Some Links Between Decision Tree and Dichotomic Lattice

S. Guillas, K. Bertet, M. Visani, J-M. Ogier, and N. Girard

Laboratory L3I - University of La Rochelle - France
{sguillas,kbertet,mvisani,jmogier,ngirard02}@univ-lr.fr

Abstract. 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. When defined from binary attributes obtained after a discretization pre-processing step, and more generally when a non-empty set of *complementarity attributes* can be associated to each binary attribute, lattices are denoted as ”dichotomic lattices”. 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.

Key words: Classification, Galois lattice, Concept lattice, Decision tree, Navigation

1 Introduction

Galois lattice (or *concept lattice*) has first been introduced in a formal way in graph and ordered structures theory [1,2,3]. Afterwards it has been developed in the field of Formal Concept Analysis (FCA) [4] for data analysis and classification. The concept lattice structure, based on the notion of *concept*, enables to describe data and to preserve their diversity and complexity. A study realized by Mephu Nguifo and Njiwoua [5] confirms the effectiveness of concept lattices for classification and describes selection-oriented methods. Even though these structures are associated to a high time and space complexity (exponential in the worst case), the technological improvements that have been performed during the last decades enable their use.

Galois lattice gives a representation of all the possible correspondences (denoted as concepts) between a set of *objects* (or examples) O and a set of *attributes* (or features) I. Whereas in decision trees the path from the root to a given leaf is unique, in Galois lattices there are multiple paths from the maximal concept to a given terminal concept. Since a lattice is defined from binary attributes, the continuous-valued primitives have to be discretized (after being normalized) in a pre-processing step.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 193–205,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

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. The selection-oriented methods come from the field of data mining and rely on a selection step where the Galois lattice is used to choose concepts which encode relevant information from the huge amount of available data. The classification step is then performed by some usual classifier (k-nearest neighbours, Bayesian classifier. . .).

On the contrary, in the navigation-oriented methods there is no selection step and classification is performed by navigating through the complete lattice. Similar to the classification tree, we navigate from a node to its successors until a labeled (terminal) concept is reached. Indeed, Galois lattice is a graph whose structure is similar to that of a decision tree. Whereas in decision trees the path from the root to a given leaf is unique, in Galois lattices there are multiple paths from the maximal boundary to a given terminal concept.

This similarity between lattices used by navigation-oriented methods and decision trees has been mentionned and stated in some works [6,7,8]. Similar to the classification tree, we navigate from a node to its successors until a labeled (terminal) concept is reached. It is mentionned in particular for the Navigala method we have developped, dedicated to symbols classification [9,10] for an objective of noisy symbols recognition. In order to reduce the size of the lattice, which is generally more important than the size of the tree, the Navigala method proposes a lattice generation performed on-demand during the classification step.

As a first consequence of the similarity between lattice and decision tree, the navigation-oriented methods shares the advantages of the decision tree in terms of readability and ability to automatically select discriminatory variables among a large number of variables. And, contrary to decision trees where there is a unique navigation path to a given node, lattices propose several paths. This property provides to lattices enhanced robustness towards noise.

In this paper, we precise and extend the links between these two structures of lattice and decision tree in the particular case of dichtomic lattices, i.e lattices defined from binary features where a non-empty set of *complementarity attributes* can be associated to each feature:

- Every decision tree is included in the dichotomic lattice, when both structures are built from the same binary attributes.
- Every dichotomic lattice is the merger of all the decision trees when these structures are built from the same binary attributes.

Galois lattice and the navigation-oriented method Navigala are described in Section 2. Section 3 provides a proper definition for dichotomic lattices and the two main results of this paper concerning structural links between dichotomic lattice and decision tree.

2 Navigala: recognition of symbols by navigation in a Galois lattice

2.1 Galois lattice definition

The *concept lattice* is built from a relation R between objects O and attributes I. This graph is composed of a set of concepts ordered by inclusion. It verifies the properties of a lattice: the relation between concepts is an order relation (transitive, reflexive and antisymmetric), and there are a lower bound and an upper bound for each pair of concepts in the graph. We associate to a set of objects $A \subseteq O$ the set $f(A)$ of attributes in relation R with the objects of A:

$$f(A) = \{x \in I \mid pRx \; \forall \, p \in A\}$$

Dually, for every set of attributes $B \subseteq I$, we define the set $g(B)$ of objects in relation with the attributes of B:

$$g(B) = \{p \in O \mid pRx \; \forall \, x \in B\}$$

The relations between the set of objects and the set of attributes are described by a *formal context*. A formal context C is a triplet $C = (O, I, R)$ represented by a table (see for instance Table 1).

Table 1. Example of formal context

Class	Id	Sunniness			Humidity		Wind	
		Sun	Cloudy	Rain	< 77.5	>= 77.5	Yes	No
Y	1	X			X		X	
N	2	X				X	X	
N	3	X				X		X
N	4	X				X		X
Y	5	X			X			X
Y	6		X			X	X	
Y	7		X			X		X
Y	8		X		X		X	
Y	9		X		X			X
N	10			X		X	X	
N	11			X	X		X	
Y	12			X		X		X
Y	13			X		X		X
Y	14			X		X		X

The two functions f and g defined between objects and attributes form a *Galois connection*. The composition $\varphi = f \circ g$ defined on the attributes set enables to associate to each subset of attributes $X \subseteq I$ the smallest concept containing X: $(g(\varphi(X)), \varphi(X))$. This composition φ verifies the properties of a closure operator: φ is idempotent (*i.e.* $\forall X \subseteq S$, $\varphi^2(X) = \varphi(X)$), extensive (*i.e.* $\forall X \subseteq S$, $X \subseteq \varphi(X)$) and isotone (*i.e.* $\forall X, X' \subseteq S, X \subseteq X' \Rightarrow \varphi(X) \subseteq \varphi(X')$).

The *Galois lattice* associated to a formal context C is a graph composed of a set of *formal concepts* equipped with a particular binary relation. Intuitively this graph is a representation of all the possible maximal correspondences between a subset of *objects* (or instances, examples) O and a subset of *attributes* (or primitives, features) I. A *formal concept* is a maximal objects-attributes subset where objects and attributes are in relation. More formally, it is a pair (A, B) with $A \subseteq O$ and $B \subseteq I$, which verifies $f(A) = B$ and $g(B) = A$. Let us introduce the binary relation $\leq$ defined on the set of all the concepts by, for two formal concepts (A_1, B_1) and (A_2, B_2):

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow \left\| \begin{array}{l} A_2 \subseteq A_1 \\ \text{(equivalent to } B_1 \subseteq B_2) \end{array} \right.$$

All the set of formal concepts equipped with the order relation $\leq$ forms a lattice called a *concept lattice* or *Galois lattice*. Thus, for each concepts (A_1, B_1) and (A_2, B_2), it exists a greatest lower bound (resp. a least upper bound) called *meet* (resp. *join*) denoted as $(A_1, B_1) \wedge (A_2, B_2)$ (resp. $(A_1, B_1) \vee (A_2, B_2)$) defined by:

$$(A_1, B_1) \wedge (A_2, B_2) = (g(B_1 \cap B_2), (B_1 \cap B_2)) \tag{1}$$

$$(A_1, B_1) \vee (A_2, B_2) = ((A_1 \cap A_2), f(A_1 \cap A_2)) \tag{2}$$

Therefore, a lattice contains a minimum (resp. maximum) element according to the relation $\leq$ called the *bottom* (resp. *top*) of the lattice, and denoted as $\perp = (O, f(O))$ (resp. $\top = (g(I), I)$). For more information on Galois lattice and closure systems, the reader can refer to [1,4].

Figure 1 shows an example of concept lattice built from the formal context in Table 1. This formal context is composed of a set of 14 objects described by 7 attributes (sun, $cloudy$, $rain$, $hum < 77.5$, $hum \geq 77.5$, $windY$ and $windN$).

2.2 Navigala method description

The navigation-base recognition method named Navigala (NAVIgation into GAlois LAttice) has been introduced in [11]. This method is fitted for recognizing *noisy graphic objects* and especially *symbol images*. Such symbols appear in technical documents such as architectural plans or electrical schemes. Graphic objects may be described by statistical or structural primitives. As statistical features describe the spatial distributions of the pixel values of the symbol, structural primitives describe the spatial or topological relations between some sub-patterns extracted from the symbol images. In the following, the primitives vector of each symbol is called the signature of this symbol.

Navigala is a supervised classification approach, no matter if the discretization preprocessing relies on a supervised or unsupervised criterion. This method relies on the classical steps of recognition: data preparation that mainly consists in discretizing continuous data, learning where the Galois lattice is built, and classification where the samples to recognize are labeled after navigating through the graph until they reach a labeled concept.

Data preparation Firstly, several signatures are extracted from the symbol images: statistical signatures (Fourier-Mellin invariants [12], Radon transform-based Radon transform [13], Zernike moments [14]), and a structural signature named *flexible structural signature* [15]. Data preparation then consists in normalizing the various features. The continuous valued primitives must then be *discretized*. At each step of discretization, a

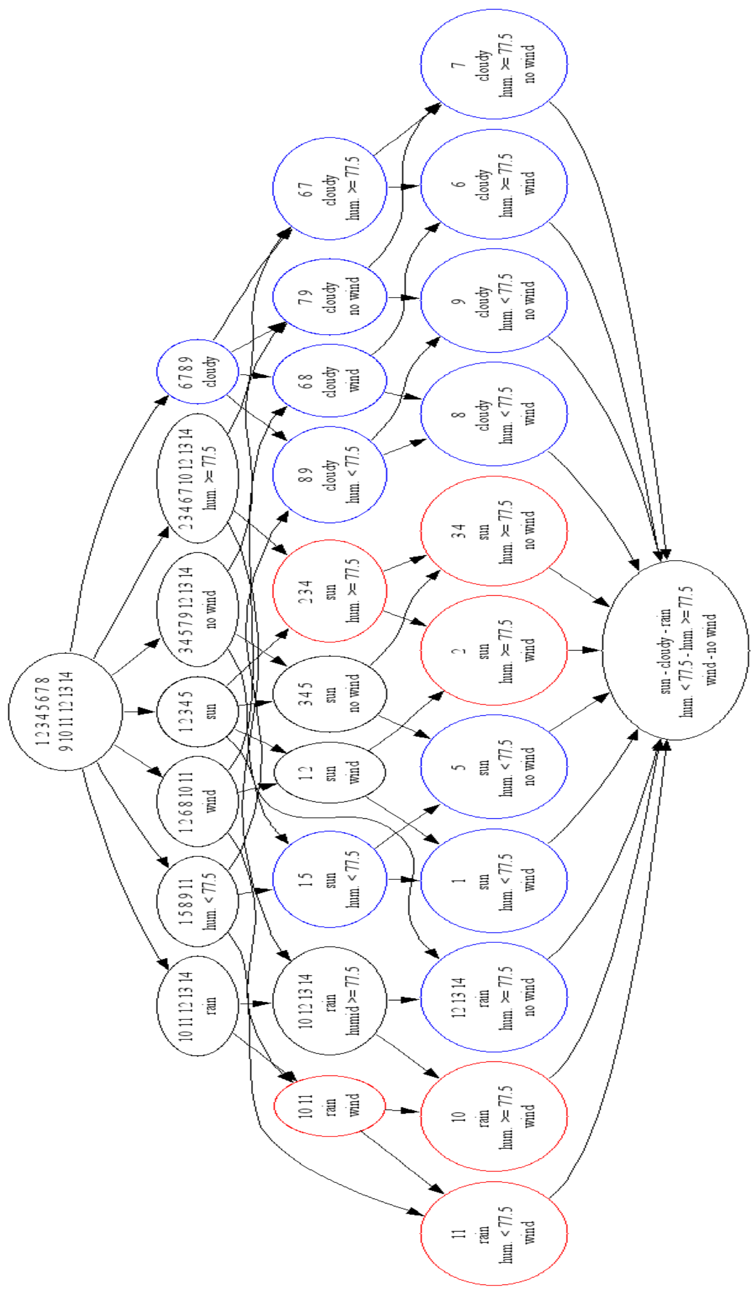

Fig. 1. Example of Galois lattice

criterion selects both the primitive to divide and the optimal cutting point. Let $x \in I$ be a primitive interval composed of values $V_x = (v_1 \dots v_n)$ sorted by ascending order. The interval will be cut between the values v_j and v_{j+1}, where v maximizes a given "cutting" criterion of the primitive values objects. We can define a lot of cutting criteria, supervised or not. Among these criteria, let us mention maximal distance, entropy and Hotelling's coefficient. Discretization is processed until a given *stopping criterion* is met. The stopping criterion used for Navigala is based on class separation, *i.e.* this criterion is met when each class of objects can be represented by its own set of intervals. More precisely, one class can be separated from the others when the objects characterizing this class share at least one interval which enables to distinguish them from the objects of the other classes. At the end of the process, the continuous-valued primitives are converted into intervals of values, called discretized data. Once intervals are computed, they are extended to a fuzzy number.

Learning The discretized data obtained from the data preparation will then be used as a training set, in order to compute the Galois lattice. The generation algorithm [11] is an extension of the Bordat's algorithm [16] since navigation uses the *Hasse diagram* of the lattice (an example of Hasse diagram of a Galois lattice is shown on Figure 1). Once the Hasse diagram is computed, as each concept contains a set of objects, it is possible to label them depending on these objects. Indeed, when all the objects in a given concept correspond to the same class, this concept is named *final concept* and can be labeled. Intuitively, these final concepts correspond to the classes to reach when the Galois lattice is explored for the classification of a new object.

Classification Using the Hasse diagram of the Galois lattice, we can process recognition of the new symbols belonging to the test set. Classification of a new symbol is then processed by navigating through the graph, from the minimal concept (the top $\top$) to a final concept which has been previously labeled by a class. Intuitively, during this progression, we observe a specification of the objects set and a generalization of the attributes set, that is to say that the number of objects is reduced while the number of attributes is increased. Thus, we *refine the description* of the object to recognize, until it corresponds to the description of one of the learning objects whose class will be assigned to the object to recognize. The progression in the graph from a concept to its successor is done according to a *fuzzy distance measure* and a *choice criterion* (for more details please refer to [11]).We estimate the distance between the signature values of the object to recognize and the signatures values of the learning objects, and we choose the successor concept in the Galois lattice whose description best corresponds to the object.

The Galois lattice construction algorithm we use holds several advantages: it is quite easy to implement, and it enables an *on-demand concepts generation* of the Galois lattice. In other words, it enables to generate from a given concept every successor concept in the lattice. This is interesting because it avoids the construction of the whole graph, which can be of exponential complexity in the worst case. Indeed, recognition is performed by exploring only a small region of the lattice. On-demand concepts generation therefore considerably reduces the complexity of the structure generation.

3 Lattice and decision tree

In Navigala, the classification based on a navigation into the graph is quite similar to the one proposed with a decision tree. In this section, we describe both of these structures, and the handled data.

3.1 Decision tree definition

Since years 1960-1970, the decision tree built from a data set has been used in several research works [17,18]. Among the most widely used decision tree generation methods, we can cite CART [19], C4.5 [20] and ID3 [21]. As with a Galois lattice, the data are represented by a table containing a set of objects, set described by a set of attributes. This table can contain discrete, ordinal or continuous data. Decision tree nodes are built from its top, called the root, to its basis where the terminal nodes are called leaves. The construction of a decision tree requires three criteria: a selection criterion, which enables, at each division step, to select one/several feature(s) in the table, a second criterion to discretize the continuous data, and a last criterion to stop the divisions in the tree, which is generally based on a purity measure of the leaves. The root regards all the set of objects in the table; a feature of the table is then selected to separate the objects into two distinct subsets corresponding to two children nodes. This process is likewise iterated on each subset until the stopping criterion is satisfied (see for instance Figure 2).

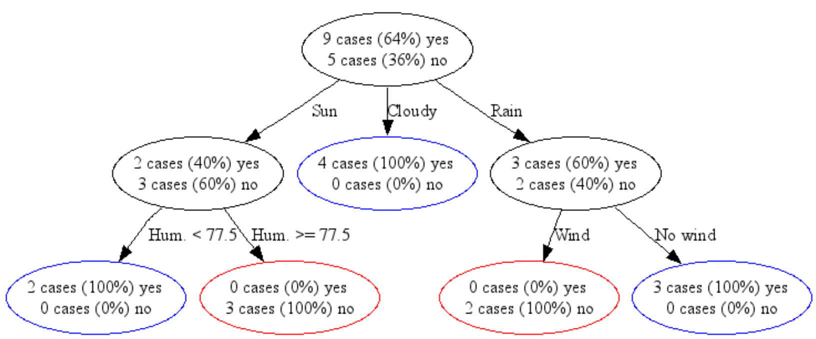

Fig. 2. Example of decision tree

When the features are continuous-valued, as it is the case with the signatures we consider, they need a discretization step which can be processed:

- during the tree construction. Only the selected features will then be discretized.
- before the tree construction, in a pre-processing stage. The data will then be discretized until the classes are separated.

Several heuristics can be used for decision tree construction. For example, a pruning stage can be performed on the decision tree to avoid over-partitioning the data. The pruning principle is to raise into the tree from the leaves by changing nodes in leaves depending on a purity criterion of the nodes. In the structural comparison we describe in the following, the considered decision trees are not pruned.

3.2 Dichotomic lattice

As the decision tree construction infers a discretization of the data, it is possible to consider the binary data table issued from this discretization, and consequently the resulting Galois lattice. We thus find in this binary table (see Table 1), and in the Galois lattice (see Figure 1), the same binary attributes as those proposed by the decision tree (see Figure 2). Thus, when a feature V is proposed, with two children, one for *yes*, and the other for *no*, we had to consider the two binary attributes $V = yes$ and $V = no$. In a more general way, the binary attributes issued from the children of a node are present into the table and separate the set of objects.

In Navigala method, features are continuous data which are discretized in a pre-processing stage in order to obtain classes' separation. The binary attributes in the table are intervals issued from this discretization. A symbol is described by a fixed-size signature before the discretization, and then by a set of binary intervals with the same cardinality after discretization. A symbol is associated to only one interval among the set of intervals issued from a same feature.

Notice that the obtained binary attributes infer an automatic selection of the discriminant features. Indeed, an attribute belonging to all the set of objects will not been proposed in the decision tree, and consequently will not been taken into account in the table. It is the same in Navigala method where a non discretized continuous feature will not appear in the table.

When all objects in a binary table are associated to a same number of binary attributes, the final concepts (*i.e.* the concepts corresponding to a unique class) contain the same number of attributes. The final concepts of a lattice cannot be related the ones to the others (because two concepts in relation $\leq$ can not be composed of a same number of attributes). The final concepts thus have as a unique direct successor the concept $\top$. This property can be found in lattice theory with the notion of *co-atomisticity*. It is the case in our approach Navigala. When discretization is performed (during decision tree construction), the table depends on the proposed attributes in the tree, and two different trees could infer two different binary attributes sets. These two attributes sets can then infer two different lattices. The discretization can also be performed in pre-processing, as in the method Navigala. From this table, several decision trees can be generated but a unique lattice will be associated.

Whatever the case, to each binary attribute x we can associate a non empty set $\overline{X}$ of binary attributes such as the objects having the attribute x, and those having the attributes in $\overline{X}$ are all distinct. The binary attributes are deduced from the decision tree: when x is a feature proposed by a node of the tree, then $\overline{X}$ is a set of all the other features proposed by this same node. Using continuous features discretized in a pre-processing stage, x corresponds to an interval, and $\overline{X}$ contains all remaining intervals corresponding to this same feature. From this property, lattices issued from a tree belong to particular lattices called *dichotomic lattices*. More formally, dichotomic lattices are characterized by the fact to be $\vee$-complementary, that is to say that for each concept (A, B), a *complementary concept* (A', B') always exists such as

$$(A, B) \vee (A', B') = \top = (\emptyset, I) \tag{3}$$

Proposition 1 *Each dichotomic lattice (i.e. lattice issued from a tree) is $\vee$-complementary.*

Proof. Let (A, B) be any concept of a dichotomic lattice. It consists in showing the existence of a complementary concept to (A, B). We consider x any binary attribute of B, and $\overline{x}$ a complementary attribute of x belonging to the set $\overline{X}$. Thus, the objects having x, and those having $\overline{x}$ are distinct. This is formalized by $g(\{x\}) \cap g(\{\overline{x}\}) = \emptyset$. Then we consider the smallest concept containing $\overline{x}$ which, by definition, will be the concept $(g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\}))$ where the set of attributes is $\varphi(\{\overline{x}\})$. From the definition of functions f and g, we deduce that $g(\varphi(\{\overline{x}\})) = g(\{\overline{x}\})$, and that $A \subseteq g(\{x\})$. Assuming that $g(\{x\}) \cap g(\{\overline{x}\}) = \emptyset$, we can then deduce that $A \cap g(\varphi(\{\overline{x}\})) = \emptyset$. Consequently, $(A, B) \vee (g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\})) = (\emptyset, I)$, and the concept $(g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\}))$ is the complementary concept of (A, B). It proves the $\vee$-complementarity of the lattice.

3.3 Structural links between dichotomic lattice and decision tree

A first structural link between decision tree and dichotomic lattice consists in the fact that both structures can be used in classification, and can be defined from a table of binary attributes.

We can notice that the use of navigation-based lattices for classification is similar to the one of decision trees. This similarity is formalized by a structural link between nodes and concepts: indeed, every node in the decision tree may be associated to a unique concept in the lattice. We consider a node n in the tree, and the set of binary attributes X_n proposed from the root to this node. Assuming that these binary attributes belong to the table corresponding to the lattice construction, we then associate to the node n the smallest concept containing the features of X_n:

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

Figure 2 presents the decision tree associated to the data of the example. Notice that all the nodes of the decision tree are present into the lattice whatever the construction criterion of the decision tree. Moreover, the structure of the decision tree is also preserved in the lattice as shown in figure 3, where the tree (in bold) is included in the lattice. This property is verified in the general case. Thus, we show that each decision tree is included in the Galois lattice. We also prove that the lattice is the merger of all the decision trees.

Proposition 2 *Each decision tree is included in the dichotomic lattice, when both structures are built from the same binary attributes.*

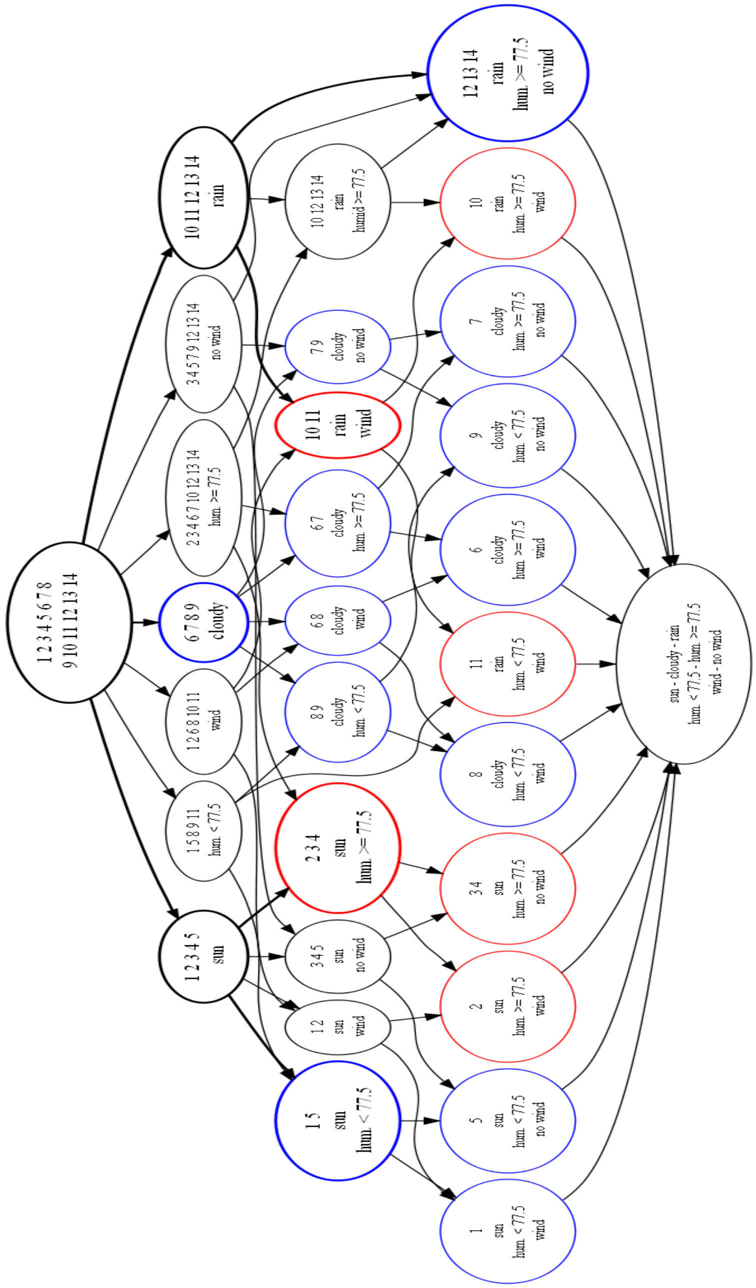

Fig. 3. Inclusion of the decision tree (in bold) in the Galois lattice

Proof. Let us consider a decision tree and a dichotomic lattice issued from the same binary attributes. As mentioned before, these two structures handle the same binary attributes. Moreover, to a node n of the decision tree accessible by validation of the set of attributes X_n we associate the concept $(g(\varphi(X_n)), \varphi(X_n))$. To prove that the decision tree is included into the lattice, it is necessary to prove the three following points:

1. Two different nodes of a decision tree are associated to different concepts:
   By contradiction, when two nodes n_1 and n_2 are associated to a same concept, then $\varphi(X_{n_1}) = \varphi(X_{n_2})$. It means that the same objects share the attributes of X_{n_1} and X_{n_2}, that is in contradiction with the fact that two nodes n_1 and n_2 are two different nodes of the decision tree.
2. When two nodes are ancestors in the decision tree, then their associated concepts are related in the lattice:
   Clearly when a node n_1 is ancestor of a node n_2 in a decision tree, then $X_{n_1} \subseteq X_{n_2}$. The operator φ being isotone, we deduce that $\varphi(X_{n_1}) \subseteq \varphi(X_{n_2})$, and consequently that these two concepts $(g(\varphi(X_{n_1})), \varphi(X_{n_1}))$, $(g(\varphi(X_{n_2})), \varphi(X_{n_2}))$ are related depending on the relation $\leq$.
3. Conversely, when two nodes are not ancestor in the decision tree, then their associated concepts are not related in the lattice:
   When a node n_1 is not ancestor of a node n_2, then we need to consider all the children of the smallest common ancestor to n_1 and n_2, and particularly the child n'_1 ancestor of n_1 and the child n'_2 ancestor of n_2. These two nodes n'_1 and n'_2 exist by construction of the table. Clearly, as n'_1 and n'_2 are brothers, their attributes in the associated concepts, being $\varphi(X_{n'_1})$ and $\varphi(X_{n'_1})$, are not shared by any object. That is formalized by $g(\varphi(X_{n'_1})) \cap g(\varphi(X_{n'_2})) = \emptyset$. Then, n'_1 being ancestor of n_1, we can deduce that $X_{n'_1} \subseteq X_{n_1}$, where $\varphi(X_{n'_1}) \subseteq \varphi(X_{n_1})$ by isotony of the operator φ, and reversely $g(\varphi(X_{n'_1})) \supseteq g(\varphi(X_{n_1}))$ by definition of g. We also have $g(\varphi(X_{n'_2})) \supseteq g(\varphi(X_{n_2}))$ because n'_2 is ancestor of n_2. Thus we deduce that $g(\varphi(X_{n_2})) \cap g(\varphi(X_{n_1})) = \emptyset$, and that proves that the concepts associated to the nodes n_1 and n_2 are not in relation by $\leq$.

Proposition 3 *A dichotomic lattice is the merger of all the decision trees when these structures are built from the same binary attributes.*

Proof. We previously proved that each decision tree is included into the dichotomic lattice built from the same binary attributes. To prove that the dichotomic lattice is the merger of all the decision trees, we must prove that each concept potentially belong to a decision tree. This proof is given by construction.

We consider an any concept (A, B). Then we build the subset of concepts C of the lattice containing: the concept (A, B), a complementary concept (A', B') to (A, B), the minimal concept $\perp$, and all the final successors concepts of (A, B) and (A', B'). The existence of the complementary concept (A', B') is deduced from the $\vee$-complementarity property of the dichotomic lattice. Moreover, it infers that this subset C in addition to the relation $\leq$ forms a tree. Then we add in the set C a maximal number of concepts of the dichotomic lattice such as $(C, \leq)$ preserves the property to be a tree. Thus, by construction, we obtain a sub-tree included into the dichotomic lattice, containing (A, B).

In this sub-tree, the leaves are final concepts and correspond to subsets of objects which can not be separated by any binary attribute, *i.e.* the classes when the data have been discretized until the classes are separated. This tree can thus be considered as a decision tree, what finishes this proof.

4 Conclusion

This paper is about Galois lattice which is used as a classifier in the *Navigala* approach and, more generally, about *dichotomic lattices* defined from a structural way: to every binary feature, a non-empty set of *complementarity features* can be associated.

There is some published work about using Galois lattices as a classifier: as most of the proposed approaches consider the lattice as a concept selection tool, Navigala performs classification by navigating through the lattice from one node to its successors, similar to a classification tree.

As a first consequence, the Navigala approach shares the advantages of the decision tree in terms of readability and ability to automatically select discriminatory variables among a large number of variables. As another consequence, contrary to decision trees where there is a unique navigation path to a given node, lattices propose several paths. This property provides to lattices enhanced robustness towards noise.

The inclusion result (Proposition 2) of this paper implies that navigation paths proposed by decision tree are included in the dichotomic lattice issued from the same binary features. Moreover, Proposition 3 states that the every dichotomic lattice is equal to the merge of navigation paths of all the decision trees.

This work opens the way for the definition of a new method that would combine the advantages of both trees and lattices.

References

1. Barbut, M., Monjardet, B.: Ordre et classification, Algèbre et combinatoire, Paris (1970) 2 tomes.
2. Birkhoff, G.: Lattice theory. 3rd edn. Volume 25. American Mathematical Society (1967)
3. Davey, B., Priestley, H.: Introduction to lattices and orders. 2nd edn. Cambridge University Press (1991)
4. Wille, R.: Restructuring lattice theory: an approach based on hierarchy on contexts. Ordered sets (1982) 445–470
5. Mephu-Nguifo, E., Njiwoua, P.: Treillis des concepts et classification supervisée. Technique et Science Informatiques, RSTI **24** (2005) 449–488 Hermès - Lavoisier, Paris, France.
6. Kuznetsov, S.: Machine learning on the basis of formal concept analysis. **62** (2001) 1543 – 1564
7. Kuznetsov, S.: Machine learning and formal concept analysis. **3029** (2004) 287–312
8. Guillas, S., Bertet, K., Ogier, J.M.: Concept lattice classifier : a first step towards an iterative process of recognition of noised graphic objects. In: Fourth International Conference on Concept Lattices and their Applications (CLA'2006). (2006) 257–263
9. Guillas, S.: Reconnaissance d'objets graphiques détériorés : approche fondée sur un treillis de Galois. PhD thesis, Université de La Rochelle (2007)

10. Guillas, S., Bertet, K., Ogier, J.M.: A generic description of the concept lattices' classifier: Application to symbol recognition. In Wenyin, L., Lladós, J., eds.: Graphics Recognition: Ten Years Review and Future Perspectives - Selected papers from GREC'05. Volume 3926., LNCS (2006) 47–60
11. Bertet, K., Guillas, S., Ogier, J.M.: Extensions of bordat's algorithm for attributes. In: Fifth International Conference on Concept Lattices and their Applications (CLA'2007), Montpellier, France (2007) 38–49
12. Derrode, S., Daoudi, M., Ghorbel, F.: Invariant content-based image retrieval using a complete set of Fourier-Mellin descriptors. Int. Conf. on Multimedia Computing and Systems (ICMCS'99) (1999) 877–881
13. Tabbone, S., Wendling, L.: Recherche d'images par le contenu à l'aide de la transformée de Radon. Technique et Science Informatiques (2003)
14. Teague, M.: Image analysis via the general theory of moments. Journal of Optical Society of America (JOSA) **70** (2003) 920–930
15. Coustaty, M., Guillas, S., Visani, M., Bertet, K., Ogier, J.M.: Flexible structural signature for symbol recognition using a concept lattice classifier. In: Seventh IAPR International Workshop on Graphics Recognition (GREC'07), Curitiba, Brazil (2007)
16. Bordat, J.: Calcul pratique du treillis de Galois d'une correspondance. Math. Sci. Hum. **96** (1986) 31–47
17. Rakotomalala, R.: Graphes d'induction. PhD thesis, Université Claude Bernard, Lyon I (1997)
18. Rakotomalala, R.: Arbres de décision. Revue MODULAD **33** (2005)
19. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth Inc., Belmont, California (1984)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman, Los Altos, California (1993)
21. Quinlan, J.: Induction of decision trees. Machine Learning **1** (1986)