Alpha Galois Lattices

Véronique Ventos
L.R.I., UMR-CNRS 8623, Université Paris-Sud, 91405 Orsay, France
Henry Soldano
L.I.P.N, UMR-CNRS 7030, Université Paris-Nord, 93430 Villetaneuse, France
Thibaut Lamadon
L.R.I., UMR-CNRS 8623, Université Paris-Sud, 91405 Orsay, France

Abstract

In many applications there is a need to represent a large number of data by clustering them in a hierarchy of classes. Our basic representation is a Galois lattice, a structure that exhaustively represents the whole set of concepts that are distinguishable given the instance set and the representation language. What we propose here is a method to reduce the size of the lattice, and thus simplify our view of the data, while conserving its 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 also discuss the related implication rules defined as inclusion of such α-extents.

1 Introduction

One way to cluster instances in classes organized in a hierarchy is to build a concept lattice [4], a structure in which each node corresponds to a class represented as its *extent* (the set of the instances of the class) and its $intent$ (the common properties of these instances expressed as a term of a given language). Concept lattices express all the subsets of instances distinguishable when using the language. Various techniques have been proposed to reduce the size of concept lattices by eliminating part of the nodes. In particular, frequent concept lattices [11, 10] represent the topmost part of a concept lattice, i.e. the nodes which $extent$ cardinality exceeds a given threshold. In our approach, we reduce the number of nodes of the concept lattice by accounting in a flexible manner a prior partition of data. The partition is a set of *basic classes* which are clusters of instances sharing the same basic type. 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 the α-$extent$ of a term T of the language, 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 $ext(T)$. Defining α-$extents$ results in a family of flexible concept lattices that we call *Alpha Galois lattices*. For instance, in an Alpha Galois lattice representing the C/net electronic catalog, with α= 92, the "support" property will appear since in the $HardDrives$ basic class, 92 % instances were sold with support. Actually, this property is not frequent (13 products out of 2274, i.e. 0.5 %) and so would not apppear in a frequent concept lattice.
We show that the set of nodes of an *Alpha Galois Lattices* is a subset of the set of nodes of the corresponding concept lattice and that the values of α define a total order on *Alpha Galois lattices*. Finally, the inclusion of α-$extents$ corresponds to particular implication rules, representing some kind of approximation of usual implication rules (i.e. association rules with confidence 1), that depends on the *a priori* partition of the data. Such α-implication rules can be extracted in the same way that ordinary implication (and association) 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. 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). Finally, discussion and conclusion are given in section 5.

2 Preliminaries and definitions

Detailed definitions, results and proofs regarding Galois connections and lattices may be found in [1], and, in the framework of Formal Concept Analysis in [4] . However we need for our purpose a more general presentation than

the one in [4] as our *extents* are different from those used in *concept lattices* .

Definition 1 (Galois connection and Galois lattice) *Let m1: P → Q and m2: Q → P be maps between two lattices (P,$\leq_P$) and (Q,$\leq_Q$). (m1, m2) s 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 ⇒ m1(p2) $\leq_Q$ m1(p1)

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

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

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, then:

(G,$\leq$) is a lattice called a Galois lattice. When necessary it will be denoted as $G(P, m1, Q, m2)$

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.*
The example is represented in Figure 1 where each line is an individual and each column is an attribute. Together with $\mathcal{L}$ and $\mathcal{P}(I)$, int and ext define a Galois connection. We also 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 denoted as a concept lattice.

	t1	t2	t3	a3	a4	a5	a6	a7	a8
i1	1			1	1		1		1
i2	1			1		1	1		
i3		1			1		1		1
i4		1			1		1	1	
i5		1		1			1		1
i6			1	1			1		1
i7			1	1			1		1
i8			1	1		1	1		1

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

In concept lattices, a node (c, e) is a concept, c is the *intent* and e is the *extent* of the concept. A characteristic property of Galois lattices is that each node (c, e) is a pair of closed terms, so we have in particular $int(ext(c)) = c$, and c is the greatest term whose extent is $e = ext(c)$. So the intent c is a representative of the equivalence class of terms whose extent is e. We refer to the corresponding equivalence relation as $\equiv_{\mathcal{L}}$.

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 $int(ext(\{a4\}) = \{a4, a6\}$

3 Alpha Galois lattices

In this section we start with the concept lattice $G(\mathcal{L}, ext, \mathcal{P}(I), int)$ as previously examplified. Then we modify ext to obtain an equivalence relation $\equiv'_{\mathcal{L}}$ coarser than the original one. This results in larger equivalence classes on $\mathcal{L}$ and therefore in 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 [9]). 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 2 as a particular case of Alpha Galois lattice, and is much smaller than the original concept lattice (6 *vs*19 nodes).
Now, by relaxing the constraint that enforces to consider only whole basic classes we define *Alpha Galois lattices.*

Definition 2 (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* $\mid ext(T) \cap e \mid\ \geq \frac{|e|.\alpha}{100}$

We check now whether at least α % of a basic class satisfies a term of $\mathcal{L}$ and add this constraint to isa, the membership relation between individuals and terms:

Definition 3 (Alpha membership and Alpha extent) *Let $\mathcal{BC}$ be a partition of the set of individuals I as a set of basic classes. Let us denote as $BCl(i)$ 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* $BCl(i)\ sat_\alpha\ T$

The α-extent of T in I w.r.t. $\mathcal{BC}$ is then:

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

Example. *Let T={a6,a8}, ext(T) = {i1,i3,i5,i6,i7,i8}.* $BC2$

sat_{60} T *since* $\mid ext(T) \cap BC2 \mid \geq \frac{|BC2|.60}{100}$. *So we have* $i3$ *and* $i5$ isa_{60} T. *We also have* $BC3$ sat_{100} T, *as 100 % of* $BC3$ *belong to the extent of* T, *and so* $BC3$ sat_{60} T. *So we have* $i6$, $i7$, *and* $i8$ isa_{60} T *and* isa_{100} T. *As a result,* ext_0*(T)= ext(T) =* $\{i1, i3, i5, i6, i7, i8\}$, ext_{60}*(T)=* $\{i3, i5, i6, i7, i8\}$ *and* ext_{100}*(T)=* $\{i6, i7, i8\}$

We now define the corresponding *Alpha Galois lattices*:

Proposition 1 (Alpha Galois lattices) *Let* $E_\alpha = \{e \in \mathcal{P}(I) \mid \forall i \in e, \mid e \cap BCl(i) \mid \geq \frac{|BCl(i)|.\alpha}{100}\}$.
Then, int *and* ext_α *define a Galois connection on* $\mathcal{L}$ *and* E_α *and the corresponding Galois lattice* $G_\alpha = G(\mathcal{L}, ext_\alpha, E_\alpha, int)$ *is called an Alpha Galois lattice.*

When α is equal to 0, $E_\alpha = \mathcal{P}(I)$ and $ext_\alpha = ext$. Therefore, G_0 simply is is the corresponding concept lattice. The extents of the nodes of G_{100} are whole basic classes gathered and here the Alpha Galois lattice is the concept lattice obtained by considering as instances the *prototypes* of the basic classes (Figure 2).

Figure 3 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

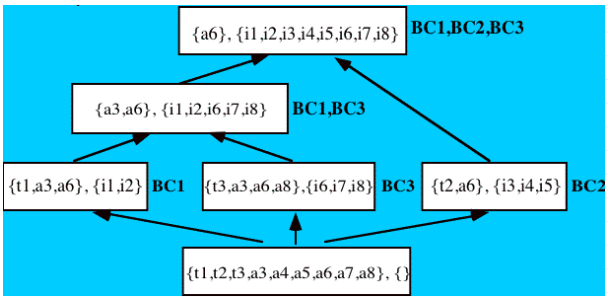

Figure 2. The G_{100} Alpha Galois lattice of example 1

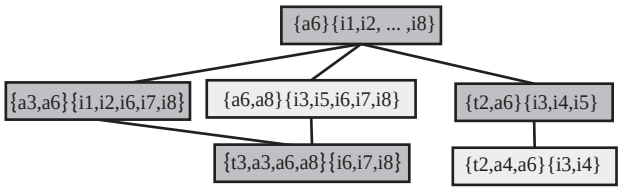

Figure 3. α = 60 : The topmost part of G_{60} of example 1. New nodes, w.r.t. G_{100} are the lighter ones.

In [3] the authors extend 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. [9] also introduces *extensional* projections that reduces the concept lattice by modifying the notion of *extent*. It is easy to show that there exists an extensional projection $proj_\alpha$ such that $E_\alpha = proj_\alpha(\mathcal{P}(\mathcal{I}))$ and that then ext_α=$proj_\alpha \circ ext$. Applying a theorem presented in [9] allows then to prove Proposition 1. By changing *ext*, an extensional projection changes a Galois lattice to a smaller one corresponding to larger equivalence classes on $\mathcal{L}$.

An interesting case is the one of the partition $\{I\}$ in which we consider only one basic 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* [10, 11] where $\frac{\alpha}{100}$ corresponds to the value of the support threshold *minsupp*.

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):

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

We are first interested in what happens with high values of α. Starting from G_{100}, new nodes appear as α slowly decreases. For instance at α = 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}. So, 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.

A second 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:

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

Here 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.

5 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 [4, 2, 6, 3]. 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 $\mathcal{P}(I)$ and we obtain a new family of Galois lattices. As mentioned above Iceberg (or frequent) concept lattices [11, 10] 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 $\alpha - implication$ rules can be extracted from the Alpha-Galois lattices in the same way as from frequent concept lattices (the $intents$ of the nodes are usually denoted as *closed frequent itemsets*. Association rules are then built using closed frequent itemsets [7, 12]). Note that $\alpha - implication$ rules inherit from the Galois lattice structure interesting properties (as transitivity) 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. [5]). Now there is another particular set theory view of *a priori* partitioned data referred to as *rough sets* theory ([8]). As the partitioning on rough sets expresses some indiscernibility between individuals of the same basic class, the rough sets view results in some degree of membership of i to an extent e, even if the individual i does not belong to e. At the contrary in the Alpha Galois lattice view, membership of i to an extent e is a prerequisite for α-membership. A fortunate consequence of the latter view is the opportunity to construct 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.

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] G. 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 S. O. Kuznetsov. Pattern structures and their projections. *ICCS-01, LNCS*, 2120:129–142, 2001.

[4] B. Ganter and R. Wille. *Formal Concept Analysis: Logical Foundations*. Springer Verlag., 1999.

[5] 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.

[6] M. Liquiere and J. Sallantin. Structural machine learning with galois lattice and graphs. In *ICML98, Morgan Kaufmann*, 1998.

[7] 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.

[8] Z. Pawlak. Rough sets, rough relations and rough functions. *Fundamenta Informaticae*, 27(2/3):103–108, 1996.

[9] 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.

[10] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with titanic. *Data and Knowledge Engineering*, 42(2):189–222, 2002.

[11] 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.

[12] M. J. Zaki. Generating non-redundant association rules. *Intl. Conf. on Knowledge Discovery and DataMining (KDD 2000)*, 2000.