

NAVIGALA: AN ORIGINAL SYMBOL CLASSIFIER BASED ON NAVIGATION THROUGH A GALOIS LATTICE

M. VISANI*, K. BERTET and J.-M. OGIER

*Department of Computer Science — Laboratory L3I — University of La Rochelle
Ple Sciences et Technologie, Avenue Michel Crpeau
17042 La Rochelle Cedex 1, France
muriel.visani@univ-lr.fr

This paper deals with a supervised classification method, using Galois Lattices based on a navigation-based strategy. Coming from the field of data mining techniques, most literature on the subject using Galois lattices relies on selection-based strategies, which consists of selecting/choosing the concepts which encode the most relevant information from the huge amount of available data. Generally, the classification step is then processed by a classical classifier such as the k -nearest neighbors rule or the Bayesian classifier. Opposed to these selection-based strategies are navigation-based approaches which perform the classification stage by navigating through the complete lattice (similar to the navigation in a classification tree), without applying any selection operation. Our approach, named Navigala, proposes an original navigation-based approach for supervised classification, applied in the context of noisy symbol recognition. Based on a state of the art dealing with Galois Lattices classification based methods, including a comparison between possible selection and navigation strategies, this paper proposes a description of NAVIGALA and its implementation in the context of symbol recognition. Some objective quantitative and qualitative evaluations of the approach are proposed, in order to highlight the relevance of the method.

Keywords: Supervised classification; Galois (or concept) lattice; decision tree; graphical documents; graphical symbols recognition.

1. Introduction

Galois lattices (or *concept lattices*) were first introduced in a formal way in the graph and ordered structures theory.^{3,9} Later, it was developed in the field of Formal Concept Analysis (FCA)¹³ for data analysis and classification. The concept lattice structure, based on the notion of *concept*, enables data description while preserving its diversity.

Galois lattice is a graph with a structure similar to that of a tree. It provides a representation of all the possible correspondences 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 boundary to a given terminal concept. The technological improvements of the last decades enable the use of these structures for data mining problems though they are exponential in space/time (worst case). It has to be noted that in practice, in most cases, the size of the lattice remains reasonable. Recent studies realized by Mephu Nguifo *et al.*^{22,20} provide a comprehensive review of some of the state-of-the-art classification approaches based on concept lattices, which are generally based on a selection of the most pertinent concepts in the lattice. This review shows that these methods are able to catch up with (and sometimes even outperform) more classical approaches such as decision trees. Multiple approaches have been proposed so far, confirming the relevance of using a Galois lattice for a classification task. Among these approaches, we can mention LEGAL and LEGAL-E,¹⁹ Galois,⁷ Zenou and Samuelides',²⁸ GRAND²⁵ and RULEARNER²⁷ which are based on a selection of the concepts directly, the CIBLe approach²¹ which is based on object filtering and the CLNN and CLNB methods³⁴ where contextual rules are used.

The first objective of this paper is to introduce an original supervised classification method that does not rely on a selection step, named Navigala. Indeed, differing from the state-of-the-art approaches, Navigala relies on navigation through the lattice. The second objective of this paper is to compare Navigala (both formally and experimentally) to several other classification methods based on the Galois lattice. Navigala has been developed in the field of content-based graphical documents indexing; it is dedicated to noisy symbol recognition. These symbols, which are issued from digitized paper documents such as architectural or electrical plans, are most often noisy. In the proposed scheme, each symbol image is represented by a feature vector (signature), which may be statistical, structural or hybrid. The signatures are discretized to obtain discrete attributes and then classified using the Galois lattice.

This paper is organized as follows. In Sec. 2, we describe the Galois lattice structure and its properties and provide a review of the state-of-the-art classification approaches based on a Galois lattice. In Sec. 3, we present our navigation-based approach named Navigala. Then, Sec. 4 proposes various experimental results assessing the effectiveness of the proposed approach and an experimental comparative study towards *selection*-oriented approaches and decision trees. The conclusion and future works are presented in Sec. 5.

2. Description of a Galois Lattice

2.1. Definition

The *concept lattice* is a particular graph defined and generated from a relation R between objects O and attributes I . This graph is composed of a set of *concepts* ordered by a relation verifying the properties of a lattice, i.e. an order relation (transitive, reflexive and antisymmetric relation) such that, for each pair of concepts in the graph, there exists both a lower bound and an upper bound.

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

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

These two functions f and g defined between objects and attributes form a *Galois correspondence*. 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 $C = (O, I, R)$ (or $C = (O, I, (f, g))$) represented by a table. Table 1 gives an example of a formal context composed of a set of ten objects described by six attributes (a_1, a_2, b_1, b_2, c_1 and c_2). Additional information (class, feature and interval) is given in italics; for more details about this additional information please refer to Sec. 3.1.

A *formal concept* represents maximal objects-attributes correspondences (following relation R) by a pair (A, B) with $A \subseteq O$ and $B \subseteq I$, which verifies $f(A) = B$ and $g(B) = A$. The whole set of formal concepts thus corresponds to all the possible maximal correspondences between a set of objects O and a set of attributes I .

Two formal concepts (A_1, B_1) and (A_2, B_2) are in relation in the lattice when they verify the following inclusion property:

$$(A_1, B_1) \leq (A_2, B_2) \iff \begin{cases} A_2 \subseteq A_1 \\ \text{(equivalent to } B_1 \subseteq B_2) \end{cases}$$

The whole set of formal concepts fitted out by the order relation \leq is called *concept lattice* or *Galois lattice* because it verifies the lattice property: the relation \leq

Table 1. Example of formal context and (in italics) the classes of the objects and the features and intervals defining the attributes (for more details about the information in italics please refer to Sec. 3.1).

		Attributes					
		<i>Feature f_1</i>		<i>Feature f_2</i>		<i>Feature f_3</i>	
<i>Class</i>	<i>Id</i>	<i>a_1</i> [0-3]	<i>a_2</i> [6-20]	<i>b_1</i> [0-4]	<i>b_2</i> [12-20]	<i>c_1</i> [0-2]	<i>c_2</i> [11-20]
<i>1</i>	1	X		X			X
	2	X		X			X
<i>2</i>	3	X			X		X
	4	X			X		X
	5	X			X		X
<i>3</i>	6		X		X		X
	7		X		X		X
	8		X		X		X
<i>4</i>	9		X	X		X	
	10		X		X	X	

is clearly an order relation, and for each pair of concepts (A_1, B_1) and (A_2, B_2) , there exists a greatest lower bound (resp. a least upper bound) called *meet* (resp. *join*) denoted $(A_1, B_1) \wedge (A_2, B_2)$ (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 $\perp = (O, f(O))$ (resp. $\top = (g(I), I)$). The *Hasse diagram* of a graph³ is the suppression on the graph of both transitivity and reflexivity edges.

Figure 1 shows an example of concept lattice (represented by its Hasse diagram) built from the formal context in Table 1. For more information, the reader can refer to Ref. 33.

2.2. Generation algorithms

Numerous generation algorithms for concept lattices have been proposed in the literature.^{5,7,12,15,23,24,29,32} Although all these algorithms generate the same lattice, they propose different strategies. Some of these algorithms are incremental.^{7,15,23} Ganter’s NextClosure¹² is the reference algorithm that determines the concepts in lexicographical order (next, the concepts may be ordered by \leq to form the concept lattice) while Bordat’s algorithm⁵ is the first algorithm that computes directly the Hasse diagram of the lattice. Recent work¹⁴ proposed a generic algorithm unifying the existing algorithms in a unique framework, which facilitates the comparison of these algorithms. A formal and experimental comparative study of the different algorithms has been published.¹⁸

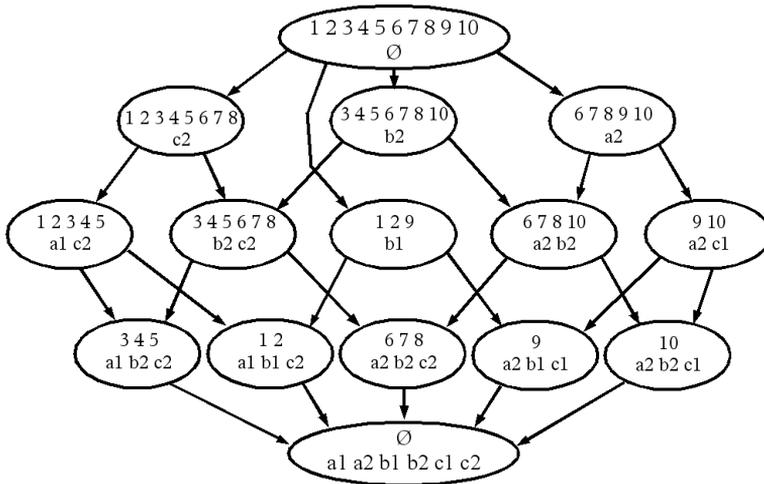


Fig. 1. Example of concept lattice (Hasse diagram).

All of these proposed algorithms have a polynomial complexity with respect to the number of concepts (at best quadratic in Ref. 24). The complexity is therefore determined by the size of the lattice, this size being bounded by $2^{|O+I|}$ in the worst case and by $|O+I|$ in the best case. Studies on average complexity are difficult to perform because the size of the concept lattice depends both on the dimensionality of the data to classify and on their organization and diversity. However, in practice, the size of the Galois lattice generally remains reasonable, as shown in various experiments.^{20,22}

In Ref. 1, we introduced an extension of Bordat's algorithm, which has the advantage of enabling on-demand concept generation. With this algorithm, only the small portion of the lattice that is necessary for our particular classification task is constructed during the recognition stage. This leads to a drastic decrease in the complexity of the generation algorithm, as shown in Sec. 4.2.2, and is useful in many contexts where incrementality is needed, or where the learning set is different from the gallery. In the latter case, we can imagine a system where discretization is performed offline using a generic learning set, and then the set of symbols to recognize (gallery) is given online to the system during the recognition stage. For instance, we can imagine a generic symbol recognizer which has to be specialized online to a given set of symbols (architectural symbols, road-signs...), the specialization including the generation of the Galois lattice using the objects in the gallery and the attributes obtained after discretization (performed using the learning set), and the recognition of the query symbols itself.

2.3. Application to data mining

As most classification methods, both the selection-based and navigation-based approaches rely on the three following stages: data preparation, learning and classification itself, which are detailed in the following parts of this section.

2.3.1. Data preparation

The first step is feature extraction. In the context of graphic objects recognition, many different primitives may be extracted. The Galois lattice is defined only for primitives that can be organized using *formal contexts*, i.e. for discrete data.

Continuous-valued primitives must therefore be partitioned into a finite set of disjoint intervals (called attributes) which are referred to by using codes. This procedure is commonly called *discretization*. Discretization methods may be classified according to three criteria¹¹:

- **supervised/unsupervised:** while unsupervised discretization techniques only use similarity between objects, supervised discretization methods also take into account the classes of the objects;
- **global/local:** the data space may be partitioned into intervals *before* the construction of the classifier (global discretization), or *as the construction of the classifier goes along* (local discretization);

- **mono dimensional/multidimensional:** while mono dimensional discretization processes each primitive independently from the others, multidimensional discretization simultaneously uses all the primitives to partition the data space into intervals. The main advantage of the latter technique is that it is capable of taking into account the interactions between primitives. However, mono dimensional discretization methods are the most widely used.

An experimental comparison of the effectiveness of various discretization techniques for classification is provided in Ref. 11. The experimental results show that supervised discretization techniques slightly outperform unsupervised discretization methods for a classification task.

Each of the global and local discretization methods has its own advantages and drawbacks. While local discretization has the advantage of taking into account the interactions between primitives, global techniques are more efficient because they process a feature space with lower dimensionality. The experimental comparisons provided in Refs. 11 and 26 do not settle the question of which strategy is the best for all circumstances; the choice of the technique is strongly related to the objective.

Depending on the application, the primitives may be continuous-valued and/or discrete so the discretization stage is not described for every method in the literature. In particular, the choice of the discretization strategy is not specified for the selection-based approaches described in Sec. 1. The discretization method we propose for the Navigala approach is detailed in Sec. 3.1.

2.3.2. Learning

During the learning stage, the Galois lattice will be constructed as a classifier from the set of discrete (or discretized) training data. For Galois lattice-based classification, the learning stage is *supervised* and therefore the training data consists of training objects primitives labeled by their associated classes (desired outputs). Preliminary to the training stage, we consider that the training data has been prepared (i.e. continuous-valued primitives have been discretized). The different steps that are carried out during the learning stage depend on the type of classification method.

For selection-based classification strategies, the learning stage includes three steps:

- The *lattice generation* step and, possibly, a pruning step. Different generation algorithms are described in Sec. 2.2;
- The *selection* step. The objective is to reduce the learning space using different relevance criteria, such as the occurrence frequencies of the different attributes. The selection step may lead to filtering out concepts,^{7,19,25,27,28} objects²¹ and/or contextual rules³⁴;
- Possibly the classifier's learning stage (e.g. the extraction of classification rules for the GRAND²⁵ and RULEARNER²⁷ methods).

The learning stage of navigation-based classification methods, detailed in Sec. 3.2, only involves two steps:

- The *lattice generation* step;
- The *labeling* step: the nodes which are pure enough are labeled with their corresponding class.

We can note that the lattice generation step may lead to a high complexity (exponential complexity in the worst case). That drawback is counterbalanced by the fact that the learning stage is offline and can be carried out before classification itself in most applications. In some applications where the learning step cannot be performed offline (for examples see Sec. 2.2), on-demand generation may be used (see Sec. 3.3.2). Changes in the training set generally lead to a new learning stage, even though some incremental solutions exist (see Sec. 2.2).

2.3.3. Classification

Once the classifier has been built from the training data, one can classify new samples. The aim is to classify these new elements on the basis of their description (primitives values). Differing from the learning stage which is generally performed offline, the classification stage is generally performed online.

Selection-based methods rely on classical classifiers such as the k-nearest neighbors or Bayesian classifiers.

Conversely, in navigation-based approaches, classification is based on the use of the whole Galois lattice. This step is of very low complexity (for more information about the computational times please refer to Sec. 4.2.2). Each object to be recognized (denoted by $p \in A$) progresses through the lattice from \perp to \top (see Sec. 2.1), moving from a formal concept to one of its successors (connected by an edge), until it reaches a labeled concept. At each concept $C_i = (A_i, B_i)$, the choice of the following concept $C_{i+1} = (A_{i+1}, B_{i+1})$ is made among its direct successors according to the set of attributes $x \in I$ where pRx and $x \in C_{i+1} \setminus C_i$. We must note that at each step, the choice of the successor concept is unique thanks to the inclusion property (see Sec. 2.1).

2.4. Comparison with a classification tree

At this point, the reader could naturally question the links between the decision tree and the Galois lattice. Indeed the navigation step is quite similar to the one proposed with a decision tree. The main difference lies in the existence of multiple paths to reach a given concept in the lattice, contrary to the decision tree where there is a unique path to reach a given node. This property confers flexibility to the recognition process using a lattice and therefore noise robustness is increased. Experiments (see Sec. 4.2.1) have shown that the navigation-based approach Navigala provides better recognition rates than decision trees in a context of noisy symbols recognition.

Moreover, we have recently shown the existence of structural links (inclusion and fusion) between a particular type of concept lattices and decision trees. For more details please refer to Ref. 2.

3. Description of the Proposed Approach: Navigala

We have developed a recognition system named Navigala (NAVIGATION into GALois LAttice), where classification is navigation-based. This method is fitted to recognize noisy graphical objects and especially symbol images. Such symbols appear in technical documents such as architectural plans or electrical diagrams. The possible origins of the noise are paper deterioration (stains, blotting out), scanning artefacts or vectorial distortions in the context of handwritten symbols (for examples of noisy symbols see Fig. 5).

Graphic objects may be described by *various types of primitives*. As statistical features describe the spatial distributions of the pixel values of the symbol, structural primitives describe the spatial or topological relations between certain subpatterns extracted from the symbol images. In the following, the primitive vector of each symbol is called the signature of this symbol.

Navigala is a supervised classification approach, whereas the discretization stage can be performed by using either a supervised or unsupervised criterion. In this section, we will describe the three steps of Navigala: data preparation, learning and classification. We will also provide a comparison of Navigala with the existing classification methods based on the use of a Galois lattice and mentioned in Sec. 1.

3.1. Data preparation

Firstly, several signatures are extracted from the symbol images: statistical signatures (Fourier–Mellin invariants,¹⁰ Radon transform-based Radon transform,³⁰ Zernike moments³¹), and a structural signature named *flexible structural signature*.⁸

As presented in Sec. 2.3.1, the continuous valued primitives must be discretized in a preprocessing stage. Let us consider that the dataset is represented by an array of data where each row corresponds to the feature vector of one symbol image and every column corresponds to the (continuous) values of a given primitive f_i in the feature vector. The objective of the discretization stage is to obtain a formal context (as illustrated in Table 1) where each column (attribute) corresponds to an interval that separates the images corresponding to different classes (symbols). For example, in Table 1, the images of the first two symbols (classes 1 and 2) differ following the values of their second feature f_2 : while for the images of the first symbol $f_2 \in [0, 4]$, the images of the second symbol verify $f_2 \in [12, 20]$.

Discretization is performed as follows. Initially, we consider that every column f_i in the data array is described by one interval V_i , the lower and upper bounds of which are respectively the minimum and maximum values in the corresponding column. At each iterative step of the discretization process, a criterion selects both the primitive

to split into intervals and the optimal cutting point. At iteration t , let $x \in I$ be a primitive interval, where $V_x = (v_1, \dots, v_n)$ are the values of x observed in the training set and sorted by ascending order. The interval will be cut between the values v_j and v_{j+1} , where v_j maximizes a given “cutting” criterion $C(v_j)$. Numerous cutting criteria can be proposed; these criteria may be supervised or unsupervised, global or local and multidimensional or mono dimensional (see Sec. 2.3.1). The supervised, global and mono dimensional criteria are among the most widely used. We experimented three global and mono dimensional criteria (see Eqs. (3)–(5)): maximal distance, entropy and Hotelling’s coefficient.¹⁷ While maximal distance is an unsupervised criterion that aims at maximizing the gap between two consecutive values, entropy and Hotelling’s coefficient are two supervised criteria which respectively minimize the degree of mixture of the classes and jointly maximize the scatter between classes while minimizing the within-class scatter.

- maximal distance:

$$C_{MD}(v_j) = v_{j+1} - v_j \tag{3}$$

- entropy:

$$C_E(v_j) = E(V_x) - \left(\frac{j}{n} E(v_1, \dots, v_j) + \frac{n-j}{n} E(v_{j+1}, \dots, v_n) \right) \tag{4}$$

with $E(V) = - \sum_{k=1}^{|c(V)|} \frac{n_k}{n} \log_2 \left(\frac{n_k}{n} \right)$, where n and $c(V)$ are respectively the number of images and the set of classes (symbols) corresponding to the values belonging to interval V (in the training set). n_k is the number of images, among the n images with values in V , which belong to class k .

- Hotelling’s coefficient:

$$C_H(v_j) = H(V_x) - \left(\frac{j}{n} H(v_1, \dots, v_j) + \frac{n-j}{n} H(v_{j+1}, \dots, v_n) \right) \tag{5}$$

where $H(V) = \frac{\sigma_B(V)}{\sigma_W(V)}$.

With $\sigma_B(V) = \frac{1}{n} \sum_{k=1}^{|c(V)|} n_k (g_k - g)^2$ is the between-class variance and $\sigma_W(V) = \frac{1}{n} \sum_{k=1}^{|c(V)|} n_k (\sum_{i=1}^{n_k} (v_{k_i} - g_k)^2)$ is the within-class variance, where $g = \frac{1}{n} \sum_{j=1}^n v_j$ is the mean of the values belonging to V , v_{k_i} is the i th value in V corresponding to class k and $g_k = \frac{1}{n_k} \sum_{i=1}^{n_k} v_{k_i}$ is the mean of the values of images from class k and belonging to V .

The discretization process is iterated until a *stopping criterion* is met. In Navigala, the stopping criterion is class separation, which is met when each set of images sharing the same attributes is classified into one given class. In some cases where class separation cannot be achieved, we stop the discretization process when Hotelling’s cutting criterion is less than a certain predefined threshold.

In Navigala, the obtained intervals are then extended as fuzzy intervals, to be more robust towards noise. During the classification stage, each query symbol image

will be considered as corresponding to its set of nearest fuzzy intervals in the feature space.

The distance $d(f_i, V)$ between the value f_i of the i th element in the query signature and an interval V obtained from the discretization of f_i can be expressed as:

$$d(S, V) = d(f_i, V) = 1 - \mu(f_i, V)$$

where $\mu(f_i, V)$ is the *membership functional* that specifies the level of membership of $f_i \in V$.

We propose the extension of an initial interval $[b, c]$ to a fuzzy number (described by a trapeze $[a, b, c, d]$ with $[a, d]$ as support and $[b, c]$ as kernel, see Fig. 2) by taking into account both the closest intervals and the objects distribution in the interval:

$$\begin{aligned} a &= b - \theta \times \min(d_{V^-}, d(g, c)) \\ d &= c + \theta \times \min(d_{V^+}, d(b, g)) \end{aligned}$$

where d_{V^-} (resp. d_{V^+}) is the distance with the closest previous interval (resp. closest next interval); g is the gravity center of the values in the initial interval $[b, c]$ and θ is a fuzzy parameter. Distances d_{V^-} and d_{V^+} are necessarily positive since intervals are disjoint. Each interval has at least one neighbor interval since undiscretized primitives are not selected. In the special case where the current interval has only one neighbor, we replicate that distance so as to obtain a symmetrical fuzzy interval.

3.2. Learning

The discretized data (issued from the data preparation stage) will then be used as a training set for the Galois lattice construction. Our algorithm¹ is an extension of Bordat’s algorithm⁵ which computes directly the Hasse diagram of the lattice (see Sec. 2.2). Indeed, during the classification, we use the successor relation to navigate through the graph, so we have to compute the successors of a given concept starting with the bottom concept.

Once the Hasse diagram of the discretized data is computed, we can label its concepts by using the classes in the training set. The terminal concepts (direct successors of the minimal boundary, located at the bottom of the Galois lattice) are labeled by using the formal context used for its generation (for an example of a formal context see Table 1). To each terminal concept, we associate the class that is most frequently associated to its objects (symbol images) in the training set. The labels associated to the terminal concepts will be used during the classification stage.

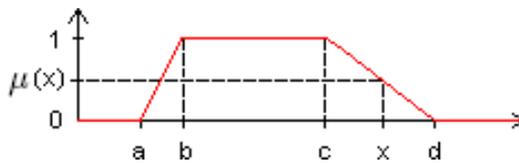


Fig. 2. Example of a fuzzy number.

3.3. Classification

3.3.1. Navigation

New symbols can be classified by using the Hasse diagram of the Galois lattice. Classification is performed by using the feature vector of the query symbol and navigating through the graph, from the minimum concept \perp until a terminal (labeled) concept is reached. At each step of this navigation stage, the nearest fuzzy interval is selected (according to a fuzzy distance and a choice criterion). Intuitively, during the progression of a query image, the description of the query object is refined, until it is considered similar enough to a given set of objects belonging to the same terminal concept. When the query symbol reaches a terminal concept, it is labeled with the corresponding class.

More formally, at each step, given the current concept (A, B) , one of its direct successors $(A_1, B_1), \dots, (A_m, B_m)$ in the lattice is selected by validating one (or more) fuzzy intervals. Each set of attributes B_i corresponds to a set of intervals containing the set of intervals B : $B \subset B_i \forall i = \{1, \dots, m\}$, because the concept (A_i, B_i) is a successor of the concept (A, B) in the lattice. At the current concept (A, B) , all the intervals in the interval set B have been validated. Let us isolate, for each successor concept (A_i, B_i) , the set of intervals \tilde{B}_i that are candidates for validation (the set of intervals that have not been previously validated in (A, B)):

$$\tilde{B}_i = B_i - B$$

Let us further denote by \tilde{B} the family of sets of intervals which are candidates for validation:

$$\tilde{B} = \bigcup_{i=1}^m \tilde{B}_i$$

The navigation elementary step therefore consists in selecting a set of intervals \tilde{B}_i among the family of candidate interval sets \tilde{B} . This selection is performed according to a *choice criterion* defined using a fuzzy distance $d(S, \tilde{B}_i)$ between the signature S of the query object and the candidate sets of intervals \tilde{B}_i , for every candidate successor concept (A_i, B_i) .

We have to define a choice criterion to select, among the candidate sets of intervals $\tilde{B} = \cup_{i=1}^m \tilde{B}_i$ (corresponding to the successors of the current concept), the set of intervals \tilde{B}_i that best correspond to the signature S of the query object. The choice criterion relies on the use of the fuzzy distances $d(S, \tilde{B}_i)_{i=\{1, \dots, m\}}$ between the signature S and the candidate sets of intervals \tilde{B}_i . Several choice criteria are possible, hereafter is a (nonexhaustive) list of these criteria:

- (1) Choosing i where the sum of the distances between the signature S and the intervals V constituting the set of intervals \tilde{B}_i is minimum. More formally,

$$i = \text{Argmin}_{i=1, \dots, m} (\sum_{V \in \tilde{B}_i} d(S, V))$$

- (2) Choosing i where the set of intervals \tilde{B}_i contains the maximum number of intervals among the k nearest intervals from the signature S (according to the fuzzy distance measure d). More formally,

$$i = \operatorname{Argmax}_{i=1,\dots,m} \left| \tilde{B}_i \cap \{V^{(1)}, \dots, V^{(k)}\} \right|$$

where the $V^{(j)}$ are the intervals in \tilde{B} , sorted by descending order following the distance $d(S, V)$ and k is a parameter of the choice criterion.

- (3) Choosing i where the set of intervals \tilde{B}_i contains the maximal number of intervals located at a distance inferior to the threshold c for the given query signature S . More formally,

$$i = \operatorname{Argmax}_{i=1,\dots,m} |\{V \in \tilde{B}_i \text{ such that } d(S, V) \leq c\}|$$

We can note that the first criterion, defined globally on all the intervals contained in \tilde{B}_i , has the drawback of “swallowing up” the noise. The second criterion relies on the principle of the k -nearest neighbor rule. We can also note that the third criterion is a particular case of the second criterion. All of these proposed criteria being local for each $i = 1, \dots, m$, one can define more sophisticated criteria in order to benefit from the advantages of the different alternatives. In our case, we chose to use a combination of these criteria, which consists in:

- Applying criterion (3) with $c = 0$, which is equivalent to defining, for every interval V in \tilde{B}_i , a rectangular fuzzy number whose support is defined by the boundaries of V .
- Then, in case of an ambiguity, we apply criterion (3) with $0 < c < 1$. The support of the fuzzy number is extended to the fuzzy boundaries of the fuzzy interval V proportionally to its size.
- If the ambiguity remains, we apply criterion (1), which is equivalent to a symmetrical fuzzy number whose zero (center, gravity center or median) is the center of the interval.

3.3.2. On-demand concepts generation

The Galois lattice construction algorithm used for Navigala¹ presents several advantages: it is quite easy to implement, and it enables an *on-demand concepts generation* of the Galois lattice: concepts are generated only when they are proposed for selection during the recognition process. This is interesting, especially in some applicative contexts where the graph cannot be constructed offline (examples of such applications are given in Sec. 2.2). Indeed, it avoids the construction of the whole graph, which can be of an exponential complexity in the worst case. Indeed, recognition is performed by exploring only a small region of the lattice. As shown in Sec. 4.2.2, it leads to a slight increase in the complexity of the classification step but it considerably reduces the complexity of the learning stage.

3.3.3. Iterative classification

In the field of symbols classification, we also developed an *iterative recognition* system (see Ref. 16), which takes advantage of the complementarity of statistical and structural approaches. Indeed, this method can integrate several descriptions of various types for a more effective classification.

During navigation in the Galois lattice, in the case of uncertainty regarding the symbol to be recognized, it is possible to stop the progression and thus avoid certain classification errors. For example, let C_1 and C_2 be two successor concepts of the current concept C , where C_1 and C_2 contain objects of different classes whose descriptions are very similar to the query object. To avoid any doubt, the descriptions of the objects in C_1 and C_2 can be replaced by new descriptions issued from another type of feature extractor. In the iterative process, these new descriptions are then used to build a new Galois lattice especially designed to discriminate the objects from concepts C_1 and C_2 according to their classes.

3.4. Comparison with other Galois lattice-based methods

This section is dedicated to a comparison between selection-based methods and Navigala: a synthesis of the similarities and the differences between the various approaches is provided. For an experimental comparison see Sec. 4.3.

Figure 3 provides a comparison of the different classification methods based on a Galois lattice. Selection-based methods can be gathered depending on the elements used: concepts only, concepts and rules, concepts and prototypes or rules only. The Navigala approach is characterized by the use of the whole Galois lattice with an object classification by navigation.

Moreover, Table 2 proposes a comparison of the computational complexities of the construction stages of these different methods, and a synthesis of the experimental results obtained by the authors of those methods. In this table we have added the characteristics of Navigala. We can see that Navigala's complexity is very low compared to other lattice-based methods, especially when our applicative context enables the use of on-demand generation. We can see that its experimental results are encouraging.

In the following, we discuss the behavior of these eight methods when classes are weakly represented, in the presence of noise, and when the number of classes is large.

3.4.1. Weakly represented classes

In most selection-based approaches, the learning stage is limited to the most represented objects (in the learning set). That is why, with LEGAL-E for example, some objects may not be recognized even though they are very similar to a learning sample (if this learning sample is not representative enough of the learning set to be learnt). The opposite of this is Navigala, where the whole learning set of objects is learnt without favoring the most represented, which enables us to be exhaustive. However,

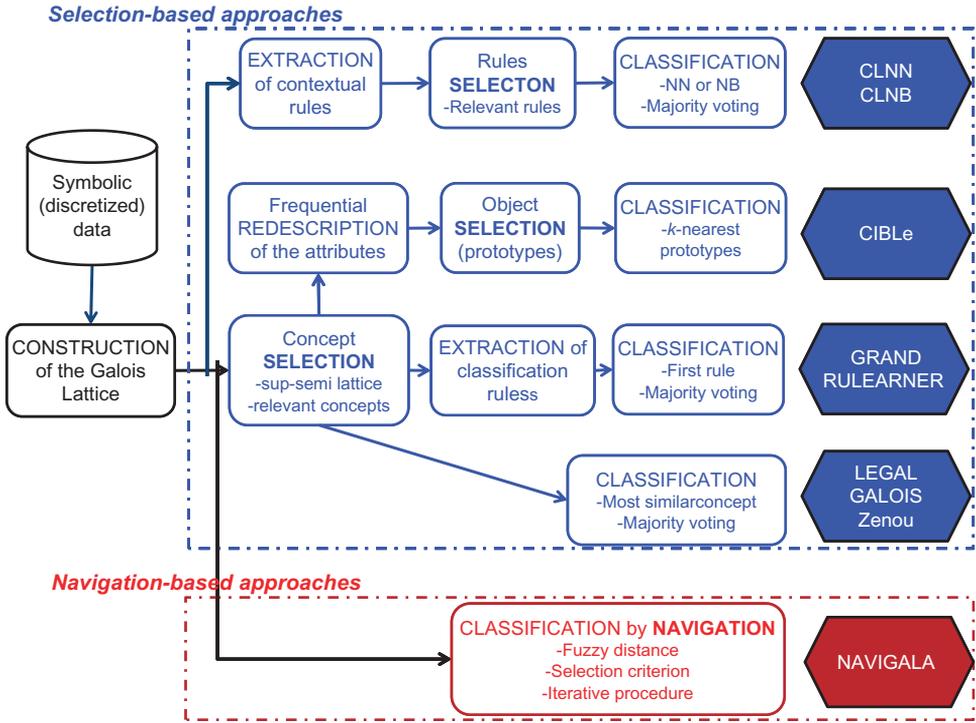


Fig. 3. Comparison of some Galois lattice-based classifiers. The acronym NN stands for “Nearest Neighbor”, while NB stands for “Naive Bayes”.

Table 2. Synthesis of the properties of the classification methods based on lattices.

Methods	Construction Complexity	Experimental Comparison
GRAND	$O(2^{l^4})$ with l the minimum between the number of examples and the number of attributes	Performances similar to Assistant, AQ15, AQR, Bayes and CN2
LEGAL	$O(L n(1 - \alpha))$ with $ L $ the number of concepts in the lattice	Performances similar to GLUE and superior to C4.5
GALOIS	$O(3^m 2^m n) < O(3^{2m} n)$ with m the number of attributes and n the number of examples	Performances similar to other methods in the literature
RULEARNER	idem GRAND	Performances similar to C4.5 and CN2, or even slightly better
CIBLe	$O(\min(n - 1, m - 1)^{h+1} m^3)$ (sup-semi lattice construction) + $O(\min(n - 1, m - 1)^{h+1})$ (threshold search)	Better performances than IBI, K^* and Pebls
CLNN and CLNB	$O(L E ^3 + L m + L') + O(L' m)$. $O(NN/NB)$ with $ E $ the maximal number of successors of a concept $ L' < L $ et $O(NN/NB)$ the complexities of NN or NB classifiers	Better performances than NBTree, CBA and C4.5 Rules

Table 2. (Continued)

Methods	Construction Complexity	Experimental Comparison
Zenou	$O(3^m 2^m n + L ^2 m + L m)$	Encouraging performances
Navigala	$(O(L n^3)$ optional lattice +) $O(nm^2)$ (classification)	Performances similar to Bayesian classifier and greater than CART (see Sec. 4.2.1)

while selection-based methods enable outlier detection (and further suppression), Navigala cannot detect them and they will be integrated in the Galois lattice. Nonetheless, Navigala is designed to be robust enough to accommodate these outliers, as detailed in the following section.

3.4.2. Noise robustness

The navigation enables avoiding the influence of a noise carried on several attributes. Indeed, the attributes are successively validated, as opposed to selection-based approaches where the validation is given by an average. Moreover, the validation order of the attributes is modifiable depending on their robustness to noise. The most represented attributes are proposed at the beginning of the navigation within the lattice and the frequency decreases during the progression within the graph. Finally, the fuzzy distance measure softens the interval boundaries and absorbs the disturbances due to noise. Noise robustness is a problem for a selection-based approach: while LEGAL-E resists quite well to noise using the validity quasi-coherence criteria, the thresholds' choice of validity and quasi-coherence can require considerable working time.¹⁹

3.4.3. Large number of classes

Some selection algorithms are not designed to manage a large number of classes. For instance, CIBLe has difficulties characterizing data containing a large number of classes especially with complex data.²¹ With navigation, it is possible to perform classification at different levels, using different signature types (using iterative classification, see Sec. 3.3.3) and therefore to discriminate between a higher number of classes.

4. Experimental Results

In this section, we present various experimental results. Firstly, we study the effects of variations in the parameters required for Navigala (for a symbol recognition task). Second, we provide a comparative study of Galois lattice selection-based methods and Navigala.

4.1. Setting the parameters of Navigala for symbol recognition

The main objective of this first experimental study is to tune the parameters of the proposed approach for a symbol recognition task. For experimentations, we use two

different symbols image databases: GREC 2003^a and GREC 2005 (Graphics RECOgnition).^b These databases were developed for international symbol recognition contests organized by the IAPR TC 10 committee.^c

Each database contains one *model* symbol image per class (where the model image does not contain any noise) and noisy versions of these model symbols. The noise mimics deteriorations generated when scanning or copying paper documents. GREC 2003 database contains 35,139 symbol images from 39 classes, with exactly 901 symbol images per class. Each class contains one model symbol and 900 noisy symbols (ten symbol images for each of the nine types of deterioration). GREC 2005 database contains 175 symbol images from 25 classes (one model symbol and a varying number of noisy symbols per class). The noisy symbols are distinguished among six types of deterioration. Figures 4 and 5 respectively show samples of model symbols and noisy symbols extracted from these two databases.

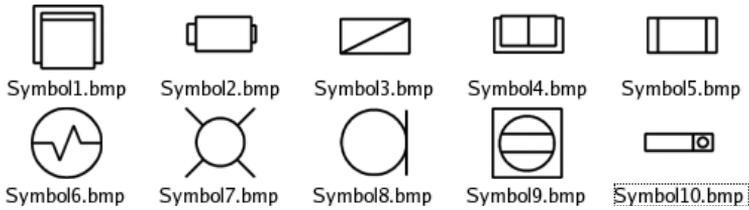


Fig. 4. Ten examples of model symbols (without noise) from GREC 2003.

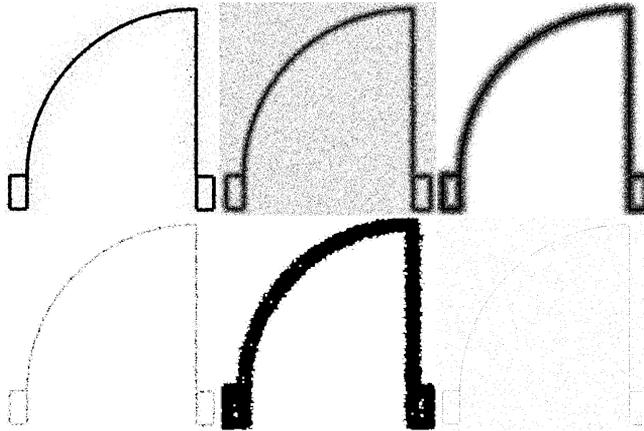


Fig. 5. Six examples of noisy symbols of the GREC 2005 database.

^awww.cvc.uab.es/grec2003/SymRecContest/index.htm

^b<http://www.cs.cityu.edu.hk/grec2005>

^c<http://iapr-tc10.univ-lr.fr>

4.1.1. Cutting criterion

In this experiment, we choose to test the adequacy of the cutting criteria (maximal distance, entropy or Hotelling’s coefficient) to our recognition system (see Sec. 3.1).

To evaluate these criteria, we use a subset of the GREC 2003 database. Learning is performed by using only ten symbols per class and performance evaluation is made by using 90 noisy symbols per class. Figure 6 provides the recognition rates and Fig. 7 gives the size of the Galois lattice for each cutting criterion. The three statistical signatures presented in Sec. 3.1 are studied: Fourier–Mellin invariants, R-signature (Radon) and Zernike moments.

From these experimental results, we show that Hotelling’s coefficient almost always provides the best results, no matter which signature is used. Moreover, we can see that the size of the lattice can explode using the maximal distance criterion as shown in Fig. 7. For the sake of effectiveness and efficiency, we therefore chose to use Hotelling’s coefficient criterion.

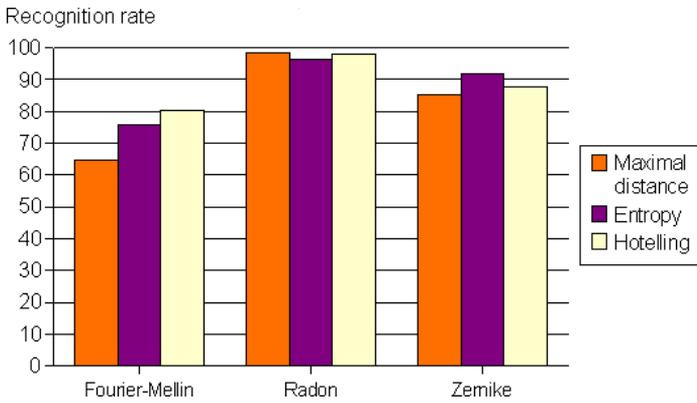


Fig. 6. Recognition rates depending on the cutting criterion.

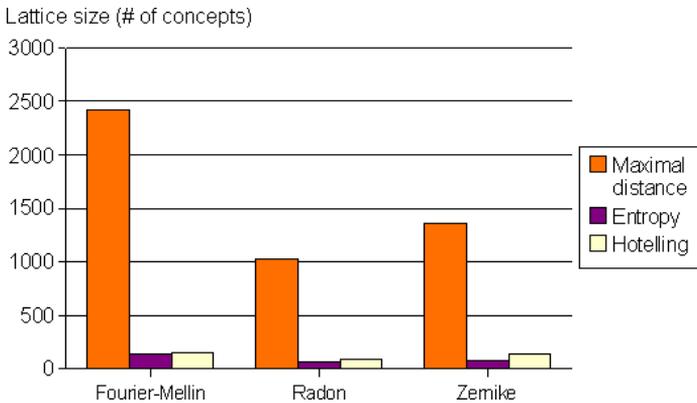


Fig. 7. Number of concepts in the Galois lattice depending on the cutting criterion.

4.1.2. Signatures comparison

In this subsection, we compare the effectiveness of certain statistical and structural signatures for our recognition system Navigala. We use a subset of the GREC 2003 database. The learning set is composed of eight classes (ten symbols per class) and performance characterization is performed by using a test set containing 90 noisy symbols per class. The recognition rates are presented in Fig. 8, and Galois lattice sizes in Fig. 9.

From these figures (and we can consider additionally Figs. 6 and 7) we can see that the R-signature (Radon) is the most interesting option both in terms of recognition rates and lattice size.

4.2. Performance characterization for symbol recognition

This section shows the results of different experiments.

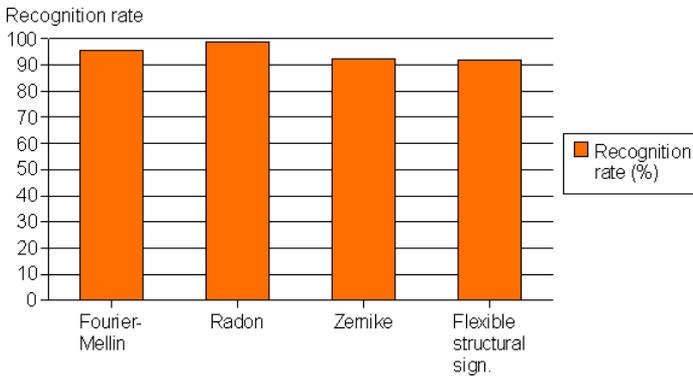


Fig. 8. Recognition rates depending on the signature.

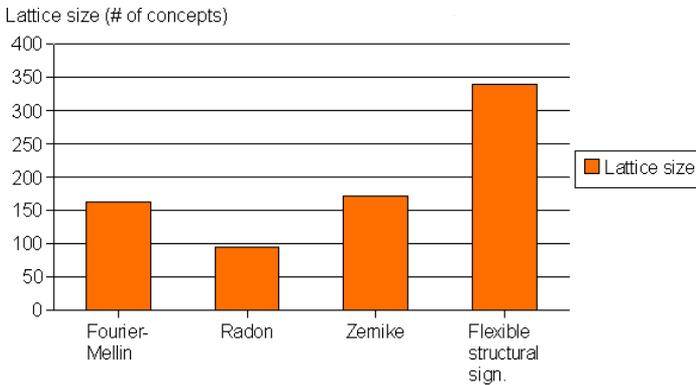


Fig. 9. Number of concepts in the Galois lattice depending on the signature (obtained by using the Hotelling cutting criterion).

Firstly, we compare the performances of Navigala and other standard classification approaches. Navigala is compared to a probabilistic classifier (Bayesian classifier), a statistical classifier (SVM) and a symbolic classifier (decision tree).

Secondly, we provide computational results when using on-demand generation algorithm.

4.2.1. Comparison with other standard classification approaches

In this subsection we show the results of two experiments where the performance of the proposed approach is compared to other standard classification approaches.

In the first experiment, we compare the recognition rates obtained using Navigala with those of the naive Bayesian classifier and a SVM classifier. We consider a dataset composed of symbol images from the two GREC databases: we use two sets of ten classes of symbols from the GREC 2003 database (named cl1-10 and cl11-20) and one set of 25 classes of symbols from the GREC 2005 database (named cl1-25), where the noise is stronger. The symbols are described by the statistical Radon signature (R-signature) composed of 50 values.

The classifiers are evaluated using cross-validation with varying sizes of the learning and test data: 5 blocks of 182 symbols from GREC 2003 (Test1), 10 blocks of 91 symbols from GREC 2003 (Test2), 26 blocks of 35 symbols from GREC 2003 (Test3), and 5 blocks of 35 symbols from GREC 2005 (Test4). The average recognition rates are given in Fig. 10.

While the naive Bayesian classifier is more effective than Navigala in the presence of only ten classes and of a limited noise (Test1 and Test2), Navigala outperforms the Bayesian classifier in more difficult situations where the number of images is increased (Test3) or where the noise is significant and the number of classes to be discriminated is increased (Test4). As a consequence, we can consider that our approach is more robust towards noise and towards an increase of the number of

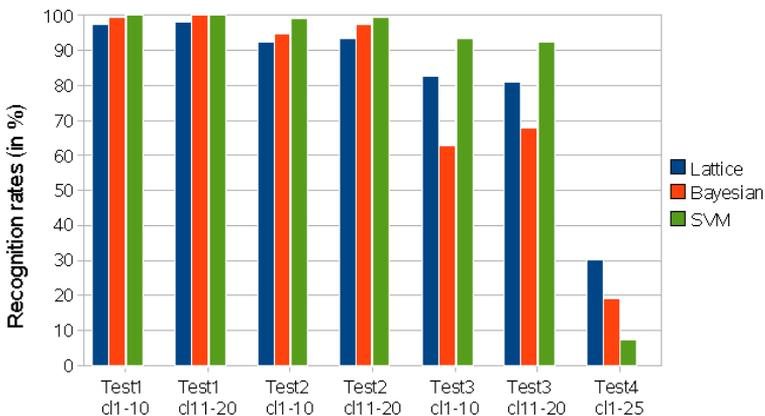


Fig. 10. Compared recognition rates of the four classifiers using cross-validation.

classes than the Bayesian classifier. In the experiment Test4, our approach even outperforms the SVM classifier.

Furthermore we can note that, Navigala performs feature selection prior to classification (during the discretization stage) and only uses 6 to 15 of the 50 elements of the feature vectors whereas the other classifiers use all of the 50 elements. For these reasons, we can consider that these results are encouraging.

In the second experiment, we compare the performances of Navigala and decision trees. The experimental protocol is the following. We consider ten classes from GREC 2003 dataset (10 model symbols for learning and 900 noisy symbols for test). This data is prepared as presented in Sec. 3.1: the Radon signature is extracted from the images, and then the signatures are discretized by using the proposed discretization approach (based on the Hotelling coefficient). From the discretized training signatures we generate both the concept lattice (the Navigala classifier is then built) and the decision tree using CART algorithm.⁶

The recognition rates we obtain are 57% for the decision tree versus 72% for the lattice. One of the main differences between a concept lattice and a decision tree is that in decision trees the path from the root to a given leaf is unique whereas in Galois lattices there are multiple paths from the maximal boundary to a given terminal concept (see Fig. 1). The improvement of the recognition rates when using the lattice shows that the existence of multiple paths gives the lattice better noise robustness.

In return, the size of the lattice is generally greater than the size of the decision tree. In our experiment, the number of discretization steps is 9 and the number of discretized intervals is 17. The number of concepts in the lattice is 70 against (only) 18 nodes for the decision tree. Thus, we can see that, when using the same discretized data, the size of the lattice is greater than the size of the decision tree. But this drawback may be counterbalanced by the possibility of generating the lattice on-demand.

4.2.2. On-demand generation

As presented in Sec. 3.3.2, the Galois lattice generation algorithm used for Navigala¹ enables an *on-demand concepts generation* of the Galois lattice. Recognition is performed by exploring only a small region of the lattice. The experimental results presented in Table 3 show the processing times for the learning and classification steps when using a 1.83 GHz processor with 512 MB RAM. It also gives the number of generated concepts. The learning set is composed of 25 model symbols (one per

Table 3. Processing times (using a 1.83 GHz processor with 512 MB RAM) and number of concepts generated.

	Learning	Classification	Number of Concepts
Whole lattice	430.2 sec	2 sec	3185
On-demand generation	0.5 sec	9.8 sec	282

each of the 25 classes) from GREC 2003. The test set is composed of 250 noisy symbols: ten symbols per class.

From this table we can see that the number of concepts generated on-demand (282) is significantly reduced compared to the construction of the whole lattice (3185), while the recognition (navigation) path is identical. Therefore, we can note that on-demand generation gives the same performances as offline generation of the lattice and reduces the size of the structure to be generated. Table 3 also shows that when using on-demand generation the computational cost of the generation step is partially moved from the learning stage to the classification stage (compared to offline generation). Nevertheless, the computational time is globally reduced while remaining reasonable.

4.3. Comparison with other Galois lattice-based methods

This section is dedicated to experimental comparisons between selection-based methods and Navigala on certain databases from the UCI Repository⁴: Breast-cancer (BC), Iris (IR), Soybean-small (SS) and Zoo (ZO). Table 4 provides a description of these databases: number of records, number of continuous attributes, number of discrete attributes and number of classes to distinguish.

We consider experimental results available in the papers describing the methods: RULEARNER,²⁷ CIBLE,²¹ CLNB — CLNN and C4.5Rules.³⁴ This is why our experimentation results are not exhaustive. Table 5 gives the classification error rates obtained using cross-validation.

Table 4. UCI Repository databases.

Database	Number of Objects	Number of Attributes		Number of Classes
		Continuous	Discrete	
BC	699	9	0	2
IR	150	4	0	3
SS	47	0	35	4
ZO	101	1	15	7

Table 5. Results obtained using cross-validation on certain databases of the UCI Repository.

DB	Cross-Validation	Classification Error Rates				
		Navigala	Cible ²¹	CLNB ³⁴	CLNN ³⁴	C4.5R ³⁴
BC	10 fold	5.4%		3.1%	3.4%	5.0%
	5 fold	5.5%	4.6%			
IR	10 fold	7.4%		5.3%	5.3%	4.7%
	5 fold	4.1%				
SS	10 fold	2.5%				
	5 fold	2.3%	8%			
ZO	10 fold	4.0%		3.9%	3.9%	7.8%
	5 fold	4.9%	6.1%			

In general, Navigala provides classification error rates relatively close to those obtained by other classifiers. It has to be noted that Navigala catches up with and even outperforms the other methods when the number of classes is increased, as in the Soybean-Small and Zoo databases. Therefore, we can note that Navigala is somewhat generic, as it has been designed for a very specific task of symbol recognition using statistical (continuous) signatures, and can be successfully applied to other types of data.

5. Conclusion and Discussion

The two main contributions of this paper are: firstly, the introduction of a classification method named Navigala dedicated to noisy symbol recognition and its experimental assessment and secondly, a comparative study (both formal and experimental) of eight classification methods based on Galois lattices (including Navigala).

Contrary to most of the previously proposed approaches, which use the Galois lattice as a selection tool, Navigala classifies the symbols by navigating through the lattice. While most selection-based approaches are well-suited for data mining applications with little noise and a limited number of classes, Navigala is dedicated to a task of noisy symbol image recognition, where the number of classes may be huge. By using the whole lattice as a classifier, Navigala has the advantage of being exhaustive and proposing multiple paths to reach a given class-labeled concept, which makes Navigala more robust towards noise. It has to be noted that the inherent complexity is limited thanks to our on-demand generation algorithm.

We are now working on the structural links between Galois lattices and classification trees in order to propose a new classification method based on a Galois lattice with local discretization, similar to the discretization stage of decision trees.

Acknowledgments

Grateful acknowledgement is made for financial support by the Poitou-Charentes Region (France). We would also like to acknowledge Dr. Stéphanie Guillas for her work on this subject in the context of her PhDs.

References

1. K. Bertet, S. Guillas and J. M. Ogier, Extensions of Bordat's algorithm for attributes, in *Fifth Int. Conf. Concept Lattices and Their Applications (CLA'2007)* (Montpellier, France, October 24–26, 2007), pp. 38–49.
2. K. Bertet, M. Visani, J. M. Ogier and N. Girard, Some links between decision trees and dichotomic lattices, *Sixth Int. Conf. Concept Lattices and Their Applications (CLA'2008)* (Olomouc, Czech Republic, October 2008), pp. 193–205.
3. G. Birkhoff, *Lattice Theory*, Vol. 25, 3rd edn. (American Mathematical Society, 1967).
4. C. Blake, E. Keogh and C. Merz, UCI repository of machine learning databases, 1998.
5. J. P. Bordat, Calcul pratique du treillis de Galois d'une correspondance, *Math. Sci. Hum.* **96** (1986) 31–47.

6. L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees* (Wadsworth Inc., Belmont, California, 1984).
7. C. Carpineto and G. Romano, Galois: An order-theoretic approach to conceptual clustering, *Proc. Int. Conf. Machine Learning (ICML'93)* (Amherst, July, 1993), pp. 33–40.
8. M. Coustaty, S. Guillas, M. Visani, K. Bertet and J. M. Ogier, Flexible structural signature for symbol recognition using a concept lattice classifier, *Seventh IAPR Int. Workshop on Graphics Recognition (GREC'07)* (Curitiba, Brazil, September, 2007), pp. 20–21.
9. B. A. Davey and H. A. Priestley, *Introduction to Lattices and Orders*, 2nd edn., (Cambridge University Press, 1991).
10. S. Derrode, M. Daoudi and F. Ghorbel, Invariant content-based image retrieval using a complete set of Fourier–Mellin descriptors, *Int. Conf. Multimedia Computing and Systems (ICMCS'99)* (1999), pp. 877–881.
11. J. Dougherty, R. Kohavi and M. Sahami, *Supervised and Unsupervised Discretization of Continuous Features* (Morgan Kaufman, 1995).
12. B. Ganter, Two basic algorithms in concept analysis, *Technische Hochschule Darmstadt* (Preprint 831) (1984).
13. B. Ganter and R. Wille, *Formal Concept Analysis, Mathematical Foundations* (Springer-Verlag, Berlin, 1999).
14. A. Gely, A generic algorithm for generating closed sets of binary relation, *Third Int. Conf. Formal Concept Analysis (ICFCA 2005)* (2005), pp. 223–234.
15. R. Godin, R. Missaoui and H. Alaoui, Learning algorithms using a Galois lattice structure, *Third Int. Conf. Tools for Artificial Intelligence* (San Jose, California, 1991) pp. 22–29.
16. S. Guillas, K. Bertet and J. M. Ogier, Concept lattice classifier: A first step towards an iterative process of recognition of noised graphic objects, *Fourth Int. Conf. Concept Lattices and their Applications (CLA'2006)* (2006), pp. 257–263.
17. H. Hotelling, Relations between two sets of variates, *Biometrika* **XX-VIII** (1936) 321–377.
18. S. Kuznetsov and S. Obiedkov, Comparing performance of algorithms for generating concept lattices, *J. Exper. Theor. Artif. Intell.* **14**(2–3) (2002) 189–216.
19. E. Mephu Nguifo, Galois lattice: A framework for concept learning. design, evaluation and refinement, *Proc. IEEE Int. Conf. Tools with Artificial Intelligence (IEEE-ICTAI-94)* (New-Orleans, November 1994), pp. 461–467.
20. E. Mephu Nguifo, V. Duquenne and M. Liquiere, Concept lattice-based knowledge discovery in databases: Introduction, *J. Exper. Theor. Artif. Intell.* **14**(2/3) (2002) 75–79.
21. E. Mephu Nguifo and P. Njiwoua, Using lattice-based framework as a tool for feature extraction, *Proc. European Conf. Machine Learning (ECML-98)* (Chemnitz, Allemagne, April 1998), LNCS 1398 (Springer Verlag), pp. 304–309.
22. E. Mephu Nguifo and P. Njiwoua, Iglue: A lattice-based constructive induction system, *Int. J. Intell. Data Anal. (IDA)* **4**(4) (2000) 1–49.
23. E. Norris, An algorithm for computing the maximal rectangles in a binary relation, *Revue Roumaine de Mathématiques Pures et Appliquées* **23**(2) (1978).
24. L. Nourine and O. Raynaud, A fast algorithm for building lattices, *Third Int. Conf. Orders, Algorithms and Applications* (Montpellier, France, August 1999).
25. G. Oosthuizen, *The Use of a Lattice in Knowledge Processing*, PhD thesis, University of Strathclyde, Glasgow, 1988.
26. J. R. Quinlan, *Bagging, Boosting and C4.5* (AAAI Press, Menlo Park, CA, 1996).

27. M. Sahami, Learning classification rules using lattices, eds. N. Lavrac and S. Wrobel, *Proc. European Conf. Machine Learning (ECML'95)* (Heraclion, Crete, Greece, April 1995), pp. 343–346.
 28. M. Samuelides and E. Zenou, Learning-based visual localization using formal concept lattices, *2004 IEEE Workshop on Machine Learning for Signal Processing* (Sao Luis (Brasil), September 29–October 1st 2004), pp. 43–52.
 29. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier and L. Lakhal, Computing iceberg concept lattices with TITANIC, *Data Know. Engin.* **42**(2) (2002) 189–222.
 30. S. Tabbone, L. Wendling and J. P. Salmon, A new shape descriptor defined on the Radon transform, *Comput. Vis. Imag. Underst.* **102**(1) (2006) 42–51.
 31. M. Teague, Image analysis via the general theory of moments, *J. Opt. Soc. America (JOSA)* **70** (2003) 920–930.
 32. P. Valtchev, R. Missaoui and P. Lebrun, A partition-based approach towards constructing Galois (concept) lattices, *Discr. Math.* **3**(256) (2002) 801–829.
 33. R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, *Ordered sets*, ed. I. Rival (Dordrecht-Boston, Reidel, 1982), pp. 445–470.
 34. Z. Xie, W. Hsu, Z. Liu and M. Lee, Concept lattice based composite classifiers for high predictability, *J. Exper. Theoret. Artif. Intell.* **14**(2/3) (Taylor and Francis Ltd, 2002), pp. 143–156.
-



Muriel Visani received in 2005 her PhD in computer science from the Institut National des Sciences Appliquées of Lyon, France. During this period, she worked on face recognition from images for Orange Labs Company. In 2006, she joined as an Associate Professor in the

Computer Science Laboratory (L3i) of the University of La Rochelle (France).

Her research activities focus on image analysis and recognition, especially the recognition of complex objects in images, enriching image description for better indexing and retrieval. For more details please refer to <http://perso.univ-lr.fr/mvisani>.



Karelle Bertet received her PhD degree in computer science from the University of Paris 7, France, in 1998. During her PhD thesis (1995–1998), she worked on some algorithmical and structural aspects of lattices. Since 1999, she is Assistant Professor at the

University of La Rochelle, working on the links between fundamental aspects of lattice theory and their applications, such as symbolic classification methods and knowledge based representation for document images. Her present research interests focus on and knowledge based representation.



Jean-Marc Ogier received his PhD degree in computer science from the University of Rouen, France, in 1994. During this period (1991–1994), he worked on graphic recognition for Matra Ms & I Company. From 1994 to 2000, he was an Associate Professor at the University

of Rennes 1 during the first period (1994–1998) and at the University of Rouen from 1998 to 2001. Currently he is full professor at the University of La Rochelle.

Dr Ogier's works in the L3i laboratory in which he manages a research group (12 permanent staff, 20 PhD) dealing with document analysis. He manages several French and European projects dealing with historical document analysis, either with public institutions, or with private companies. He is a Deputy Director of the GDR I3 of the French National Research Center (CNRS). He is also Chair of the Technical Committee 10 (Graphic Recognition) of the International Association for Pattern Recognition (IAPR). Finally, he is also Vice Rector of the University of La Rochelle.