Alpha Galois Lattices : an overview

Véronique Ventos[1], Henry Soldano[2]

[1] LRI, UMR-CNRS 8623, Université Paris-Sud, 91405 Orsay, France
ventos@lri.fr
[2] L.I.P.N, UMR-CNRS 7030, Université Paris-Nord,
93430 Villetaneuse, France
soldano@lipn.univ-paris13.fr

Abstract. What we propose here is to reduce the size of Galois lattices still conserving their formal structure and exhaustivity. For that purpose we use a preliminary partition of the instance set, representing the association of a "type" to each instance. By redefining the notion of *extent* of a term in order to cope, to a certain degree (denoted as α), with this partition, we define a particular family of Galois lattices denoted as *Alpha Galois lattices*. We discuss the related implication rules defined as inclusion of such α-extents and show that Iceberg (or frequent) concept lattices are Alpha Galois lattices where the partition is reduced to one single class.

1 Introduction

Galois lattices (or concept lattices) are well-defined and exhaustive representations of the concepts embedded in a data set since they allow us to obtain every subset of instances distinguishable according to a language of class. However, when dealing with real-world data sets the size of such a lattice can be too large to be handled. Various techniques have been proposed to reduce the size of concept lattices by eliminating part of the nodes (e.g. [6]). In particular, frequent concept lattices [13, 15] represent the topmost part of a concept lattice w.r.t. a global criterion of frequency: only nodes with an *extent* cardinality satisfying a threshold according to the whole data set are kept. In this paper, we present more flexible Galois lattices in which the number of nodes is remained in control according to a local criterion of frequency linked to a prior partition of data.

The partition is a set of *basic classes* which are clusters of instances sharing the same basic type. For instance, in real data concerning the electronic catalog of computer products C/Net (http://www.cnet.com), there is 59 different basic types (e.g. *Laptops*, *HardDrives*, *NetworkStorage*) for 2274 instances. Basic classes are then used in order to add a local criterion of frequency to the notion of *extent* as follows: an instance i now belongs to $ext_\alpha(T)$, the α-*extent* of a term T of the language of classes, when it belongs to $ext(T)$, the extent of T, (i.e. i has every T's property), and when at least α % of the instances of the basic class of i also belong to $ext(T)$. This new notion of α-*extent* is used in the galois connection related to the family of *Alpha Galois lattices*.

In comparison with concept lattices (frequent or not), *Alpha Galois lattices* are mainly characterized by the following principles:

- A property (i.e. attribute) can appear in the lattice even if it is not globally enough frequent. For instance, in an Alpha Galois lattice with α less than or equal to 92, the "support" property will appear since in the $HardDrives$ basic class, 92 % instances of $HardDrives$ were sold with support. Actually, this property is not globally frequent (13 products out of 2274, i.e. 0.5 %) and so would not apppear in a frequent concept lattice.
- For the same language, *Alpha Galois Lattices* are coarser than concept lattices since the set of nodes of an *Alpha Galois Lattices* is a subset of the set of nodes of the corresponding concept lattice.
- The values of α define a total order on *Alpha Galois lattices* where the *Alpha Galois lattice* induced by $ext_{\alpha 1}$ is coarser than the *Alpha Galois lattice* induced by $ext_{\alpha 2}$ if $\alpha 1 \geq \alpha 2$. This total order allows to gradually refine our view of the data by partially constructing a sequence of *Alpha Galois lattices.*
- When all individuals share the same type (i.e. when there is only one basic class), the corresponding Alpha Galois lattice is a frequent concept lattice where $\frac{\alpha}{100}$ corresponds to the value of the support threshold *minsupp.*
- The inclusion of α-*extent* corresponds to particular implication rules, i.e. association rules with confidence 1, representing some kind of approximation of usual implication rules, that depends of the *a priori* partition of the data. Such α-implication rules can be extracted from Alpha Galois lattices in the same way as ordinary implication rules are extracted from concept lattices.

The general framework of Galois lattices is given in section 2. In section 3, we present Alpha Galois lattices illlustrated with a simple example. Section 4 presents experimental results on the C/net data set and discuss the ability of such a representation to deal with exceptional data (α near 0 or near 100). Section 5 discusses the particular implications rules related to Alpha Galois lattices and a set theory view of α-extents. Finally, related work and future work are given in section 6.

2 Preliminaries and definitions

Detailed definitions, results and proofs regarding Galois connections and lattices may be found in [1]. Other results concerning Galois lattices in the field of Formal Concept Analysis can be found in [3]. However we need a more general presentation than the one in [3] as our main goal is to construct Galois lattices where the notion of *extent* is not the usual one. In the rest of the paper we denote as Galois lattice the formal structure that we define hereunder and we will denote as concept lattice the Galois lattice as presented in [3].

Definition 1 (Ordered sets and Lattices) *An ordered set is a pair (M,$\leq$) with M being a set and $\leq$ an order relation (reflexive, antisymmetric and transitive) on M. An ordered set (M,$\leq$) is a lattice iff for any pair (x,y) of elements* *of M there exists one least upper bound (or supremum) $x \vee y$, and one greatest lower bound (or infimum) $x \wedge y$*

Definition 2 (Galois connection) *Let m1: $P \rightarrow Q$ and m2: $Q \rightarrow P$ be maps between two ordered sets (P,$\leq_P$) and (Q,$\leq_Q$). Such a pair of maps is called a Galois connection if for all p, p1, p2 in P and for all q, q1, q2 in Q:*

C1- $p1 \leq_P p2 \Rightarrow m1(p2) \leq_Q m1(p1)$

C2- $q1 \leq_Q q2 \Rightarrow m2(q2) \leq_P m2(q1)$

C3- $p \leq_P m2(m1(p))$ and $q \leq_Q m1(m2(q))$

The following simple example will be used in order to illustrate the different notions presented in section 2 and in section 3.

Example 1. The two ordered sets are ($\mathcal{L}$,$\preceq$) and ($\mathcal{P}(I)$, $\subseteq$). $\mathcal{L}$ is a language a term of which is a subset of a set of attributes $\mathcal{A}$ = {t1, t2, t3,a3,a4,a5,a6,a7,a8}. Here c1 $\preceq$ c2 means that c1 is less specific than c2 (e.g. {a3,a4} $\preceq$ {a3,a4,a6}), I is a set of individuals = {i1,i2,i3,i4,i5, i6,i7,i8}. Let *int* and *ext* be two maps *int*: $\mathcal{P}(I) \rightarrow \mathcal{L}$ and *ext*: $\mathcal{L} \rightarrow \mathcal{P}(I)$ such that *int*($\{e1\}$) is the subset of attributes common to all the individuals in $e1$ and *ext*($\{c1\}$) is the subset of instances of I that *belongs* to the term $c1$, i.e. the set of individuals which have all the attributes of c1.
Example 1 is fully described in Figure 1 where each line i represents the *intent* *int*($\{i\}$) of an individual of I and each column j represents the *extent* *ext*($\{j\}$) of an attribute of $\mathcal{A}$.
Together with $\mathcal{L}$ and $\mathcal{P}(I)$, *int* and *ext* define a Galois connection.

Fig. 1. Example 1. $Tab(i,j) = 1$ if the j^{th} attribute belongs to the i^{th} individual.

We also need to recall the definition of closure operators:

Definition 3 (Closure) *w is a closure operator on an ordered set (M,$\leq$) iff for any pair (x,y) of elements of M:*
$x \leq w(x)$ (extensity)
if $x \leq y$ then $w(x) \leq w(y)$ (monotonicity)
$w(x) = w(w(x))$ (idempotency)
An element of M such that x=w(x) is called a closed element of M w.r.t. w.

In a Galois connection $m1 \circ m2$ and $m2 \circ m1$ are closure operators for P and Q. **Example**: *In example 1, $ext(\{a4\}) = \{i1, i3, i4\}$, $int(\{i1, i3, i4\}) = \{a4, a6\}$. The term $\{a4, a6\}$ is therefore a closed term as $int(ext(\{a4\}) = \{a4, a6\}$*

Definition 4 (Galois lattices) *Let m1: P $\rightarrow$ Q and m2: Q $\rightarrow$ P be maps between two lattices (P,$\leq_P$) and (Q,$\leq_Q$), such that (m1,m2) is a Galois connection.*
Let G=$\{$ (p,q) with p an element of P and q an element of Q such that p=m2(q) and q = m1(p)$\}$
Let $\leq$ be defined by: (p1,q1) $\leq$ (p2,q2) iff q1 $\leq_Q$ q2.
(G,$\leq$) is a lattice called a Galois lattice. When necessary it will be denoted as $G(P, m1, Q, m2)$.
Example*: In Example 1, we have G=$\{$(c,e) where c belongs to $\mathcal{L}$, and e belongs to $\mathcal{P}(\mathcal{I})$ and are such that e=ext(c) and c=int(e)$\}$. Then (G, $\leq$) is a Galois lattice where $\leq$ is defined by: (c,e) $\leq$ (c',e') iff e $\subseteq$ e' (which is equivalent to c $\succeq$ c'). The Galois lattice corresponding to example 1 is presented in Figure 2.*

Fig. 2. The Galois Lattice corresponding to example 1

Note that this means that in Galois lattices a node is a pair of closed elements of P and Q. Furthermore the functions $m1$ and $m2$ define equivalence relations on the lattices P and Q as follows:

Definition 5 (Equivalence relations on P and Q) *Let $\equiv_P$ and $\equiv_Q$ denote the equivalence relations defined on P and Q by the correspondances $m1$ and $m2$, i.e. let p1, p2 be elements of P and q1, q2 be elements of Q:*
$p1 \ \equiv_P \ p2$ *iff* $m1(p1) = m1(p2)$*, and* $q1 \ \equiv_Q \ q2$ *iff* $m2(q1) = m2(q2)$

Lemma 1 *Let p be an element of P, and q be an element of Q, then m2(m1(p)) is the unique greatest element of the equivalence class of $\equiv_P$ containing p and m1(m2(q)) is the unique greatest element of the equivalence class of $\equiv_Q$ containing q.*

So, a characteristic property of Galois lattices is that each node (p, q) is a pair of representatives of their respective equivalence classes.

In our previous example, we used the language $\mathcal{L}$, defined as the powerset $\mathcal{P}(\mathcal{A})$ of a set of attributes A, as the first lattice, and the powerset $\mathcal{P}(\mathcal{I})$ of a set of individuals I as the second lattice. This, together with the two correspondances *int* and *ext* define a particular Galois lattice known as a *concept lattice* [3]. In concept lattices, a node (c, e) is a concept, c is the *intent* and e is the *extent* of the concept. Concept lattices are interesting both from a practical point of view, as they express in a rigorous way the two sides of a concept, and from a theoretical point of view, as it has be shown that any finite lattice may be rewritten as a concept lattice [3] .

3 Alpha Galois lattices

In what follows we consider, with no generality loss, $\mathcal{L} = \mathcal{P}(A)$ and we start with the concept lattice $G(\mathcal{L}, ext, \mathcal{P}(I), int)$ as previously examplified. Then we will discuss a variation on *ext* which purpose is to obtain an equivalence relation $\equiv'_{\mathcal{L}}$ coarser than the original one (see definition 11) thus resulting in larger equivalence classes on $\mathcal{L}$ and so on less nodes in the corresponding Galois lattice.

The new *ext* function relies on the association of a predefined type to each individual of I. The corresponding clusters of instances are denoted as *basic classes.*The first idea is then to gather such clusters rather than individuals (see [11]).
For instance, let us assume that the attributes $t1, t2, t3$ express the types of the individuals of example 1. These types corresponds to three basic classes $BC1$, $BC2$, $BC3$ whose descriptions are the following:
BC1={i1,i2}, int(BC1)= {t1,a3,a6}; BC2={i3,i4,i5}, int(BC2)= {t2,a6};
BC3={i6,i7,i8}, int(BC3)= {t3,a3,a6,a8}.
Let us consider the concept lattice built on a new set of individuals: {bc1,bc2,bc3} (let us call them the *prototypes* of their respective basic classes) such that, for any index i, $int(BCi) = int(\{bci\})$. This concept lattice is represented in Figure 3 as a particular case of Alpha Galois lattice, and is much smaller than the original concept lattice.
Now, it seems interesting to have an intermediary approach where the entities gathered could be other subsets of I than only individuals or whole basic classes. This leads to the definition of *Alpha Galois lattices*.

3.1 Alpha definitions

Definition 6 (Alpha satisfaction) *Let α belong to [0,100]. Let e=$\{i_1, \ldots, i_n\}$ be a set of individuals and T be a term of $\mathcal{L}$. Then,*
$e\ \alpha-satisfies\ T\ (e\ sat_\alpha\ T)\ iff \mid ext(T) \cap e \mid\ \ \geq \frac{|e|.\alpha}{100}$

Since the Alpha satisfaction is defined according to a set of individuals and to a term of the language $\mathcal{L}$, we can use it to check whether at least α % of a basic class satisfies a term of $\mathcal{L}$ and add this constraint to isa, the classical membership relation between individuals and terms. We call this notion (membership relation plus Alpha satisfaction of the basic class) the *Alpha membership relation*.

Definition 7 (Alpha membership relation) *Let I be a set of individuals and $\mathcal{BC}$ be a partition of I in a set of basic classes. Let $BCl : I \rightarrow \mathcal{BC}$ be such that $BCl(i)$ is the basic class to which belongs i, and let T be a term of $\mathcal{L}$, then:*
$i\ isa_\alpha\ T$ *iff* $i\ isa\ T$ *and* $BCl(i)\ sat_\alpha\ T$

Example (example 1). *Let T=$\{a6,a8\}$, $ext(T) = \{i1,i3,i5,i6,i7,i8\}$. $BC1\ sat_{50}$ T since $i1$ isa T and $\mid BC1 \mid =2$. As a result $i1\ isa_{50}\ T$. $BC2\ sat_{60}\ T$ since $\mid ext(T) \cap BC2 \mid \geq \frac{|BC2|.60}{100}$. So we have $i3$ and $i5$ $isa_{60}\ T$. Finally $BC3\ sat_{100}\ T$ since 100 % of the individuals of $BC3$ belong to the extent of T. So we have $i6$, $i7$, and $i8$ $isa_{100}\ T$.*

Finally, we use the *Alpha membership relation* to define the notion of *extent* used in Alpha Galois Lattices.

Definition 8 (Alpha extent of a term) *The α-extent of T in I w.r.t. the set $\mathcal{BC}$ of basic classes is the following set:*

$$ext_\alpha(T) = \{i \in I \mid i\ \ isa_\alpha\ \ T\}$$

Example (example 1) *: Let T=$\{a6,a8\}$, $ext_0(T)$= $ext(T) = \{i1, i3,i5, i6,i7,i8\}$. $ext_{60}(T)$= $\{i3,i5,i6,i7,i8\}$ and $ext_{100}(T)$= $\{i6,i7,i8\}$*

The following proposition about the new Galois connection needs the definition of E_α, a subset of $\mathcal{P}(\mathcal{I})$ whose elements are made of large enough parts of basic classes.

Proposition 1 *Let E_α be the following subset of $\mathcal{P}(\mathcal{I})$:*
$E_\alpha = \{e \in \mathcal{P}(I) \mid\ \forall i \in e, \mid e \cap BCl(i) \mid\ \ \geq \frac{|BCl(i)|.\alpha}{100}\}$.
Then : int and ext_α define a Galois connection on $\mathcal{L}$ and E_α
Proof*: The proof relies on theorem 1 given in the next section and is presented as the proof of a corollary.*

We can therefore define Galois lattices from this new Galois connection and we called them *Alpha Galois lattices*.

Definition 9 (Alpha Galois lattices) *The Galois lattice $G(\mathcal{L}, ext_\alpha, E_\alpha, int)$ corresponding to the connection defined above is called an Alpha Galois lattice and is denoted as G_α.*

When α is equal to 0, $E_\alpha = \mathcal{P}(I)$ and $ext_\alpha = ext$. Therefore, the Alpha Galois lattice is the Concept lattice corresponding to the same attributes and instances. When α is equal to 100, the nodes of the Galois lattice are only whole basic classes gathered. As a consequence the Alpha Galois lattice is the concept lattice obtained by considering as instances the *prototypes* of the basic classes.

The whole G_{100} Alpha Galois lattice of Example 1 is represented in Figure 3. Figure 4 presents the topmost part of G_{60}. Note that *intents* of the nodes of G_{100} are also intents of nodes of G_{60} that in turn are all intents of nodes of the original concept lattice G_0 (see Figure 2).

Fig. 3. When α =100 : the G_{100} Alpha Galois lattice of example 1 is much smaller than the original Concept lattice presented in Figure 2.

Fig. 4. $\alpha = 60$: The topmost part of G_{60} of example 1. New nodes, w.r.t. G_{100} are the lighter ones.

Moreover, there exists a total order on alpha Galois lattices defined in the next section.

3.2 Alpha Galois lattice order

In [4] the authors give a formal view to the extension of formal concept analysis to more sophisticated languages of terms and use the notion of *projection* as a way to obtain smaller lattices by reducing the language. [11] independently uses the same notion of projection with a similar scope and also introduce *extensional* projections to modify the *ext* function. We recall hereunder the notion of projection:

Definition 10 (Projection) *Proj is a projection of an ordered set (M,$\leq$) iff for any pair (x,y) of elements of M:*
$x \geq$ Proj(x) (minimality)
if $x \leq y$ then Proj(x) $\leq$ Proj(y) (monotonicity)
Proj(x) = Proj(Proj(x)) (idempotency)

By changing *ext*, an extensional projection changes a Galois lattice to a smaller one which corresponding equivalence relation $\equiv'_{\mathcal{L}}$ is coarser than the original one, thus resulting in larger equivalence classes on $\mathcal{L}$ [11]:

Definition 11 *Let $\equiv^1_{\mathcal{L}}$ be associated to the function ext1 and $\equiv^2_{\mathcal{L}}$ be associated to ext2, then, $\equiv^2_{\mathcal{L}}$ is said coarser than $\equiv^1_{\mathcal{L}}$ iff for any pair (c1,c2) of elements of $_{\mathcal{L}}$ we have:*
if ext1(c1) = ext1(c2) then ext2(c1) = ext2(c2).

The following theorem [11] has a corollary that proves the proposition 1:

Theorem 1 (An extensional order on Galois connections) *Let int and ext define a Galois connection on $\mathcal{L}$ and E, and let $proj$ be a projection of E. Let $E' = proj(E)$ and $ext' = proj \circ ext$. Then:*

1) int, ext' define a Galois connection on $\mathcal{L}$ and E'.

2) The Galois lattice $G'(\mathcal{L}, ext', E', int)$ has the following property: for any node g'=(c,e') in G' it exists a node g=(c,e) in $G(\mathcal{L}, ext, E, int)$, with the same intent c, such that e'=proj(e).

3) $\equiv'_{\mathcal{L}}$ is coarser than $\equiv_{\mathcal{L}}$.

We will say then that G' is coarser than (or nested in) G and write G'= proj(G). Let (c,e) be a node of G, then proj(e,c) = (int$^\circ$proj(e), proj(e)) is the projected node in G'.

Corollary 1 *Let $G(\mathcal{L}, ext, P(I), int)$ be a Galois lattice. Let ext_α=$proj_\alpha \circ ext$ and let $E_\alpha = proj_\alpha(\mathcal{P}(\mathcal{I}))$ such that for any e in $\mathcal{P}(\mathcal{I})$:*
$proj_\alpha(e) = e - \{i/\ i\ \in e \text{ and } \mid e \cap BCl(i) \mid < \frac{|BCl(i)|.\alpha}{100}\}$.
- int, ext_α define a Galois connection on $\mathcal{L}$ and E_α and $G(\mathcal{L}$, ext_α, E_α, int) is a Galois lattice coarser than G. **proof :** *In order to prove this corollary, we simply have to show that $proj_\alpha$ is a projection: -$proj_\alpha(e)$ is included in e since we remove elements of e, so $proj_\alpha$ is minimal. - If e is included in e', every element of e removed when applying $proj_\alpha$ will also be removed when applying $proj_\alpha$ on e', so $proj_\alpha$ is monotonic. - finally, $proj_\alpha$ is idempotent since no more*

element of $proj_\alpha(e)$ can be removed by applying again $proj_\alpha$.

Furthermore, we can order the *alpha extents* according to the value of α: For every pair ($\alpha 1$, $\alpha 2$) such that $\alpha 1 \leq \alpha 2$, $ext_{\alpha 2} = proj_\alpha \circ ext_{\alpha 1}$ with $\alpha = \alpha 2$. As a consequence, the value of α defines a total order on Alpha Galois lattices:

Proposition 2 (A total order on Alpha Galois lattices) *Let us denote as $\equiv_\alpha$ the equivalence relation on $\mathcal{L}$ associated to ext_α. Then for every pair ($\alpha 1$, $\alpha 2$) such that $\alpha 1 \leq \alpha 2$, $\equiv_{\mathcal{L}}^{\alpha 2}$ is coarser than $\equiv_{\mathcal{L}}^{\alpha 1}$*
proof: *$proj_\alpha$ is a projection for every value of α belonging to [0,100].*
$ext_{\alpha 1}$=$proj_\alpha \circ ext$ with $\alpha = \alpha 1$ and $ext_{\alpha 2}$=$proj_\alpha \circ ext_{\alpha 1}$ with $\alpha = \alpha 2$. According to 3) of theorem 1, $\equiv_{\mathcal{L}}^{\alpha 2}$ is then coarser than $\equiv_{\mathcal{L}}^{\alpha 1}$

Example: *$\equiv_{\mathcal{L}}^{100}$ is coarser than $\equiv_{\mathcal{L}}^{60}$ which is in turn coarser than $\equiv_{\mathcal{L}}^{0}$ that is the equivalence relation $\equiv_{\mathcal{L}}$ of the concept lattice.*

The previous proposition is the basis to make successive refinements in Alpha Galois lattices (see section 4)

There is also a partial order associated to the initial partition $\mathcal{BC}$ of I in Basic classes. Let us suppose that we substract some basic classes from I and so from $\mathcal{BC}$ thus obtaining a reduced instance set I' together with a reduced partition $\mathcal{BC}'$ made of the remaining basic classes. It is then easy to show (proof omitted here) that there is a projection $proj$ such that the corresponding E'_α simply rewrites as $proj(E_\alpha)$. As a consequence we have the following property:

Proposition 3 (A partial order on Alpha Galois lattices) *Let $\mathcal{BC}'$ be a subset of the basic classes constituting $\mathcal{BC}$, then, for the same α, the Alpha Galois lattice G' associated to $\mathcal{BC}'$ is coarser than the Alpha Galois lattice G associated to $\mathcal{BC}$.*

An interesting case is the one of the partition $\{I\}$ in which we consider only one single class, i.e. the case in which all individuals share the same type. The corresponding Alpha Galois lattice is the topmost part of the concept lattice defined by the same language $\mathcal{L}$ and the same set I of individuals. The lattice then only contains nodes whose extents have a size greater than $\frac{\alpha}{100}|I|$ (plus the bottom node whose extent is empty). This structure has been previously investigated and is denoted as an *Iceberg* (or *frequent*) *Concept lattice* [13, 15] where $\frac{\alpha}{100}$ corresponds to the value of the support threshold *minsupp*.

Note that because of proposition 3, the frequent lattice of any basic class BCi of a partition $\mathcal{BC}$ is always coarser than the Alpha Galois lattice corresponding to $\mathcal{BC}$.

4 Experiments

The program ALPHA that computes Alpha Galois lattices relies on a straightforward top-down procedure in which nodes are generated as follows: a current node intent c is specialized by adding a new attribute a, then $int^{\circ}ext_{\alpha}$ is applied to $c \cup \{a\}$ in order to obtain a closed term; the corresponding node has then to be compared to previous nodes in order to avoid duplicates. We have experimented ALPHA on a real dataset composed of 2274 computer products extracted from the C/Net catalog. Each product is described using a subset of 234 attributes. There are 59 types of products and each product is labelled by one and only one type.

In our first experiment we have built G_{100} using the whole data set (so practically restricted to 59 prototypical instances), Then we smoothly lowered the value of α and recomputed the corresponding G_{α} lattice. As we can see hereunder the number of nodes (and so the CPU time) exponentially grows from 211 concepts to 107734 as α varies from 100% to 92%. This means that it is here impossible to have a complete view of the data at the level of instances (α=0) and that even relaxation of the basic class constraint (starting with α=100) has to be limited:

Alpha	100	98	96	94	92
Nodes	211	664	8198	44021	107734

Our second experiment concerns the part of G_{100} between the node which extent contains the 3 basic classes (*Laptop* (252 instances, 39 attributes involved), *Harddrive*(45 instances, 22 attributes), *Network-storage*(4 instances, 16 attributes)) and the *Bottom* node. The new G_{100} contains now 5 nodes (to be compared to the maximum number of nodes $2^3 = 8$). Here computation of G_{α} is performed for a complete set of α values from 100% to 0% together with the corresponding frequent lattices (see Figure 5). We are first interested in what happens with

Fig. 5. Number of nodes vs Alpha values for frequent lattices and Alpha Galois lattices

high values of α. Starting from G_{100}, new nodes appear as α slowly decreases.

For instance at $\alpha = 99\%$, a new node appears under the G_{100} node standing for the basic class *Laptop*. The intent of the new node now contains the attribute "network-card". This is due to the fact that most instances of the class *Laptop* do possess a network card. So by relaxing the basic class constraint we get rid of the few, exceptional, instances of *Laptop* found in the catalog and that were hiding this "default" property of *Laptop* in G_{100}. In the same way most *hard-drives* are sold with "support". So at $\alpha = 92\%$, a new node representing *hard-drives* with "support" appears. Note that in this case, the attribute "support" is unfrequent when considering all the instances ("support" appears in 13 products out of 301) and so would not be considered in a *frequent concept lattice*, whereas it is frequent within the *hard-drive* class (13 products out of 15) and so comes out in the Alpha Galois lattice G_{90}. As a summary, by slowly decreasing α from 100 % we have a more accurate view of our data by revealing properties that are relevant to at least some basic class.
Now as α slowly grows from 0 to small values (say 10%), some instances, which behavior is *exceptional* within their basic class w.r.t. some term t of $\mathcal{L}$, will disappear from the corresponding α-extent. These instances are exceptional as they belong to the extent of the term t whereas very few instances of the same basic class do belong to this extent. As a result some properties that are very unfrequent within some basic class will no more be allowed to discriminate concepts. For example, only few *Laptops* have the property "Digital-Signal-Protocol", and so when $\alpha = 6\%$, nodes which intent contains the "Digital Signal Protocol" property no more include instances of *Laptop* in their extent. As a result terms including" Digital-Signal-Protocol" become equivalent whenever their extent only differed because of Laptop instances, thus resulting on a smaller (and so simpler) lattice. However a closer look to Figure 5 shows there can be a large number of nodes even for high values of α. In this particular example this is due to the fact that one basic class, namely *Laptop*, has a huge frequent lattice that invades the Alpha Galois lattice (data not shown). An experiment with 24 basic classes and 1187 objects (some large basic classes are removed thus resulting in a more homogenous class size distribution) shows that the size of Alpha Galois lattices can be really different from the one of frequent lattices (see also Figure 6) :

Alpha Values	100	80	50	30	0
Alpha Nodes	158	842	1493	1900	2202
Frequent Nodes	2	18	18	50	2202

5 Issues related to the Alpha view of data

5.1 Combining global and local frequency constraints : Frequent Alpha Lattices

In one hand, frequent concept lattice apply a global frequency constraint to the concept lattice : all nodes whose extents are small enough are eliminated (i.e. sent to the bottom node). When the threshold is high this unfortunately tends

Fig. 6. Number of nodes vs Alpha values for frequent lattices and Alpha Galois lattices

to eliminate many attributes that are relevant to at least some basic classes. In an other hand, as we have seen, Alpha Galois lattices apply a local frequency constraint in such a way that attributes relevant to at least some basic class appear in the lattice. However, a side effect is that an Alpha Galois lattice may still be very large, especially when using small values of *alpha*. Our proposal here is to combine these two constraints : we will only consider nodes whose α-*extent* is large enough. Applying the global constraint allows to eliminate nodes that are locally frequent on some basic classes, and so would be interesting, but still represent few instances and so can be discarded when we want a simpler view of the data.

The result of such a filter is again a Galois lattice. More precisely let us consider the function $proj^f$ on E_α such that $proj^f(e) = e$ whenever $\frac{|e|}{|I|} \geq f$ and $proj^f(e) = \emptyset$ otherwise. $proj^f$ clearly is a projection, and therefore G_α^f $=proj^f(G_\alpha)$ is a Galois lattice coarser than G_α. More precisely it corresponds to the topmost part of G_α plus a bottom node.

We denote G_α^f as a *Frequent Alpha Lattice* associated to the instance set I, the partition $\mathcal{BC}$ of I, the Alpha value α and the global frequency threshold f. The corresponding α-implication rules (see next section) have a support greater than f. Note that we will speak here of a α-support since the support is computed using α-extents.

5.2 Alpha implications rules

Association rules, as usually defined in data mining, are implications whose truth values are observed on a set of instances I. Each association rule is related to a "support" value, i.e. the frequency of its antecedent part within the instance set, and to a "confidence" value. When the confidence values is 1 they are called *implication rules*. When considering Concept lattices the partial order induced on terms by the Galois connection corresponds to implications rules. More precisely $ext_I(T_1) \subseteq ext_I(T_2)$ means that the logical implication $T_1 \rightarrow T_2$ holds on I. In such rules, T_1 will be denoted as the left part and T_2 as the right part. In frequent concept lattices, the extent of a term is redefined as empty whenever the term is unfrequent in I, i.e., when its original extent contains less than $minsupport * |I|$ instances of I. As a consequence the corresponding implication rules all have a support greater than $minsupport$.

Association rules corresponding to an instance set I are efficiently constructed in two steps, first constructing the frequent concept lattice corresponding to I. The intent of the nodes are usually denoted as *closed frequent itemsets.* Association rules are then built using closed frequent itemsets [9, 16]. The basic idea is that, as mentioned before, a node in the concept lattice corresponds to an equivalence class of terms, all sharing the same extent. In particular, the intent of the node, i.e., the unique greatest term, has the same extent as all the smallest terms. We obtain then a set of implication rules whose left parts are the smallest elements, and whose right part is the intent of the node. Part of this set corresponds to the non-redundant Guigues-Duquenne Basis of implications rules [5] related to I. For sake of clarity, the left part of each rule is substracted from the rigth part. For instance, let the node be $(abc, \{i1, i2, i3\})$ and suppose that the smallest terms of the corresponding equivalence classe are $\{a, b\}$ (this means that $ext_I(a)= ext_I(b) = \{i1, i2, i3\}$). We obtain then the implications rules $\{a \rightarrow abc, b \rightarrow abc\}$ that are rewritten as $\{a \rightarrow bc, b \rightarrow ac\}$.

Now, in Alpha Galois lattices, whenever $ext_\alpha(T_1) \subseteq ext_\alpha(T_2)$ we will say that the *α-implication* $T_1 \rightarrow_\alpha T_2$ holds on the pair $(I, \mathcal{BC})$. Because they are derived from a Galois lattice, *α-implication* are transitive, monotonic and additive:

-If $T_1 \rightarrow_\alpha T_2$ and $T_2 \rightarrow_\alpha T_3$, then $T_1 \rightarrow_\alpha T_3$

-If $T_1 \rightarrow_\alpha T_2$, and $T_1 \preceq$ T, then $T \rightarrow_\alpha T_2$

-If $T_1 \rightarrow_\alpha T_2$ and $T_1' \rightarrow_\alpha T_2'$, then $T_1 \cup T_1' \rightarrow_\alpha T_2 \cup T_2'$[3]

Furthermore we have the *modus ponens* as an inference rule:

If $i\ isa_\alpha\ T_1$ and $T_1 \rightarrow_\alpha T_2$, then $i\ isa_\alpha\ T_2$

Discussion about *α-implication* rules and construction of a basis of such rules, in an similar way as the original Guigues-Duquenne basis, is beyond the scope of this paper (note that basically we simply have to compute α-extents rather than extents when adapting existing algorithms). We would however emphasize on an exemple the meaning and usefulness of such rules to handle exceptions when individuals are labelled with basic classes as proposed in this paper. For this purpose, let us suppose that we have divided animals (i.e. individuals) into basic classes as *mammals, birds, insects* and that we search for general rules in the data. Some intuitive rule is the following : an animal that *flies* should have *wings.* This rule holds as well for birds (unflying birds, as ostriches, does not contradict the rule) as for insects. The rule should also hold for mammals, that generally does not fly, but is falsified by a flying squirrel. The Alpha approach benefits here from the fact that very few mammals flies (in other words the antecedent part of the rule is unfrequent within the basic class to which belong the individual that falsifies the rule). When using α-extents, the flying-squirrel is removed from the antecedent part of the rule. Here, a small value of α is sufficient to obtain an *α-implication* rule expressing that flying

[3] For instance $ab \cup bc = abc$

animals have wings. Of course greater values of α, namely close to 1, also preclude falsifying the rule. However in the latter case α-implication rules express something different : they applies to individual whenever the antecedent part is common to *most* individuals of the same basic class. In our example, only *birds* would be concerned with such a rule, but not *insects*.

The Guigues-Duquenne basis of implication rules has been extended to rules with a minimal support *minsupp*. Also the Luxemberger basis of association rules summarizes rules whose confidence is less than 1 but is greater or equal to a minimal confidence level $minconf$. The corresponding basis has also been also extended to rules with a minimal support. Both extended bases are computed using the closed terms of the corresponding frequent lattice [9, 12]. First we adapt definitions of support and confidence to Alpha rules by changing *extents* to α-*extents*:

Definition 12 *An α-association rule is a pair of terms T_1 and T_2, denoted as $T_1 \to_\alpha T_2$.*

The support and confidence of an α-association rule $r = T_1 \to_\alpha T_2$ are defined as follows :

α-*supp(r)* $= \frac{|ext_\alpha(T_1 \cup T_2)|}{|I|}$

α-*conf(r)* $= \frac{|ext_\alpha(T_1 \cup T_2)|}{|ext_\alpha(T_1)|}$

The α-association rule $r = T_1 \to_\alpha T_2$ holds on the pair $(I, \mathcal{BC})$ whenever α-supp$(r) \geq minsupp$ and α-conf$(r) \geq minconf$.

Note that when we consider the implication rules derived from a Galois lattice, the right part T_2 of the rule is an *intent* and the left part T_1 is smaller than T_2. As a consequence we have $T_1 \cup T_2 = T_2$ and the α-support rewrite as $\frac{|ext_\alpha(T_2)|}{|I|}$. This means that the set of rules whose α-support is greater than *minsupport* is obtained from the nodes of the frequent Alpha lattice $G_\alpha^{minsupport}$.

5.3 Alpha extensions and Rough sets

Note that there is a set theory view of the present work : we consider that an individual i α-belongs to a subset e of I (denoted as $i \in_\alpha e$) whenever i belongs to $proj_\alpha(e)$. So $i \in ext_\alpha(T)$ rewrites as $i \in_\alpha ext(T)$. Now there is another particular set theory view of *a priori* partitioned data referred to as *rough sets* theory ([10]). In the rough sets view, a subset e of I has a greatest lower bound $inf(e)$ and a least upper bound $sup(e)$ amongst unions of "Basic classes". A swallow formal comparison between this two views shows that, for any subset e of I, $proj_{100}(e) = inf(e)$. However rough sets theory is aimed to represent membership of an element to a set with some degree of uncertainty : the rough membership $\mu_e(i)$ is the proportion of individuals belonging to the same basic class as i, that also belongs to e . To compare the intended meaning of these two views we consider α as a threshold applied to this membership μ. This results on the following membership relation :

Definition 13 *Let i be an element of I and let e be a subset of I, then the thresholded rough membership function $\in^R_\alpha$, is defined as :*
$i \in^R_\alpha e$ *iff* $\mu_e(i) \geq \frac{\alpha}{100}$

Note that whenever $i \notin ext(T)$, then $i \notin_\alpha ext(T)$, whereas possibly $i \in^R_\alpha ext(T)$. Indeed, the partitioning on rough sets expresses some indiscernibility between individuals of the same basic class, so, the rough sets view results in some degree of membership of i to e. provided that $e \bigcap Bc(i)$ is not empty. At the contrary in the Alpha set theory view, membership of i to an extent e is a prerequisite for its α-membership. Then, the α-membership of i to e finally depends on to what extent membership to e is frequent amongst individuals of the same basic Class as i : if i is exceptional w.r.t. a term t, then it does not belong to $ext_\alpha(t)$ even if i belongs to the "classic" extent of t. A fortunate consequence of the latter view is the opportunity to construct Galois lattices smaller than concept lattices.

6 Related work and conclusion

Recent work in Knowledge Representation and Machine Learning investigates Galois connections and lattices based on languages of terms more complex than those used in concept lattices, so modifying the notion of intent of a concept [3, 2, 8, 4]. We have shown here that by restricting the notion of extent of a term with respect to a *a priori* partition of the instance set I, we also modifies the lattice of extents which is no longer P(I) and we obtain a new family of Galois lattices. As mentioned above Iceberg (or frequent) concept lattices [15, 13] formally are Alpha Galois lattices in which all individuals belong to the same basic class. Besides, the implication rules related to Alpha-Galois lattices simply correspond to inclusion of α-extents and a canonical basis of such α-*implication* rules can be extracted from the Alpha-Galois lattices in the same way as from frequent concept lattices. Note that α-*implications* rules inherit from the Galois lattice structure interesting properties (as transitiviy) unusual when dealing with "approximate" rules.

About construction of Alpha Galois lattices, it should be interesting to adapt efficient algorithms aimed at the construction of concept lattices (e.g. [7]) However, as a consequence of property 3, another way to build Alpha Galois lattice is to build the iceberg lattices corresponding to each basic class and then combine them using a *subposition* operator as previously proposed by [14] to efficiently build concept lattices. The corresponding Alpha Galois lattice simply is the result of this composition. Note that this is the basis of a *basic class incrementality* of Alpha Galois lattices. As a conclusion there is still much work to experiment and to investigate theoretical issues and practical use of Alpha Galois lattices and corresponding α-implication rules. However we do believe that they represent a flexible tool to investigate data and handle exceptions that are relative to a preliminary view of the data. This preliminary view is part of the bias that the advertised user applies to extract and learn information from data.

Acknowledgments Many thanks to Nathalie Pernelle for its valuable contribution to the work presented here, and to Philippe Dague for its patient reading of an earlier draft of this paper.

References

1. Gareth Birkhoff. *Lattice Theory.* American Mathematical Society Colloquium Publications, Rhode Island, 1973.
2. J. Ganascia. Tdis: an algebraic formalization. In *Int. Joint Conf. on Art. Int.*, volume 2, pages 1008–1013, 1993.
3. B. Ganter and R. Wille. *Formal Concept Analysis: Logical Foundations*. Springer Verlag., 1999.
4. Bernhard Ganter and Sergei O. Kuznetsov. Pattern structures and their projections. *ICCS-01, LNCS*, 2120:129–142, 2001.
5. J.L. Guigues and V. Duquenne. Famille non redondante d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences humaines*, 95:5–18, 1986.
6. Joachim Hereth, Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery and data analysis. In *Int. Conf. on Conceptual Structure.*, pages 421–437, 2000.
7. S. Kuznetsov and S. Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. of Experimental and Theoritical Art. Int.*, 2/3(14):189–216, 2002.
8. M. Liquiere and J. Sallantin. Structural machine learning with galois lattice and graphs. In *ICML98, Morgan Kaufmann*, 1998.
9. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25?46, 1999.
10. Zdzislaw Pawlak. Rough sets, rough relations and rough functions. *Fundamenta Informaticae*, 27(2/3):103–108, 1996.
11. N. Pernelle, M-C. Rousset, H. Soldano, and V. Ventos. Zoom: a nested galois lattices-based system for conceptual clustering. *J. of Experimental and Theoretical Artificial Intelligence*, 2/3(14):157–187, 2002.
12. Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Intelligent structuring and reducing of association rules with formal concept analysis. *Lecture Notes in Computer Science*, 2174:335–349, 2001.
13. Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with titanic. *Data and Knowledge Engineering*, 42(2):189–222, 2002.
14. P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards building galois (concept) lattices. *Discrete Mathematics*, 256(3):801–829, 2002.
15. K. Waiyamai and L. Lakhal. Knowledge discovery from very large databases using frequent concept lattices. In *11th Eur. Conf. on Machine Learning, ECML'2000*, pages 437–445, 2000.
16. M. J. Zaki. Generating non-redundant association rules. *Intl. Conf. on Knowledge Discovery and DataMining (KDD 2000)*, 2000.