ZooM : Alpha Galois Lattices for Conceptual Clustering

Nathalie Pernelle, Veronique Ventos
Université Paris-Sud
LRI, Bat. 490,
91405 Orsay, France
pernelle,ventos@lri.fr

Henry Soldano
Université Paris-Nord
LIPN, av. J.B. Clément
93430 Villetaneuse, France
soldano@lipn.univ-paris13.fr

Abstract

The main drawback of Galois lattices is that their size can be very large when dealing with real-world data sets. One way to reduce the number of nodes is to use a language with a weak expressivity. In this paper, we propose an approach based on the new notion of alpha extension where the number of nodes is reduced according to a change of the extensional function.

1. Introduction

Our work deals with the representation of typed and multi-valued data by clustering them in a small number of classes organized in a hierarchy and described at an appropriate level of abstraction. In our approach we propose a tool that enables the user build interactively various views of the data. More precisely, our basic representation is a Galois lattice, i.e. a lattice in which terms of a representational language are partitioned into equivalence classes w.r.t their extent (i.e. the set of instances that satisfy the term). Each node of a Galois lattice corresponds to a concept represented as its extension and its intension. The intension of a concept is the most specific term corresponding to the concept extension, and so is the representative of one equivalence class of terms. A Galois lattice is a well-defined and exhaustive representation of the concepts embedded in a data set. The main drawback of this structure is that the size of such a lattice can be very large when dealing with real-world data sets. Various techniques have been proposed to represent the Galois lattice and reduce its size by eliminating part of the nodes. Our work addresses this problem by taking into account the limitations of a user (what he can take in) and his wishes (what he is interested in).

The result is the system ZooM devoted to the construction of Galois lattices at different levels of abstraction. ZooM first builds a coarse lattice covering the whole set of instances and then refines a small subpart of the coarse lattice delimited by two nodes chosen by a user. In this system, the refinement can be done according to a change of language, but in this paper we only focus on the extensional refinement.

In this kind of refinement, the types of the data are used at different levels. The data are partitioned in basic classes according to their types. In the coarse lattice, we aggregate basic classes instead of aggregating instances. Then we partly relax this constraint in the following way : an instance belongs to the alpha extension of a term c when it belongs to the extension of c and when at least alpha % of the instances sharing the same type also belongs to the extension of c. When alpha = 0, a term then represents an unconstrained cluster of instances. This allows to deal either with some representational noise in instances (with high values of alpha) or with heterogeneous basic classes (using small values of alpha). The system has been implemented and some experimental evaluations has been made.

2. From extensions to Alpha extensions

The data that serve as instances of the classes that we build are described as a set of (Attribute, Value) pairs. The attributes used for describing the instances may vary from an instance to another. In addition, an attribute can be multi-valued. Furthermore, each instance is typed. For instance a product appearing in an electronic catalog could be typed as a removable disk drive. Thus, we can find a product whose description in this language $\mathcal{L}_1$ is:

$$\{Removablediskdrive, CD/DVD/Type{=}\{CDRW\}, StorageRemovableType{=}\{SuperDisk\}, Compatibility{=}\{MAC, PC\}\}$$

In the following, we will consider that the type t is a boolean attribute.

In this paper, for a sake of simplicity, we only consider the propositional language of classes $\mathcal{L}_2$ which has a high power of abstraction. Actually, this language will distinguish instances only if they have a different structure, that is if they are described using a different set of attributes. If this propositionnal language is used, the values are forgotten. See [1] for more details.

We propose to construct Galois lattices where the basic idea is the fact that terms of languages can be partitionned into equivalence classes w.r.t their extension (i.e. the set of instances which satisfy the term). A Galois lattice is then formed by the representative elements of these equivalence classes:

Definition 1 (Galois connection) *Let($\mathcal{L}$,$\preceq$) and (E, $\subseteq$) be two posets (i.e. sets with a partial ordering relation). Let int and ext be two correspondances defined on E and $\mathcal{L}$ by int: E$\rightarrow$ $\mathcal{L}$, ext: $\mathcal{L}$ $\rightarrow$E, and such that:*

C1- $\forall$(e,e1) $\in$ E, e$\subseteq$ e1 implies int(e) $\succeq$ int(e1)

C2- $\forall$(c,c1) $\in$ $\mathcal{L}$,c $\preceq$ c1 implies ext(c) $\supseteq$ ext(c1)

C3- $\forall$ c $\in$ $\mathcal{L}$,c $\preceq$ int(ext(c)), and $\forall$ e $\in$E, e $\subseteq$ ext(int(e))

Then (E,$\subseteq$,$\mathcal{L}$,$\preceq$,int,ext) define a Galois connection between ($\mathcal{L}$,$\preceq$) and (E,$\subseteq$).

Definition 2 (Galois lattice) *Let (E,$\subseteq$,$\mathcal{L}$,$\preceq$, int, ext) be a Galois connection.*
Let G={ (c,e) /c is a closed element of $\mathcal{L}$,and e is a closed element of E such that e=ext(c) (i.e. also c=(int(e))}
Let $\leq$ be defined by: (c,e) $\leq$ (c',e') iff e $\subseteq$ e' (which is equivalent to c $\succeq$ c').
(G,$\leq$) is called a Galois poset. Besides, if (G,$\leq$) is such that for all pair (x,y) of elements of G there exists one least upper bound (or supremum) and one greatest lower bound (or infimum), (G,int,ext) is called a Galois lattice.

Let us recall that the main goal of our approach is to reduce the complexity of the clustering step (i.e. the contruction task of galois lattices). The idea is to use the types that are associated to each data in order to aggregate types (i.e basic classes) instead of aggregating instances. The data are partitioned according to their type : for each type t , we create a *basic class* whose set of instances is the set of data of type t.

Example 1 *Let I = {i1,i2,i3,i4,i5,i6,i7,i8} be the set of instances such that their abstraction in $\mathcal{L}_2$ is represented in figure 1. Let us assume that the attributes $t1, t2, t3$ express the types of the instances.*
Then, we create a basic class BC1 = {i1,i2} whose intension is: {t1,a3,a6}, a basic class BC2 = {i3,i4,i5} whose intension is {t2,a6} and a basic class BC3 = {i6,i7,i8} whose intension is {t3,a3,a6,a8}.

The notions described below are used in order to define Galois lattices where nodes aggregate parts of basic classes which are bigger than a certain proportion α. The first one is the *alpha satisfaction* which allows us to know whether at least α % of a set of instances belong to the extension of a term.

	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. Instances description. $Tab(i,j)$ **= 1 if the** j^{th} **attribute belongs to the** i^{th} **example.**

Definition 3 (Alpha satisfaction) *Let α be a number belonging to [0,100]. Let e=$\{i_1,\ldots,i_n\}$ be a set of instances and T be a term of a class language $\mathcal{L}_2$.*
e alpha satisfies T (e sat_α T) iff $\mid ext_e(T) \mid \;\geq \frac{|e|.\alpha}{100}$

Then, we can add this constraint to the classical membership relation between instances and terms. We call this notion the *alpha membership relation*.

Definition 4 (Alpha membership relation) *Let I be a set of instances, $\mathcal{BC}$ be a finite set of disjoint basic classes such that $\mathcal{BC}$ represents a partition of I[1], and a function BCl: I $\rightarrow$ $\mathcal{BC}$ which returns from an instance the basic class of the instance. Let i be an instance description in $\mathcal{L}_1$. Let T be a $\mathcal{L}_2$ term.*

$i\ isa_\alpha\ T$ iff $i\ isa\ T$ and $BCl(i)\ sat_\alpha\ T$

Example 2 *Let I be the set of instances used in the example 1. Let T = {a6,a8}.*
$ext_I(T) = \{i1, i3, i5, i6, i7, i8\}$, $ext_{BC1}(T) = \{i1\}$, $ext_{BC2}(T) = \{i3, i5\}$, $ext_{BC3}(T) = \{i6, i7, i8\}$
$BC3\ sat_{100}\ T$ since 100 % of the instances of $BC3$ belongs to the extension of T. BC2 does not $sat_{100}\ T$. $BC2\ sat_{60}\ T$ since $\mid\ ext_{BC2}(T) \mid \;\geq \frac{|BC2|.60}{100}$.
Finally i3 isa_{60} T. This is not the case of i4 that also belongs to BC2.

Note that the *alpha membership relation* is equivalent to the classical *membership relation* when α is 0 and that when α is 100, an instance i belongs to a term T if and only if the whole basic class of i belongs to T.

1 This constraint of being a partition could be removed. If it is the case, we obtain a different definition of α membership relation that could be for instance "at least one basic class of the instance is alpha satisfied".

Finally, we use the *alpha membership relation* to define the following new notion of extension.

Definition 5 (Alpha extension of a term) *Let I be a set of instances, $\mathcal{BC}$ be a finite set of basic classes, and T a $\mathcal{L}_2$ term. The extension of T w.r.t I is the following set:*

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

Example 3 *In our example,* $ext(T)$=$\{i1, i3, i5, i6, i7, i8\}$, $ext^{60}(T)$=*{i3,i5,i6,i7,i8}*.

We have shown in [2] that ($E^{\alpha 2}$,$\subseteq$, $\mathcal{L}_2$, $\preceq$, int,ext^{α}) defines a Galois connection whose corresponding galois lattice G(int,ext^{α}) is nested in G(int,ext).
Figure 2 shows the Galois lattice G(int,ext^{100}) of example 1 and should be compared to the original Galois lattice G(int,ext) that contains 19 nodes.

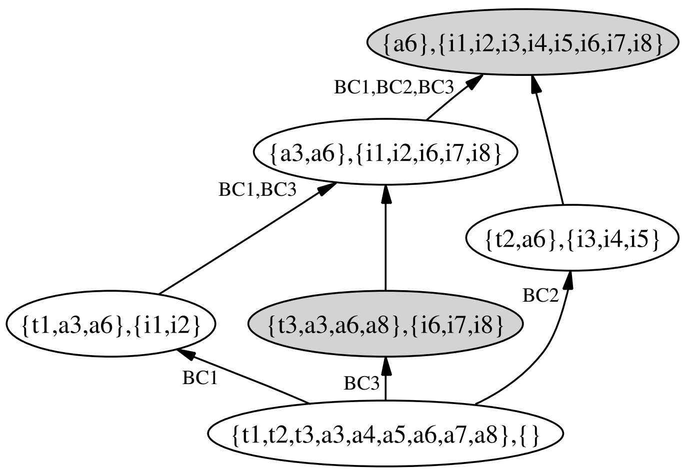

Figure 2. Galois lattice of fig.1 with ext^{100}

Finally, the following property order the Galois lattices associated to α-extensions : for every pair (α,α') such that $\alpha > \alpha'$: G(int,ext^{α}) is nested in G(int,$ext^{\alpha'}$). This order is used in ZooM to refine the coarse lattice G(int,ext^{100}). This lattice is locally refine by decreasing α to a lower α' thus decreasing the influence of the basic classes on the concept organization. Thus, if a refinement is made between the grey nodes {a6} and {t3,a3,a6,a8} of the figure 2, a new node {a6,a8} appears if alpha is equal to 60 (see figure 3). In this new node, a part bigger than 60 percent of BC2 is aggregated with BC3.

2 E^{α} is a subset of $\mathcal{P}(I)$ such that every element is a potential alpha extension w.r.t. its cardinality

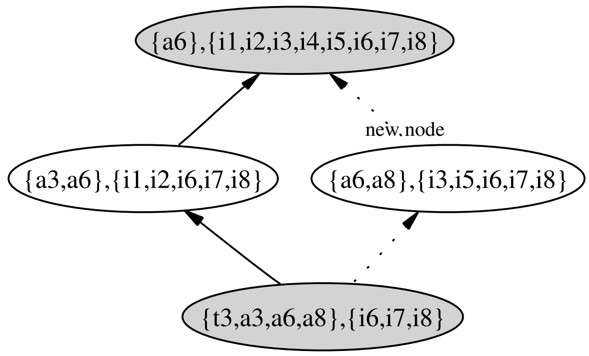

Figure 3. A local refinement with ext^{60}

Note that Frequent Galois lattices (i.e iceberg lattices [5, 3]) are particular cases of alpha galois lattices. Actually, an element (c, e) of a Galois lattice is in the frequent Galois lattice iff $\frac{|e|}{|I|}$ is greater or equal to a threshold s. We can easily see that this kind of Galois lattice is a alpha galois lattice where all the instances share the same type (i.e. there is only one basic class).

3. Conclusion and future works

Given a set of attributes, a set of instances, its partition as a set of basic classes, we have proposed as a conceptual representation a Galois lattice smaller than the usual concept lattice. The reduction corresponds to the influence of the partition and is controlled by a parameter alpha. When alpha is equal to 0, the alpha Galois lattice is the complete concept lattice and when alpha is equal to 100, only whole basic classes are gathered thus corresponding to the maximal influence of the initial partition. Finally, when alpha is near 100, it allows us to handle exceptions in the basic classes. A naive top-down algorithm devoted to the construction of alpha Galois lattices has been implemented and used on a real data set (computer products). We implement now an incremental version using the approach of [4].

References

[1] N. Pernelle, M.-C. Rousset, and V. Ventos. Automatic construction and refinement of a class hierarchy over multi-valued data. In *5th Conference on Principles and practice of Knowlewdge Discovery in Databases, PKDD'01, Lecture Notes in Artificial Intelligence*, pages 386–398. Springer-Verlag, 2001.

[2] N. Pernelle, H. Soldano, and V. Ventos. Zoom: a nested galois lattices-based system for conceptual clustering. *Journal of Experimental and Theoritical Artificial Intelligence*, 1(14):157–187, 2002.

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

[4] P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards constructing galois (concept) lattices. *Discrete Mathematics*, 256(3):801–829, 2002.

[5] K. Waiyamai and L. Lakhal. Knowledge discovery from very large databases using frequent concept lattices. In *11th European Conference on Machine Learning, ECML'2000*, pages 437–445, 2000.