts.

Recall that computing classes of similarity for a given attribute can be done either with the upper part or the lower part of the corresponding lattice. Then, a concept is not generated if its dual concept has already been generated. Bottom-up (dually top-down) algorithms are well adapted for this task: concepts (A, B) are generated from bottom to top until the concept verifies $A = B$, i.e. (A, B) belongs to the axis of polarity. Then, interesting concepts are computed by standard FCA algorithms with a modified backtracking condition. This task can be also achieved using well-known and efficient closed itemset mining algorithms [22, 16]. A second optimization relies on the fact that the set $W_m \subset \mathbb{R}$ is totally ordered. For intervals, given $a, b, c, d \in \mathbb{R}$, we have $[a, b] \leq [c, d]$ when $a \leq c$, and if $a = b$ when $b \leq d$. Then, similarity has a monotony property: given $v_1 < v_2 < v_3$, when $v_1 \not\simeq_\theta v_2$ then $v_1 \not\simeq_\theta v_3$. Intuitively, monotony means that the corresponding binary table contains lines and columns of consecutive crosses, e.g. Figure 2 (left). Then, the scan of a context by an FCA algorithm can be reduced accordingly. For example, Figure 7 shows how the performances of the bottom-up algorithm CloseByOne [13] are modified in this case (random data with $\theta = 20$ are used here, but other values of θ give the same result). Worst-case upper bound time complexity of CloseByOne for computing an arbitrary formal context (G, M, I) is $O(|G|^2 \cdot |M| \cdot |L|)$, with L being the set of generated concepts. With both optimizations, complexity becomes $O(|G| \cdot n^2 \cdot |K|)$, whith n the average number of similar elements per element and K similarity classes.

Projecting and processing a pattern structure. Processing interval pattern structures with adaptation of classical algorithms of FCA [13] has been developed in [10]. The authors showed the scalability of concept lattice design from large data, e.g. with thousands objects and dozens attributes. The projection computation is highly related with the maximal clique problem in graph theory, known to be a hard problem. However, since we are dealing with numerical data, and that attribute values can be totally ordered (see above), the projection algorithm detailed in section 4 is of complexity $|W| \cdot n^2$ with W the set of unique data values and n the average number of similar elements per value. A simple algorithm consists in, for each data value, (i) looking for similar elements from a totally ordered set and (ii) testing each pair of the resulting set to keep the maximal set of pairwise similar values. Finally, projected pattern structures are easier to process than non-projected pattern structures, as shown in [8] for graph patterns, while preserving subsumption relations.

Concept lattices and information fusion. Several fusion operators were proposed for combining uncertain information [5]. According to previous works [6],

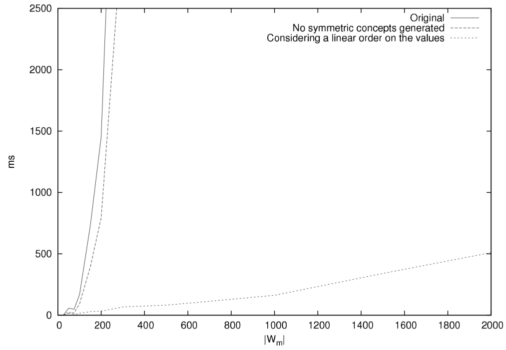

Figure 7: Processing speed of the algorithm CloseByOne with CPU 2.40 Ghz and 4 GB RAM (in ms).

there are three kinds of fusion operators, based respectively on conjunctive, disjunctive and trade-off behaviours. The conjunctive operator is equivalent to set intersection and makes the assumption that all sources are reliable and returns a precise result. The disjunctive operator is equivalent to set union and makes the assumption that at least one source is reliable and returns an imprecise result. The trade-off operators vary between the conjunctive and disjunctive behaviours, and are used when sources are partly conflicting.

Generally, the fusion operator is applied on all information sources, i.e. considering all sources simultaneously, and for one particular variable or attribute at a time, see e.g. [5]. However, this leads to some problems. In our application, a conjunctive operator corresponds to interval intersection and may lead to an empty set when considering all sources. Dually, a disjunctive operator corresponds to interval union leading to a very imprecise fused information. By contrast, the trade-off operator based on maximal coherent subsets (MCS for short) was used in [5]. For numerical information, a MCS is a maximal set of intervals having a non empty intersection. The fusion method based on MCS returns the union of all MCS. In our application, the computation of I_{gro} requires convex inputs, and we propose to control the convexification with the variation of θ. Our method considers maximal subsets of sources with their fusion results and organizes them in a concept lattice. The concept lattice allows to identify which maximal subsets of objects support the most precise results. This opens further research for the use of concept lattices in information fusion.

7 Conclusion

In this paper, we showed how to embed a similarity relation between complex descriptions in concept lattices. We formalized similarity by a tolerance relation. In this way, complex objects are grouped within a same concept when having similar descriptions, extending the ability of FCA to deal with complex data. For that purpose, we proposed two different approaches. A first classical manner is to define a scaling or discretization procedure. This leads to formal contexts from which tolerance preclasses and classes can be obtained, either with FCA algorithms (two optimizations in this case have been developed) or with itemset mining algorithms. A second way, more marginal, consists in representing data by pattern structures, from which a concept lattice can be directly risen. In this case, considering a tolerance relation can be mathematically defined by a projection in a meet-semi-lattice. This allows to use FCA for its knowledge representation and reasoning abilities while working on the same formalism. Moreover, this method is generalizable to any structured data for which a similarity measure can be defined. It remains to carry out a deep analysis on links between existing discretization methods and projections of semi-lattices. Finally, we showed that a concept lattice embedding a tolerance relation is useful for information fusion issues, allowing to characterize subsets of sources with similar and precise information. Further research concerns the embedding of existing fusion operators in concept lattices.

References

[1] F. Baader, B. Ganter, B. Sertkaya, and U. Sattler. Completing description logic knowledge bases using formal concept analysis. In *IJCAI*, pages 230–235, 2007.

[2] R. Belohlávek. *Fuzzy Relational Systems: Foundations and Principles.* Kluwer Academic Publishers, 2002.

[3] C. Bockstaller, P. Girardin, and H. Van der Werf. Use of agro-ecological indicators for the evaluation of farming systems. *European journal of agronomy*, 7:261–270, 1997.

[4] C. Carpineto and G. Romano. Exploiting the potential of concept lattices for information retrieval with CREDO. *J. Univ. Comp.*, 10(8):985–1013, 2004.

[5] S. Destercke, D. Dubois, and E. Chojnacki. Possibilistic information fusion using maximal coherent subsets. *IEEE Transactions on Fuzzy Systems*, 17:79–92, 2009.

[6] D. Dubois and H. Prade. Possibility theory in information fusion. In *Data fusion and Perception*, pages 53–76, 2001.

[7] P. C. Fishburn. *Interval graphs and interval orders: A study of partially ordered sets.* Wiley, 1985.

[8] B. Ganter and S. O. Kuznetsov. Pattern structures and their projections. In *International Conference on Conceptual Structures (ICCS)*, LNCS 2120, pages 129–142, 2001.

[9] B. Ganter and R. Wille. *Formal Concept Analysis.* Springer, mathematical foundations edition, 1999.

[10] M. Kaytoue, S. Duplessis, S. Kuznetsov, and A. Napoli. Two FCA-Based Methods for Mining Gene Expression Data. In *Formal Concept Analysis (ICFCA)*, LNCS 5548, pages 251–266, 2009.

[11] M. Kaytoue, S. O. Kuznetsov, Z. Assaghir, and A. Napoli. Embedding Tolerance Relations in Concept Lattices – An application in Information Fusion. In *International Conference on Information and Knowledge Management (CIKM10)*. ACM, 2010.

[12] S. Kuznetsov. Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In *International Conference on Formal Concept Analysis (ICFCA)*, LNCS 3626, pages 196–225, 2005.

[13] S. Kuznetsov and S. Obiedkov. Comparing Performance of Algorithms for Generating Concept Lattices. *J. Exp. Theor. Artif. Intell.*, 14:189–216, 2002.

[14] D. Lin. An information-theoretic definition of similarity. In *International Conference on Machine Learning (ICML)*, 1998.

[15] L. D. Raedt. *Logical and Relational Learning*. Springer-Verlag, 2008.

[16] L. Szathmary, P. Valtchev, A. Napoli, and R. Godin. Efficient Vertical Mining of Frequent Closures and Generators. In *International Symposium on Intelligent Data Analysis*, LNCS 5772, pages 393–404, 2009.

[17] A. Tversky. Features of Similarity. *Psychological Review*, 84(4):327–352, 1977.

[18] P. Valtchev and R. Missaoui. Similarity-based clustering versus galois lattice building: Strengths and weaknesses. In *Workshop Objects and Classification, A Natural Convergence. European Conference on Object-Oriented Programming*, 2000.

[19] P. Valtchev, R. Missaoui, and R. Godin. Formal Concept Analysis for Knowledge Discovery and Data Mining: The New Challenges. In *International Conference on Formal Concept Analysis (ICFCA)*, LNCS 2961, pages 352–371, 2004.

[20] R. Wille. Why can concept lattices support knowledge discovery in databases? *J. Exp. Theor. Artif. Intell.*, 14(2-3):81–92, 2002.

[21] Y. Yang, G. I. Webb, and X. Wu. Discretization methods. In *The Data Mining and Knowledge Discovery Handbook*, pages 113–130. Springer, 2005.

[22] M. Zaki. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):462–478, 2005.

Contents

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399