Treillis dichotomiques et arbres de décision

Dichotomic Lattices and Decision Tree

K. Bertet , M. Visani et N. Girard

Laboratory L3I - University of La Rochelle - France
kbertet,mvisani,ngirard@univ-lr.fr

Manuscrit reçu le 15 décembre 2009

Résumé et mots clés

Dans ce papier, nous nous intéressons aux treillis dits *treillis dichotomiques*, définis à partir d'attributs binaires possédant une propriété de complémentarité par la borne supérieure. Il s'agit de la structure de treillis utilisée dans la méthode Navigala, méthode de reconnaissance de symboles basée sur un parcours (de type arbre de décision) dans le treillis. Nous mettons en évidence les liens structurels unissant les arbres de décision et les treillis dichotomiques en montrant tout d'abord que tout arbre de décision est inclus dans le treillis, mais également que le treillis est en fait la fusion de tous les arbres de décision. De ce lien de fusion nous déduisons un algorithme d'extraction d'un arbre de décision à partir d'un treillis dichotomique. Nous finissons par des expérimentations visant à comparer, pour de la reconnaissance de symboles, les performances des arbres de classification et des treillis construits avec la méthode Navigala.

Reconnaissance de symboles , classification supervisée, treillis, arbre de décision.

Abstract and key words

In this paper, we introduce a family of Galois lattice denoted as "dichotomic lattices". Such lattices are defined from binary attributes, where each binary attribute may be associated to a non-empty set of complementary attributes. In particular, lattices defined by binary attributes obtained after e discretisation pre-processing step are dichotomic. There are two types of classification methods using a Galois lattice: as most of them rely on selection, recent research work focus on navigation-based approaches. In navigation-oriented methods, classification is performed by navigating through the complete lattice, similar to the decision tree. The *Navigala* approach is a navigation-based classification method that relies on the use of a dichotomic lattice. It was initially proposed for symbol recognition, in the field of technical document image analysis. In this paper, we define the structural links between decision trees and dichotomic lattices defined from the same table of data described by binary attributes. Under this condition, we prove both that every decision tree is included in the dichotomic lattice and that the dichotomic lattice is the merger of all the decision trees that can be constructed from the same binary data table.

Symbol recognition, supervised classification, lattice, decision tree.

1. Introduction

Depuis une vingtaine d'années, les treillis de Galois sont largement utilisés en classification supervisée [14], où ils donnent des résultats comparables à des méthodes connues de la littérature : ID3, C4.5, …. Leur structure à base de graphe, nous a semblé pertinente pour une application dans le contexte de la reconnaissance de symboles bruités dans des documents techniques. Ainsi, nous avons développé une méthode de classification supervisée dédiée aux symboles, appelée Navigala [9].

Le treillis est un graphe dont les nœuds, appelés concepts, sont des regroupements maximaux d'objets possédant le même sous-ensemble maximal d'attributs. La plupart des méthodes de classification supervisées basées sur le treillis [13, 14, 24, 20, 15, 23] utilisent ce graphe pour sélectionner des concepts représentatifs d'une classe qui serviront ensuite de base à la classification.

La méthode Navigala se distingue : il s'agit non pas de sélectionner des concepts dans le treillis, mais d'utiliser la structure complète du graphe pour une navigation de type arbre de décision. Elle propose ainsi plusieurs chemins vers un même concept, donc vers une même classe, ce qui apporte en robustesse par rapport à l'arbre de décision dans le cadre d'une classification de symboles détériorés. Dans cette méthode, les treillis de Galois sont définis à partir d'attributs binaires issus d'un traitement de discrétisation d'une signature, ce qui leur confère des propriétés particulières. Plus précisément nous introduisons la définition de *treillis dichotomiques* pour les caractériser (*cf.* définition [1]) : un treillis est dit *dichotomique* lorsqu'il est défini pour une table où il est toujours possible d'associer à un attribut binaire x un ensemble non vide $\overline{X}$ d'attributs binaires (avec $x \notin \overline{X}$) tel que les attributs de $\{x\} \cup \overline{X}$ soient mutuellement exclusifs.

Dans ce papier nous montrons tout d'abord que les treillis dichotomiques possèdent la propriété de complémentarité par la borne supérieure [5] noté $\vee$-complémentarité (*ie*. à tout concept du treillis on peut associer un concept complémentaire pour la borne supérieure du treillis, voir section [3]).

Cette propriété induit des liens structurels forts entre les treillis dichotomiques et les arbres de décision : alors que pour un même ensemble de données discrètes, tout arbre de décision est inclus dans le treillis dichotomique (via un opérateur de fermeture), nous montrons que dans le cas des treillis dichotomiques, ce lien d'inclusion est renforcé par un lien de fusion : le treillis dichotomique est la fusion de tous les arbres de décision issus d'un même ensemble de données. Ce lien de fusion nous permet de proposer un algorithme d'extraction d'un arbre de décision à partir d'un treillis dichotomique.

En partie 2, nous décrivons les structures de l'arbre de décision et du treillis de Galois ainsi que les données manipulées, puis la méthode Navigala. La partie 3 s'intéresse aux treillis dichotomiques issus de données qui ont été discrétisées. Les liens structurels unissant les arbres de décision et les treillis dichotomiques sont étudiés et les preuves des différentes propositions sont établies. Nous finissons par une présentation des résultats expérimentaux de la méthode de classification Navigala que nous avons développée dans la partie 4.

2. Descriptions

2.1. Données

Lorsque des données s'organisent classiquement sous forme d'une table où les lignes correspondent aux objets, et les colonnes aux attributs, une distinction est à faire entre des attributs discrets ou qualitatifs, et des attributs continus ou quantitatifs. Les méthodes de classification et d'apprentissage proposées dans la littérature varient essentiellement en fonction de la nature des attributs qu'elles peuvent manipuler. Ainsi, une large panoplie de méthodes statistiques (SVM, réseaux de neurones...) existe pour des données continues alors que les données discrètes seront traitées par des classifieurs probabilistes (Règle du maximum de vraisemblance, Règle de Bayes, ...) ou encore symboliques (Arbres de décision, extraction de règles,...). La construction d'un outil de classification par apprentissage supervisé (quel que soit son type) nécessite une base d'apprentissage contenant conjointement les valeurs des attributs explicatifs sur lesquels repose la classification et de l'attribut à expliquer (attribut discret de classe).

Le treillis de Galois et l'arbre de décision peuvent s'utiliser dans un contexte de classification supervisée. Ces deux structures s'inscrivent parmi les méthodes symboliques, et se définissent naturellement à partir de données discrètes. Néanmoins, la discrétisation des attributs continus, opérée comme prétraitement dans le cas du treillis de Galois, ou en cours de la construction dans le cas des arbres de décision permet d'utiliser des attributs explicatifs continus, au prix évidemment d'une perte d'information que l'on cherche à limiter.

Dans le cas du treillis de Galois, il se définit à partir d'une table de données binaires, attributs à deux modalités. Toute variable discrète se code naturellement sous forme binaire via un codage disjonctif. L'intégration de données continues nécessite quant à elle un traitement de discrétisation en prétraitement.

2.2. Arbre de décision

Depuis les années 1960-1970, l'arbre de décision a fait l'objet de nombreux travaux de recherche [18, 19]. Les méthodes de génération de l'arbre de décision les plus connues sont ID3 [17], C4.5 [16] et CART [4].

Les nœuds de l'arbre sont construits depuis son sommet, appelé racine, vers sa base où les nœuds terminaux sont appelés feuilles. La construction de l'arbre de décision nécessite un critère de division pour sélectionner, à chaque étape de division de

l'arbre, un attribut de la table, un deuxième critère pour discrétiser les attributs continus si nécessaire, et enfin un critère d'arrêt des divisions généralement il s'agit d'une mesure de la pureté des feuilles qui intègre l'information de classe des objets qui la composent. Le nœud-racine considère l'ensemble des objets de la table ; un attribut de la table est alors sélectionné de manière à partitionner ces objets en deux sous-ensembles distincts formant ainsi deux nœuds fils (dans le cas binaire). Le processus est ensuite réitéré sur chacun des deux sous-ensembles, et ainsi de suite jusqu'à satisfaire le critère d'arrêt.

Lorsque les données sont continues, elles nécessitent une discrétisation qui peut être réalisée :

- soit pendant la construction de l'arbre avec le critère de discrétisation. Seuls les attributs sélectionnés pour la construction de l'arbre seront alors discrétisés.
- soit en prétraitement.

La figure [1] présente un exemple d'arbre de décision construit selon une mesure d'entropie à partir de la table [1] discrétisée. L'attribut A a été discrétisé en deux intervalles a1 = [0-3] et a2=[6-20] ; l'attribut B a été discrétisé en deux intervalles b1=[0-4] et b2=[12-20] et l'attribut C a été aussi discrétisé en deux intervalles c1=[0-2] et c2=[11-20]. Pour construire cet arbre nous aurions pu utiliser n'importe quel autre critère de division tel que la distance maximale ; le critère de Hotelling… De nombreuses heuristiques sont envisageables pour réaliser la construction d'un arbre de décision. Il est par exemple très courant d'effectuer un élagage de l'arbre de décision obtenu, de manière à éviter un sur-partitionnement des données. Le principe est de remonter à partir des feuilles de l'arbre en transformant certains nœuds de décision en feuilles selon un critère de pureté des nœuds. Dans la comparaison structurelle décrite par la suite, les arbres de décision considérés ne sont pas élagués.

L'étape de classification d'une nouvelle donnée consiste à faire parcourir à cette donnée l'arbre à partir de la racine, jusqu'à atteindre une feuille, par validation de la valeur de l'attribut proposé à chaque nœud. Le résultat de la classification sera l'information de classe portée par la feuille ainsi atteinte.

Tableau 1. Table de données discrétisées.

Classe	Id	A	B	C
1	1	[0-3]	[0-4]	[11-20]
	2	[0-3]	[0-4]	[11-20]
2	3	[0-3]	[12-20]	[11-20]
	4	[0-3]	[12-20]	[11-20]
	5	[0-3]	[12-20]	[11-20]
3	6	[6-20]	[12-20]	[11-20]
	7	[6-20]	[12-20]	[11-20]
	8	[6-20]	[12-20]	[11-20]
4	9	[6-20]	[0-4]	[0-2]
	10	[6-20]	[12-20]	[0-2]

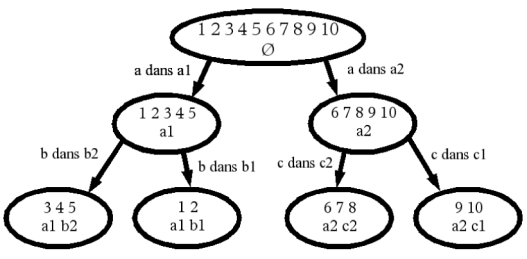

Figure 1. Arbre de décision associé à la table [1].

2.3. Treillis de Galois

Un *treillis de Galois* [2, 8] est défini à partir d'une table d'attributs binaires décrite par une relation binaire R entre un ensemble d'objets O et un ensemble d'attributs A. On associe à un sous-ensemble d'objets $X \subseteq O$ l'ensemble $f(X)$ des attributs en relation avec tous les objets de X ; duallement on associe à un sous-ensemble d'attributs $Y \subseteq A$, l'ensemble $g(Y)$ de tous les objets en relation avec les attributs de Y :

$$f(X) = \{a \in A \mid xRa \; \forall \; x \in X\}$$

$$g(Y) = \{o \in O \mid oRy \; \forall \; y \in y\}$$

Un *concept formel* est un sous-ensemble maximal d'objets possédant le même sous-ensemble d'attributs, lui-même maximal, défini formellement par un couple (X,Y) avec $X \subseteq O$ et $Y \subseteq A$, qui vérifie $f(X) = Y$ et $g(Y) = X$. On introduit alors la relation $\leqslant$ définie sur l'ensemble de tous les concepts formels de la manière suivante. Pour deux concepts formels quelconques (X_1,Y_1) et (X_2,Y_2) on a $(X_1,Y_1) \leqslant (X_2,Y_2)$ si et seulement si $X_1 \supseteq X_2$, ou de façon équivalente $Y_1 \subseteq Y_2$.

La relation $\leqslant$ possède les propriétés d'une relation d'ordre, *i.e.* une relation transitive, antisymétrique et réflexive. Les propriétés d'une relation d'ordre permettent de considérer l'ensemble de tous les successeurs et de tous les prédécesseurs d'un concept selon $\leqslant$. On peut également introduire les successeurs immédiats et prédécesseurs immédiats en considérant la relation de couverture de $\leqslant$ notée $\prec$.

L'ensemble de tous les concepts formels équipé de la relation d'ordre $\leqslant$ est appelé *treillis des concepts* ou encore *treillis de Galois* car il possède la propriété de treillis: pour tous concepts (X_1,Y_1) et (X_2,Y_2), il existe un unique plus grand successeur (resp. plus petit prédécesseur) appelé *borne inférieure* (resp. *borne supérieure*) noté $(X_1,Y_1) \wedge (X_2,Y_2)$ (resp. $(X_1,Y_1) \vee (X_2,Y_2)$) défini par :

$$(X_1,Y_1) \wedge (X_2,Y_2) = (g(Y_1 \cap Y_2),(Y_1 \cap Y_2)) \quad (1)$$

$$(X_1,Y_1) \vee (X_2,Y_2) = ((X_1 \cap X_2), f(X_1 \cap X_2)) \quad (2)$$

Cette propriété de treillis implique l'existence d'un unique plus petit élément $\perp = (O, f(O))$, et un unique plus grand élément $\top = (g(A), A)$. La figure [2] présente un exemple de treillis de

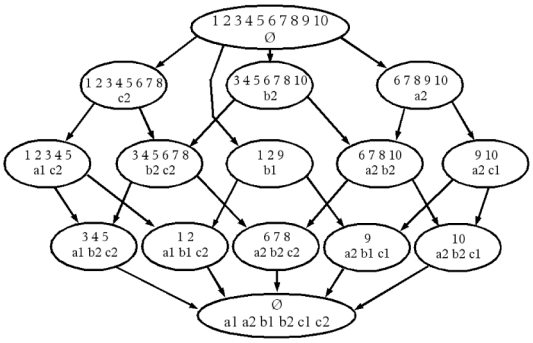

Figure 2. Treillis de Galois.

Tableau 2. Table binaire.

Id	Intervalles					
	a_1 [0-3]	a_2 [6-20]	b_1 [0-4]	b_2 [12-20]	c_1 [0-2]	c_2 [11-20]
1	X		X			X
2	X		X			X
3	X			X		X
4	X			X		X
5	X			X		X
6		X		X		X
7		X		X		X
8		X		X		X
9		X	X		X	
10		X		X	X	

Galois construit pour un ensemble de 10 objets décrits par 6 attributs explicatifs ($a1$, $a2$, $b1$, $b2$, $c1$ et $c2$) et un attribut classe, représentés dans la table [2].

Remarque. Dans l'usage habituel, $(X_1,Y_1) \leq (X_2,Y_2)$ si et seulement si $X_1 \subseteq X_2$. Nous utilisons ici la relation inverse avec un raffinement sur les attributs ($Y_1 \subseteq Y_2$) plutôt que sur les objets par similitude à la fois avec la définition de l'arbre de décision, mais aussi avec le treillis des fermés défini sur une famille d'attributs (appelés fermés) par la relation d'inclusion entre fermés. Par conséquent, $\perp = (O, f(O))$ et $\perp = (g(A), A)$. De même, toujours par similitude avec les arbres de décision, l'élément minimal $\perp$ est représenté en haut des dessins comme l'est la racine d'un AD.

Les deux fonctions f et g définies entre objets et attributs forment une *correspondance de Galois*. La composition $\varphi = f \circ g$, définie sur la famille des attributs, permet d'associer à un sous-ensemble d'attributs $Y \subseteq A$ le plus petit concept contenant Y : $(g(\varphi(Y)),\varphi(Y))$. Cette composition φ possède les propriétés d'un opérateur de fermeture : φ est idempotent (*i.e.* $\forall Y \subseteq A$, $\varphi^2(Y) = \varphi(Y)$), extensif (*i.e.* $\forall Y \subseteq A$, $Y \subseteq \varphi(Y)$) et isotone (*i.e.* $\forall Y, Y' \subseteq A$, $Y \subseteq Y' \Rightarrow \varphi(Y) \subseteq \varphi(Y')$).

L'utilisation d'un treillis de Galois en analyse de données, faisant l'objet d'un domaine appelé Analyse Formelle des Concepts (AFC) [8], est en plein essor ces dernières années. En effet, l'opérateur de fermeture φ porte une relation de corrélation entre un sous-ensemble d'attributs Y, et les attributs de $\varphi(Y)\backslash Y$; le concept $(g(\varphi(Y)),\varphi(Y))$ est une représentation ensembliste de cette corrélation, et le treillis une représentation de tous les sous-ensembles possibles d'attributs ainsi corrélés et ordonnés par inclusion. Il s'en suit que des données fortement corrélées engendreront peu de concepts, alors que des données très peu corrélées pourront engendrer jusqu'à $2^{|A|}$ concepts.

En AFC, la terminologie d'usage est celle de treillis des concepts. Une table binaire y est appelée un *contexte*, et une table discrète, un *contexte multivalué*. Un contexte multivalué y est défini par un quadruplet (O,A,W,R) avec O l'ensemble des objets, A l'ensemble des attributs, W l'ensemble des modalités des attributs et R une relation ternaire entre O, A et W. $(x,y,w) \in R$, que l'on peut également écrire $y(x) = w$, signifie que l'attribut y a la valeur w pour l'objet x.

Pour plus d'informations sur le treillis de Galois ou treillis des concepts, et les opérateurs de fermeture, le lecteur peut se reporter aux références [2, 8].

2.4. Navigala : méthode de classification de symboles par navigation dans un treillis de Galois

La plupart des méthodes de classification basées sur le treillis [13, 14, 24, 20, 15, 23] utilisent ce graphe pour sélectionner certains concepts représentatifs des classes qui serviront ensuite de base à la classification. La méthode Navigala se distingue [9] : il s'agit non pas de sélectionner des concepts dans le treillis, mais d'utiliser la structure complète du graphe pour une navigation de type arbre de décision.

La méthode Navigala a été conçue pour reconnaître des symboles issus de documents techniques. À partir des images de symboles, nous extrayons des signatures (vecteurs de caractéristiques) composées d'attributs continus. Nous avons implémenté trois signatures statistiques (la signature de Radon [21], la signature de Fourier-Mellin [7] et la signature de Zernike [22]) que nous avons comparées dans [9]. Nous avons également développé une signature structurelle dédiée aux symboles [6]. Cette signature est composée du nombre d'occurrences de chemins dans un graphe topologique qui décrit les relations entre les segments préalablement extraits du symbole, alors que les signatures statistiques décrivent la répartition spatiales des pixels.

La méthode Navigala intègre les deux étapes classiques pour réaliser le processus de reconnaissance que sont l'apprentissage et la classification. Les données d'apprentissage sont tout d'abord discrétisées selon un critère de coupe supervisé (le critère de Hotelling [10]), de manière à obtenir une table binaire. Les attributs de cette table sont des intervalles de valeurs, et les objets sont les symboles. La discrétisation se termine lorsqu'il y a séparation des classes, c'est-à-dire lorsque chaque classe se distingue des autres par au moins un des attributs de la table. La table binaire discrétisée est ensuite réduite : il s'agit de suppri-

Tableau 3. Discrétisation d'une table continue.

Id	Attributs				C
	a	*b*	*c*	*d*	
1	1	4	15	14	1
2	0	0	18	14	
3	1	12	13	14	2
4	0	16	15	14	
5	3	12	11	14	
6	8	16	15	14	3
7	6	20	20	14	
8	15	12	15	14	
9	18	4	0	14	4
10	20	12	2	14	

1- Table continue

Id	Intervalles				C
	a	*b*	*c*	*d*	
1	[0-3]	[0-4]	[11-20]	[14 14]	1
2	[0-3]	[0-4]	[11-20]	[14 14]	
3	[0-3]	[12-20]	[11-20]	[14 14]	2
4	[0-3]	[12-20]	[11-20]	[14 14]	
5	[0-3]	[12-20]	[11-20]	[14 14]	
6	[6-20]	[12-20]	[11-20]	[14 14]	3
7	[6-20]	[12-20]	[11-20]	[14 14]	
8	[6-20]	[12-20]	[11-20]	[14 14]	
9	[6-20]	[0-4]	[0-2]	[14 14]	4
10	[6-20]	[12-20]	[0-2]	[14 14]	

2- Table discrétisée

Id	Intervalles			C
	a	*b*	*c*	
1	[0-3]	[0-4]	[11-20]	1
2	[0-3]	[0-4]	[11-20]	
3	[0-3]	[12-20]	[11-20]	2
4	[0-3]	[12-20]	[11-20]	
5	[0-3]	[12-20]	[11-20]	
6	[6-20]	[12-20]	[11-20]	3
7	[6-20]	[12-20]	[11-20]	
8	[6-20]	[12-20]	[11-20]	
9	[6-20]	[0-4]	[0-2]	4
10	[6-20]	[12-20]	[0-2]	

3- Table réduite

Id	Intervalles						C
	a_1 [0-3]	a_2 [6-20]	b_1 [0-4]	b_2 [12-20]	c_1 [0-2]	c_2 [11-20]	
1	X		X			X	1
2	X		X			X	
3	X			X		X	2
4	X			X		X	
5	X			X		X	
6		X		X		X	3
7		X		X		X	
8		X		X		X	
9		X	X		X		4
10		X		X	X		

4- Table réduite binarisée

mer les attributs non discrétisés (*i.e.* possédant un seul intervalle). Cette réduction peut être vue comme une étape de sélection des attributs explicatifs. À partir de la table binaire discrétisée et réduite, la construction du treillis de Galois ne nécessite aucun paramètre et permet l'obtention d'une unique structure de graphe. La table étant réduite, le min et le max du treillis sont respectivement $\perp = (\mathrm{O}, \emptyset)$ et $\top = (\emptyset, \mathrm{A})$; cette réduction empêche d'avoir des attributs redondants. Lorsque deux objets de classes différentes ont la même signature (*ie.* faux positif), un critère d'arrêt reposant sur l'absence de création de nouveaux intervalles permet de stopper la discrétisation. La table [3] présente les étapes de discrétisation, réduction et binarisation d'une table de données continues contenant 10 objets ; 4 attributs a, b, c, d et le label de classe C de chaque objet.

L'étape de classification de nouveaux symboles consiste ensuite à naviguer dans le treillis de Galois à la manière d'une navigation dans l'arbre de décision. Plus précisément, il s'agit, à partir du concept minimal[1], de progresser pas-à-pas d'un concept vers un successeur immédiat selon un critère de choix paramétrable par l'utilisateur. Ce critère permet pour le concept courant de sélectionner un concept parmi ses successeurs en fonction d'une mesure de distance entre les attributs du symbole à reconnaître et les intervalles proposés par les concepts successeurs. Trois critères déterministes sont proposés dans Navigala [9]. La navigation se poursuit jusqu'à atteindre un concept final. Ce concept final permet d'attribuer une classe au nouveau symbole à classer, puisqu'il contient un ensemble d'objets appartenant tous à la même classe.

1. Le concept minimal d'un treillis de Galois est le concept noté $\perp = (\mathrm{O}, \emptyset)$.

3. Treillis dichotomiques

3.1. Définitions

Les *treillis dichotomiques* que nous introduisons dans ce papier se définissent par une propriété portée par la table :

Définition 1. *Un treillis est dit dichotomique lorsqu'il est défini pour une table réduite où il est toujours possible d'associer à un attribut binaire y un ensemble non vide $\overline{Y}$ d'attributs binaires (avec $y \notin \overline{Y}$) tel que les attributs de $\{y\} \cup \overline{Y}$ soient mutuellement exclusifs.*

Au sein de la méthode Navigala, les données d'apprentissage sont continues et discrétisées en plusieurs intervalles disjoints. Ces intervalles sont donc mutuellement exclusifs dans la table, et les treillis générés par Navigala sont dichotomiques.

La propriété principale des treillis dichotomiques est d'être $\vee$-pseudo-complémentés, c'est-à-dire que pour tout concept (X, Y), il existe toujours un *concept complémentaire* (X', Y') tel que :

$$(X, Y) \vee (X', Y') = \top = (\emptyset, A) \tag{3}$$

Proposition 1. *Les treillis dichotomiques sont $\vee$-pseudo-complémentés*

Preuve. Pour montrer que tout treillis dichomique est $\vee$-pseudo-complémenté, considérons un concept quelconque (X,Y) d'un treillis dichotomique. Il s'agit de montrer l'existence d'un concept complémentaire à (X,Y). Considérons un attribut binaire quelconque y de Y, et $\overline{y}$ un attribut complémentaire de y appartenant à l'ensemble $\overline{Y}$. Il s'en suit que les objets possédant y, et ceux possédant $\overline{y}$ sont différents, ce qui se formalise par $g(\{y\}) \cap g(\{\overline{y}\}) = \emptyset$. On considère alors le plus petit concept contenant $\overline{y}$ qui, par définition, sera le concept $(g(\varphi(\{\overline{y}\})),\varphi(\{\overline{y}\}))$ dont l'ensemble des attributs est $\varphi(\{\overline{y}\})$. On déduit de la définition des fonctions f et g que $g(\varphi(\{\overline{y}\})) = g(\{\overline{y}\})$, et que $X \subseteq g(\{y\})$. Étant donné que $g(\{y\}) \cap g(\{\overline{y}\}) = \emptyset$, on peut alors en déduire que $X \cap g(\varphi(\{\overline{y}\})) = \emptyset$. Par conséquent, $(X,Y) \vee (g(\varphi(\{\overline{y}\})),\varphi(\{\overline{y}\})) = (\emptyset,A)$, et le concept $(g(\varphi(\{\overline{y}\})), \varphi(\{\overline{y}\}))$ est un concept complément de (X,Y), ce qui prouve la $\vee$-pseudo-complémentarité du treillis. ❑

Une *table dichotomique* est définie dans [8] par l'existence d'un unique attribut complémentaire $\overline{y}$ pour tout attribut binaire y. La propriété d'unicité étant ici ajoutée à celle de complémentarité, ceci signifie que tout objet possède soit l'attribut y, soit son complémentaire $\overline{y}$. Ainsi, la table [2] est une table dichotomique.
Il apparaît clairement que tout treillis issu d'une table dichotomique est un treillis dichotomique, alors que l'inverse n'est pas toujours vrai. Plus précisément, ces treillis sont $\vee$-complémentés, *i.e.* ils sont $\vee$-pseudo-complémentés et vérifient en plus la propriété d'unicité du concept complémentaire : à tout concept on associe un unique concept complémentaire. La $\vee$-complémentarité est donc une restriction de la $\vee$-pseudo-complémentarité, et nous pouvons établir le corollaire suivant de la proposition 1 :

Corollaire 1. *Les treillis issus d'une table dichotomique sont $\vee$-complémentés.*

Il est possible de rendre une table dichotomique en ajoutant tous les attributs complémentaires manquants. Le nombre d'attributs dans la table peut alors doubler et le nombre de concepts du treillis peut croître de façon exponentielle.
Une comparaison structurelle [11, 12] entre le treillis issu d'une table dichotomique et tout arbre de décision issu de cette même table, les arbres variant selon le critère de segmentation, a permis d'établir le résultat suivant :

- toute branche maximale[2] de l'arbre correspond à une chaîne maximale du treillis.
- toute chaîne maximale[3] du treillis correspond à une branche maximale de l'arbre.

Dans ce papier, nous montrons que ce lien structurel fort entre arbre de décision et treillis issu d'une table dichotomique est maintenu pour les treillis dichotomiques. D'où une extension de ce résultat ainsi que ses retombées dans un contexte de classification utilisant des attributs explicatifs continus comme par exemple Navigala.
Notre approche est néanmoins différente de celle de [11, 12]. Nous établissons les liens structurels entre arbre de décision et treillis dichotomiques de la façon suivante :

- Tout arbre est inclus dans le treillis dichotomique
- Le treillis dichotomique est la fusion de tous les arbres (variant selon le critère de division)

3.2. L'arbre est inclus dans le treillis

L'inclusion de l'arbre dans le treillis est vérifiée dans le cas général et non pour les seuls treillis dichotomiques. C'est une conséquence immédiate de la propriété de fermeture d'un treillis.
Nous associons à chaque nœud de l'arbre de décision un unique concept dans le treillis en utilisant l'opérateur de fermeture. Considérons un nœud n de l'arbre, et l'ensemble des attributs binaires Y_n proposés depuis la racine jusqu'à ce nœud. On associe alors au nœud n le plus petit concept contenant les variables de Y_n :

$$(g(\varphi(Y_n)),\varphi(Y_n)) \tag{4}$$

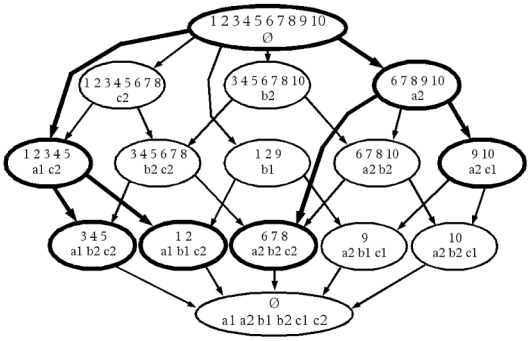

Figure 3. Inclusion de l'arbre de décision (en gras) dans le treillis de Galois.

La figure 1 représente l'arbre de décision associé aux données de l'exemple. Comme le montre la figure 3 l'arbre (en gras) est inclus dans le treillis ; on peut noter que tous les nœuds de l'arbre de décision sont présents dans le treillis via une opération de fermeture et les branches de l'arbre se retrouvent dans le treillis par transitivité. Cette propriété se vérifie dans le cas général.

Proposition 2. *Tout arbre de décision est inclus dans le treillis dichotomique, lorsque ces deux structures sont construites à partir des mêmes attributs binaires.*

Preuve. Considérons un arbre de décision ainsi que le treillis dichotomique issu des mêmes attributs binaires. Comme men-

2. Une branche maximale est une section de l'arbre partant de la racine allant jusqu'à une feuille et contenant tous les nœuds intermédiaires.

3. Une chaine maximale est une chaîne du diagramme de Hasse du treillis inclue dans aucune autre donc partant du concept minimal allant jusqu'au concept maximal et contenant tous les concepts intermédiaires.

tionné ci-dessus, à tout nœud n de l'arbre de décision accessible par validation de l'ensemble d'attributs Y_n on associe le concept $(g(\varphi(Y_n)), \varphi(Y_n))$. Pour montrer que l'arbre de décision est inclus dans le treillis, il s'agit alors de prouver les trois points suivants :

1. *Deux nœuds différents d'un arbre de décision sont associés à des concepts différents :*
   Par l'absurde, si deux nœuds n_1 et n_2 sont associés au même concept, alors $\varphi(Y_{n_1}) = \varphi(Y_{n_2})$. Ceci signifie que ce sont les même objets qui partagent les attributs de Y_{n_1} et Y_{n_2}, ce qui est en contradiction avec le fait que les deux nœuds n_1 et n_2 soient deux nœuds différents de l'arbre de décision.
2. *Si deux nœuds sont ancêtres dans l'arbre de décision alors leurs concepts associés sont en relation dans le treillis :*
   Il est clair que si un nœud n_1 est ancêtre d'un nœud n_2 dans l'arbre de décision, alors $Y_{n_1} \subseteq Y_{n_2}$. L'opérateur φ étant isotone, on en déduit que $\varphi(Y_{n_1}) \subseteq \varphi(Y_{n_2})$, et par conséquent que les deux concepts $(g(\varphi(Y_{n_1})),\varphi(Y_{n_1}))$ et $(g(\varphi(Y_{n_2})),\varphi(Y_{n_2}))$ sont en relation selon $\leqslant$.
3. *À l'inverse, si deux nœuds ne sont pas ancêtres dans l'arbre de décision alors leurs concepts associés ne sont pas en relation dans le treillis :*
   Si un nœud n_1 n'est pas ancêtre d'un nœud n_2, il s'agit alors de considérer les fils du plus petit ancêtre commun à n_1 et n_2, et en particulier le fils n'_1 ancêtre de n_1 et le fils n'_2 ancêtre de n_2. Ces deux nœuds n'_1 et n'_2 existent par construction de la table et il est clair que, n'_1 et n'_2 étant frères aucun objet ne peut avoir à la fois les attributs des concepts associés, à savoir $\varphi(Y_{n'_1})$ et $\varphi(Y_{n'_1})$. Ce qui se formalise par $g(\varphi(Y_{n'_1})) \cap g(\varphi(Y_{n'_2})) = \emptyset$. Ensuite, n'_1 étant ancêtre de n_1, on en déduit que $Y_{n'_1} \subseteq Y_{n_1}$, d'où $\varphi(Y_{n'_1}) \subseteq \varphi(Y_{n_1})$ par isotonie de l'opérateur φ, et à l'inverse $g(\varphi(Y_{n'_1})) \supseteq g(\varphi(Y_{n_1}))$ par définition de g. On aura de même $g(\varphi(Y_{n'_2})) \supseteq g(\varphi(Y_{n_2}))$ car n'_2 est ancêtre de n_2. On en déduit alors que $g(\varphi(Y_{n_2})) \cap g(\varphi(Y_{n_1})) = \emptyset$, ce qui prouve que les concepts associés aux nœuds n_1 et n_2 ne sont pas en relation selon $\leqslant$. ❑

3.3. Le treillis est la fusion de tous les arbres

La propriété de fusion se déduit quant à elle de la $\vee$-pseudo-complémentarité. Elle est par conséquent propre aux treillis dichotomiques et renforce ainsi les liens structurels entre treillis dichotomiques et arbres de décision. Nous en donnons une preuve constructive qui montre que tout sous-arbre d'un treillis de Galois est un arbre de décision (*ie*. qui sépare les objets de classes différentes). De cette preuve se déduit un algorithme d'extraction d'un arbre de décision à partir d'un treillis dichotomique.

Proposition 3. *Un treillis dichotomique est la fusion de tous les arbres de décision lorsque ces structures sont construites à partir des mêmes attributs binaires.*

Preuve. Nous avons déjà montré que tout arbre de décision était inclus dans le treillis dichotomique construit à partir des mêmes attributs binaires. Pour prouver que le treillis dichotomique est en fait la fusion de tous les arbres de décision, il suffit de montrer que tout concept est susceptible d'appartenir à un arbre de décision, ce que nous allons faire par construction.
Considérons un concept quelconque (X,Y). On construit alors le sous-ensemble C de concepts du treillis contenant : le concept (X,Y), un concept (X',Y') complémentaire à (X,Y), le concept minimal $\perp$, et tous les concepts finaux successeurs de (X,Y) et de (X',Y'). L'existence du concept complémentaire (X',Y') se déduit de la propriété de $\vee$-complémentarité du treillis dichotomique. De plus, elle induit que ce sous-ensemble C équipé de la relation $\leqslant$ forme un arbre. On ajoute alors dans l'ensemble C un nombre maximum de concepts du treillis dichotomique de telle sorte que la borne supérieure de deux éléments non comparables de l'ordre partiel $(C,\leqslant)$ soit le top $\top$. Nous obtenons ainsi par construction un sous-arbre inclus dans le treillis dichotomique, contenant (X,Y), et dont les feuilles, qui sont des concepts finaux, correspondent à des sous-ensembles d'objets qu'aucun attribut binaire ne peut séparer, *i.e*. des classes lorsque les données ont été discrétisées jusqu'à séparation des classes. Cet arbre peut donc être considéré comme un arbre de décision, ce qui termine cette preuve. ❑

Notons que le choix d'un seul concept parmi les concepts complémentaires engendre un arbre de décision non complet car certaines classes peuvent ne pas y apparaître. L'obtention d'un arbre de décision complet est cependant possible si plusieurs concepts sont choisis. L'arbre de décision est alors complet et non binaire.
L'algorithme `ExtraireArbre` permettant d'extraire un arbre de décision à partir d'un concept du treillis se déduit naturellement de cette preuve par construction. Initialement appelé pour le concept minimal du treillis, cet algorithme prend également en paramètre l'arité de l'arbre à générer, un critère de sélection d'un nœud parmi un ensemble de concepts ainsi qu'un critère d'arrêt précisant s'il s'agit ou non d'une feuille (souvent la pureté du concept vis-à-vis de l'attribut à expliquer). L'arbre ainsi généré sera défini à partir des mêmes attributs binaires que le treillis.
Différents critères de sélection sont envisageables permettant d'intégrer à la fois des informations statistiques (concept le plus hétérogène,....), mais aussi structurelles (concept dont le sous-arbre est de hauteur /largeur minimale/maximale). La table 4 présente le déroulement de l'algorithme d'extraction d'un arbre appliqué au treillis de la figure 3, cet algorithme permettant d'obtenir l'arbre représenté en gras sur cette même figure. Les critères de choix du concept successeur et du concept complémentaire sont identiques : choix du concept contenant le moins d'objets. L'algorithme débute sur le concept min $\perp$ et s'arrête lorsque les concepts successeurs ne contiennent que des objets de la même classe.
Dans une étude récente [3], un arbre de hauteur maximale est extrait à partir d'un treillis quelconque. Ses taux de reconnais-

sance améliorent ceux des arbres de décision générés par CART ou C4.5 sur plusieurs bases, offrant ainsi des perspectives intéressantes.

Nom : `ExtraireArbre`
Données : un treillis de Galois T ; un concept (X, Y) du treillis ; un critère de sélection Cr d'un nœud ; l'arité n de l'arbre à générer ; un critère d'arrêt S
Résultat : un arbre d'arité n inclus dans le sous-treillis issu de (X, Y)
début
  Initialiser un arbre de décision de racine r;
  si *le concept* (X, Y) *ne vérifie pas le critère d'arrêt* S **alors**
    Calculer l'ensemble $Succ$ des successeurs immédiats du concept (X, Y) dans le treillis ;
    Choisir un concept (X', Y') dans l'ensemble $Succ$ selon le critère Cr ;
    Initialiser avec l'arbre résultat de `ExtraireArbre` $(T, (X', Y'), \text{Cr}, n, S)$ l'ensemble des fils de r avec comme proposition les attributs de $Y' \setminus Y$;
    Calculer l'ensemble $Compl$ des concepts complémentaires à (X', Y') dans le treillis ;
    pour *i allant de 1 à n* **faire**
      Extraire un concept (X_i, Y_i) de l'ensemble $Compl$ selon le critère Cr ;
      Ajouter l'arbre résultat de `ExtraireArbre` $(T, (X_i, Y_i), \text{Cr}, n, S)$ à l'ensemble des fils de r, avec comme proposition les attributs de $Y_i \setminus Y$;
  Retourner l'arbre de décision de racine r;
fin

4. Expérimentation

L'ensemble des expérimentations présentées ont été réalisées avec la méthode Navigala sur la base de symboles GREC 2003 [1] (voir figure 4) qui sont des symboles issus de plans architecturaux.

Nous avons tout d'abord comparé les taux de reconnaissance (figure 5) et les tailles des structures générées (figure 6) par l'arbre de décision (CART [4]) et le treillis de Galois, sur un extrait de 10 classes de la base GREC 2003. La base d'apprentissage était composée de 10 symboles et la base de test de 900 symboles. Les trois signatures statistiques : invariants de Fourier-Mellin [7], R-signature (Radon) [21] et moments de Zernike [22], ont été étudiées pour ces tests. Il s'avère que pour chacune des signatures, la reconnaissance par le treillis de Galois est plus efficace que celle utilisant l'arbre de décision. La taille de l'arbre de décision est cependant plus condensée que celle du treillis de Galois (figure 6). En effet, la taille d'un arbre est polynomiale en la taille des données alors que celle d'un treillis est exponentielle dans le pire des cas, mais reste polynomiale en pratique dans les nombreuses expérimentations qui en ont été faites [14]. À la différence de l'arbre, le treillis propose plusieurs chemins de classification ou de reconnaissance, ce qui lui apporte une certaine robustesse lorsqu'il s'agit de reconnaître ou classifier des données détériorées.

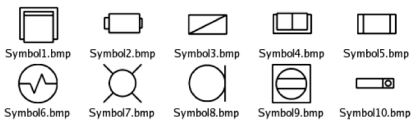

Figure 4. Exemples de symboles de la base GREC 2003.

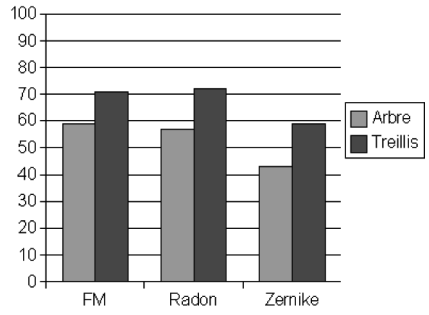

Figure 5. Comparaison des taux de reconnaissance obtenus par l'arbre de décision et le treillis de Galois.

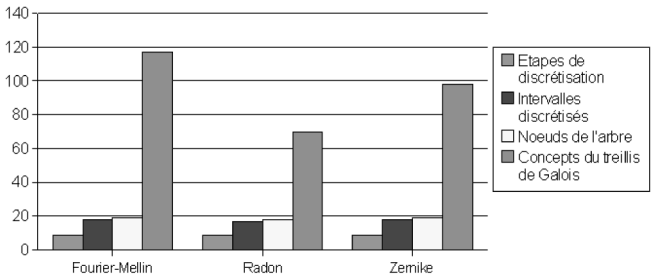

Figure 6. Complexités des structures.

Tableau 5. Comparaison génération du treillis entier/ à la demande.

	Apprent.	Classif.	Nb. de concepts
Treillis entier	430,2 sec	2 sec	3185
Génération à la demande	0,5 sec	9,8 sec	282

Tableau 4. Déroulement de l'algorithme `ExtraireArbre` *sur le treillis de la figure 3.*

Classes	(X,Y)	successeurs choisis (X',Y')	complémentaires de (X',Y') : (X_i, Y_i)
	$\perp = (O,\emptyset)$	$(6,7,8,910; a_2)$	$(1,2,3,4,5; a_1c_2)$
Plusieurs classes par concept	$(6,7,8,9,10; a_2)$ $(1,2,3,4,5; a_1c_2)$	$(9,10; a_2c_2)$ $(1,2; a_1b_1c_2)$	$(6,7,8; a_2b_2c_2)$ $(3,4,5; a_1b_2c_2)$
Une seule classe par concept	$(9,10; a_2c_2)$ $(1,2; a_1b_1c_2)$	fin de l'algorithme	

Pour remédier aux problèmes de la taille du treillis de Galois, nous avons réalisé une extension de l'algorithme de construction du treillis de Galois afin de ne générer à la demande que les concepts nécessaires à la navigation dans le graphe. Ainsi l'expérimentation suivante (Tab. 5), réalisée sur un extrait de 25 classes de la base GREC 2003, témoigne de l'apport d'une telle extension. La base d'apprentissage était composée de 25 symboles et la base de test de 10 symboles. Le nombre de concepts générés à la demande (282 concepts) est bien moins important que la construction du treillis entier (3185 concepts), et ce en effectuant le même parcours de reconnaissance dans le treillis. Les résultats de classification obtenus avec et sans génération à la demande sont donc identiques. La génération à la demande présente donc l'intérêt de garantir l'obtention de taux de reconnaissance identiques à ceux du treillis entier, tout en réduisant la taille de la structure à générer. On remarque que la méthode de génération à la demande augmente le temps nécessaire à la classification du fait que le treillis est généré selon la phase de classification.

Dans cette dernière expérimentation, nous présentons les taux de classification obtenus par le treillis de Galois à partir de la signature de Radon (donnant le meilleur taux de reconnaissance lors de la comparaison avec l'arbre de décision figure 5) et de la signature structurelle développée dans [6]. Cette signature structurelle intègre une information complémentaire de celle proposée par les approches statistiques telle que la signature de Radon. Elle décrit l'organisation spatiale entre les primitives structurelles composant chaque symbole sous la forme d'un graphe topologique. Cette information spatiale est extraite du graphe topologique en calculant des chemins caractérisant des sous-structures spatiales (carré, losange, triangle,…) incluses dans le symbole. Ces chemins sont disposés sous la forme d'un vecteur de valeurs, qui est utilisé comme signature par la méthode Navigala. D'après les taux de reconnaissance obtenus (voir figure 7), la signature structurelle semble suffisamment fiable pour être utilisée conjointement avec des approches statistiques pour améliorer les résultats actuels. Cette combinaison pourrait être mise en place par une reconnaissance hiérarchique itérative où la classification à une étape donnée serait raffinée au cours de l'étape suivante.

Le treillis de Galois offre des perspectives intéressantes pour la reconnaissance d'objets détériorés. Dans ce cadre, sa structure apporte une plus grande robustesse à la classification que celle de l'arbre de décision.

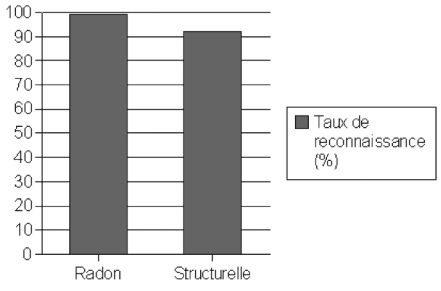

Figure 7. Comparaison des taux de reconnaissance obtenus par le treillis de Galois selon la signature utilisée.

5. Conclusion et perspectives

Ce papier s'intéresse aux treillis de Galois utilisés dans la méthode de reconnaissance de symboles Navigala, basée sur un parcours dans un treillis de Galois, et plus généralement aux treillis dits *treillis dichotomiques*.

Parmi les différentes utilisations du treillis en classification supervisée, celle mise en place dans la méthode Navigala est basée sur une navigation dans le treillis, navigation similaire à l'utilisation classique d'un arbre de décision. Une telle navigation dans un treillis de Galois induit les mêmes avantages qu'un arbre de décision, à savoir la lisibilité du modèle et la capacité à sélectionner automatiquement les variables discriminantes parmi un très grand nombre de variables.

Notons cependant que la taille de l'arbre de décision est plus condensée que celle du treillis de Galois, mais que pour atténuer cet inconvénient le treillis peut être généré à la demande en cours de classification. À la différence de l'arbre, le treillis propose plusieurs chemins de classification ou de reconnaissance, ce qui lui apporte une certaine robustesse lorsqu'il s'agit de reconnaître des données détériorées.

Dans ce papier, nous décrivons les liens structurels qui unissent arbre de décision et treillis dichotomique en montrant que tout arbre de décision est inclus dans le treillis, mais aussi que le treillis est en fait la fusion de tous les arbres de décision. Du lien de fusion nous déduisons un algorithme d'extraction d'un arbre de décision à partir d'un treillis dichotomique. Nous donnons également quelques résultats expérimentaux mettant en valeur la similarité existant entre ces deux structures, ainsi que la plus grande robustesse du treillis par rapport à l'arbre. Une perspective envisagée de ces travaux serait de mener une étude comparative de tous les arbres que l'on peut extraire d'un treillis dichotomiques dans le but de comparer différents critères de sélection.

Références

[1] Base d'images GREC (Graphics RECognition), www.cvc.uab.es/grec2003/SymRecContest/ index.htm.

[2] M. BARBUT and B. MONJARDET. *Ordre et classification, Algèbre et combinatoire*. Paris, 1970. 2 tomes.

[3] R. BELOHLAVEK, B. DE BAETS, J. OUTRATA, and V. VYCHODIL. Inducing decision trees via concept lattices. In *Fifth International Conference on Concept Lattices and their Applications* (CLA'2007), pages 38-49, Montpellier, France, October 24-26, 2007.

[4] L. BREIMAN, J. H. FRIEDMAN, R. A. OLSHEN, and C. J. STONE. *Classification and regression trees*. Wadsworth Inc., Belmont, California, 1984.

[5] N. CASPARD, B. LECLERC, and B. MONJARDET. *Ensembles ordonnés finis : concepts, résultats, usages*. Mathématiques et Applications. SPRINGER, 09 2007.

[6] M. COUSTATY, S. GUILLAS, M. VISANI, K. BERTET, and J-M. OGIER. Flexible structural signature for symbol recognition using a concept lattice classifier. In *Seventh IAPR International Workshop on Graphics Recognition (GREC'07)*, Curitiba, Brazil, September 20-21 2007.

[7] S. DERRODE, M. DAOUDI, and F. GHORBEL. Invariant content-based image retrieval using a complete set of Fourier-Mellin descriptors. *Int. Conf. on Multimedia Computing and Systems (ICMCS'99)*, pages 877-881, june 1999.

[8] B. GANTER and R. WILLE. *Formal concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 1999.

[9] S. GUILLAS. *Reconnaissance d'objets graphiques détériorés : approche fondée sur un treillis de Galois*. PhD thesis, Université de La Rochelle, 2007.

[10] H. HOTELLING. Relations between two sets of variates. *Biometrika*, 28(2) : 321-377, 1936.

[11] S. KUZNETSOV. Machine learning on the basis of formal concept analysis. *Automation and Remote Control archive*, 62(10) : 1543-1564, 2001.

[12] S. KUZNETSOV. Machine learning and formal concept analysis. *Lecture notes in computer science : Innovations in applied artificial intelligence ; From International conference on industrial and engineering applications of artificial intelligence and expert systems No17*, 3029 :287–312, 2004.

[13] E. MEPHU NGUIFO. Une nouvelle approche basée sur le treillis de Galois, pour l'apprentissage de concepts. *Mathématiques et Sciences Humaines*, 124 : 19-38, 1993.

[14] E. MEPHU-NGUIFO and P. NJIWOUA. Treillis des concepts et classification supervisée. *Technique et Science Informatiques*, *RSTI*, 24(4) : 449-488, 2005. Hermés – Lavoisier, Paris, France.

[15] G. OOSTHUIZEN. *The use of a Lattice in Knowledge Processing*. PhD thesis, University of Strathclyde, Glasgow, 1988.

[16] J. R. QUINLAN. *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[17] J. R. QUINLAN. Induction of decision trees. *Machine Learning*, 1, 1986.

[18] R. RAKOTOMALALA. *Graphes d'induction*. PhD thesis, Université Claude Bernard, Lyon I, Décembre 1997.

[19] R. RAKOTOMALALA. *Arbres de décision*. *Revue MODULAD*, 33, 2005.

[20] M. SAMUELIDES and E. ZENOU. Learning-based visual localization using formal concept lattices. In *2004 IEEE Workshop on Machine Learning for Signal Processing*, page 10, 2004.

[21] S. TABBONE and L. WENDLING. Recherche d'images par le contenu à l'aide de la transformée de Radon. *Technique et Science Informatiques*, 2003.

[22] M. TEAGUE. Image analysis via the general theory of moments. *Journal of Optical Society of America (JOSA)*, 70 : 920-930, 2003.

[23] F.J. VENTER, G.D. OOSTHUIZEN, and J.D. ROOS. Knowledge discovery in databases using lattices. *Expert Systems With Applications*, 13(4) : 259-264, 1997.

[24] E. ZENOU and M. SAMUELIDES. Utilisation des treillis de Galois pour la caractérisation d'ensembles d'images. In *14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2004)*, volume 1, pages 395-404, 2004.