Extensions of Bordat's algorithm for attributes

K. Bertet, S. Guillas, and J.-M. Ogier

Laboratory L3I - University of La Rochelle - France

Abstract. In our works, we use a concept lattice as classifier of noised symbol images. On the contrary of others methods of classification based on Formal Concept Analysis [12], our approach is adapted to the special case of noisy since it is based on a navigation into the lattice structure to classify a noised symbol image : the navigation is performed from the minimal concept, until a final concept is reached, according to the cover-relation between concepts. Class of the input noised symbol is then the class associated to the reached final concept.

We use Bordat's algorithm to generate the concept lattice since it generates the cover relation of the lattice. In this paper, we present three extensions of Bordat's algorithm : the first extension generates the reduction of the concept lattice to its attributes, i.e. a closure system on attributes ; the second extension generates concepts only when required during the navigation, thus a reduction of the total number of generated concepts ; the third extension generates the concept lattice together with the canonical direct basis, i.e. a basis of implication rules between attributes to describe them.

Key words: generation algoritm, concept lattice, classification, noised images

1 Introduction

In our current works in [16], we use a concept lattice as classifier of noised symbol images, where objects are symbols and attributes are features issued from images of symbols. On the contrary of others methods of classification based on Formal Concept Analysis [12], our approach is based on a navigation into the lattice structure to classify a symbol image : the navigation consists in a breadth-first search in the concept lattice starting from the bottom, until a final concept is reached, using a choice criteria to choose according to the cover-relation between concepts. Class of the input noised symbol is then the class associated to the reached final concept. This approach is similar to the use of a decision tree, adapted to the special case of noisy objects.

Bordat's algorithm is an appropriate algorithm to generate the concept lattice since it generates the cover relation between concepts. A concept is a pair (A, B) where A is a subset of attributes and B is a subset of objects. Since we only need attributes during the navigation in the concept lattice, Bordat's algorithm is first extended to compute the reduction of the concept lattice to the attributes. Such a reduction forms a closure system on attributes.

Only a small part of the concept lattice is explored by navigation, the others concepts are not required. Thus, the navigation implies the possibility to on-line generate a concept only when it is reached by navigation, aims of our second extension of Bordat's algorithm. As the main drawback of the concept lattice is its exponential complexity in the worst case, we understand the interest to build only the concepts necessary to the recognition.

Implication rules between attributes represents an efficient tools to describe relationship between attributes in different areas research (databases area [18] ; data-mining [26], [25] ...). Among equivalent implicational systems, there exists two basis : the canonical basis [14] that has the canonical and minimality properties ; the canonical direct basis [17] that has the directness, canonical and minimality properties. As a third extension of Bordat's algorithm, we describe the generation the canonical direct basis while generated of the cover relation of the concept lattice.

Section 2 gives definitions of a concept lattice, a closure system and an implicational system. The three extensions of Bordat's algorithm for classification of noisy symbols are descried in Section 3.

2 Definitions

In this paper, all sets are finite sets and a subset $X = \{x_1, x_2, \ldots, x_n\}$ is written as the word $X = x_1 x_2 \ldots x_n$. Moreover, we abuse notation and use $X + x$ (respectively, $X - x$) for $X \cup \{x\}$ (respectively, $X \backslash \{x\}$), with $X \subseteq S$ and $x \in S$.

Binary relation and lattice. An order relation $\leq$ on a set S is a binary relation on S which is reflexive ($\forall x \in S, x \leq x$), antisymmetric ($\forall x \neq y \in S$, $x \leq y$ imply $y \not\leq x$) and transitive ($\forall x, y, z \in S$, $x \leq y$ and $y \leq z$ imply $x \leq z$). A *partially ordered set* $P = (S, \leq)$, also called a *poset*, is a set S equipped with an order relation $\leq$. The cover relation $\prec$ on S is the transitive reduction of the order relation $\leq$. A poset $L = (S, \leq)$ is a *lattice* if any pair $\{x, y\}$ of elements of L has a *join* (i.e. a least upper bound) denoted by $x \vee y$ and a *meet* (i.e. a greatest lower bound) denoted by $x \wedge y$. Therefore, a lattice contains a minimum (resp. maximum) element according to the relation $\leq$ called the *bottom* (resp. *top*) of the lattice, and denoted $\bot_S$ or simply $\bot$ (resp. $\top_S$ or simply $\top$.)

An equivalence relation $\sim$ on a set S is a binary relation on S which is reflexive ($\forall x \in S, x \sim x$), symmetric ($\forall x \neq y \in S$, $x \sim y$ imply $y \sim x$) and transitive ($\forall x, y, z \in S$, $x \sim y$ and $y \sim z$ imply $x \sim z$). The equivalence class of $a \in S$ is the subset of all elements in S which are equivalent to a : $\lfloor a \rfloor = \{x \in S : x \sim a\}$. The set of all equivalence classes in S given an equivalence relation $\sim$ is usually denoted as $S/_\sim$, and called the quotient set of S by $\sim$. The quotient set $S/_\sim$ forms a partition of S.

Concept lattice. A *formal context* $K = (G, M, I)$ consists of two sets G and M, and a relation I between G and M. The elements of G are called the *objects*, and the elements of M are called the *attributes* of the context. We define the application f that associates to every element $o \in G$ the set $f(o) = \{a \in M \mid (o, a) \in I\}$, and the application g that associates to every element $a \in M$

the set $g(a) = \{o \in G \mid (o,a) \in I\}$. The extension of f to subsets $A \subseteq G$ provides :

$$f(A) = \cap_{o \in A} f(o) = \{a \in M \mid \forall\ o \in A,\ (o,a) \in I\ \} \quad (1)$$

Dually, the extension of g to subsets $B \subseteq M$ provides :

$$g(B) = \cap_{a \in G} g(a) = \{o \in G \mid \forall\ a \in B,\ (o,a) \in I\} \quad (2)$$

The two applications f and g forms a *Galois connection* between G and M.

A *formal concept* of a formal context K is a pair (A,B) with $A \subseteq G$, $B \subseteq M$, $f(A) = B$ and $g(B) = A$. Let $\mathbb{C}$ be the set of all the concepts of K, and $\leq_C$ be the following order relation on $\mathbb{C}$, with (A_1,B_1) and (A_2,B_2) two concepts :

$$(A_1,B_1) \leq_C (A_2,B_2) \text{ iff } A_1 \subseteq A_2 \text{ (or equivalently } B_2 \subseteq B_1) \quad (3)$$

The poset $(\mathbb{C}, \leq_C)$ is a lattice called the *concept lattice*. This lattice is also denoted *Galois lattice* by reference to the Galois connection (f,g) of the formal context C. In particular $\bot = (f(G), G)$ and $\top = (M, g(M))$. Moreover for all (A_1,B_1), $(A_2,B_2) \in \mathbb{C}$, $(A_1,B_1) \vee (A_2,B_2) = (f(B_1 \cap B_2), B_1 \cap B_2)$ and $(A_1,B_1) \wedge (A_2,B_2) = (A_1 \cap A_2, g(A_1 \cap A_2))$. In *formal concept analysis* (FCA) concept lattices are used to analyze data when organized by a binary relation between object and attributes. See the complete book of Ganter and Wille [12] for a complete description of formal concept analysis.

Closure system. A *set system* on a set S is a family of subsets of S. A *closure system* $\mathbb{F}$ on a set S, also called a *Moore family*, is a set system stable by intersection and which contains S : $S \in \mathbb{F}$ and $F_1, F_2 \in \mathbb{F}$ implies $F_1 \cap F_2 \in \mathbb{F}$. The subsets belonging to a closure system $\mathbb{F}$ are called the *closed sets* of $\mathbb{F}$.

The poset $(\mathbb{F}, \subseteq)$ is a lattice with, for each $F_1, F_2 \in \mathbb{F}$, $F_1 \wedge F_2 = F_1 \cap F_2$ and $F_1 \vee F_2 = \bigcap\{F \in \mathbb{F} \mid F_1 \cup F_2 \subseteq F\}$. Moreover, any lattice L is isomorphic to the lattice of closed sets of a closure system (see [2] for more details).

A *closure operator* on a set S is a map φ on $\mathcal{P}(S)$ satisfying the three following properties : φ is *isotone* (i.e. $\forall X, X' \subseteq S$, $X \subseteq X' \Rightarrow \varphi(X) \subseteq \varphi(X')$), *extensive* (i.e. $\forall X \subseteq S$, $X \subseteq \varphi(X)$) and *idempotent* (i.e. $\forall X \subseteq S$, $\varphi^2(X) = \varphi(X)$). Closure operators are in one-to-one mapping with closure systems. On the first hand, the set of all closed elements of φ forms a closure system $\mathbb{F}_\varphi$:

$$\mathbb{F}_\varphi = \{F \subseteq S \mid F = \varphi(F)\} \quad (4)$$

Dually, given a closure system $\mathbb{F}$ on a set S, one defines the closure $\varphi_{\mathbb{F}}(X)$ of a subset X of S as the least element $F \in \mathbb{F}$ that contains X :

$$\varphi_{\mathbb{F}}(X) = \bigcap\{F \in \mathbb{F} \mid X \subseteq F\} \quad (5)$$

In particular $\varphi_{\mathbb{F}}(\emptyset) = \bot_{\mathbb{F}}$. Moreover for all $F_1, F_2 \in \mathbb{F}$, $F_1 \vee F_2 = \varphi_{\mathbb{F}}(F_1 \cup F_2)$ and $F_1 \wedge F_2 = \varphi_{\mathbb{F}}(F_1 \cap F_2) = F_1 \cap F_2$.

A concept lattice $(\mathbb{C}, \leq_C)$ is composed of two particular closure systems $\mathbb{C}_G$ and $\mathbb{C}_M$ respectively defined on the set G of objects, and on the set M of attributes : $\mathbb{C}_G$ is the restriction of $\mathbb{C}$ to the objects where each concept (A, B) is replaced by the subset A of objects ; the associated closure operator is $f \circ g$. Dually, $\mathbb{C}_M$ is the restriction of $\mathbb{C}$ to the attributes and the associated closure operator is $g \circ f$. Moreover, the three lattices $(\mathbb{C}_G, \subseteq)$, $(\mathbb{C}_M, \supseteq)$ and $(\mathbb{C}, \leq)$ are isomorphic. See the survey of Caspard and Monjardet [5] for more details about closure systems.

Implicational system. An *Unary Implicational System* (UIS for short) Σ on S is a binary relation between $\mathcal{P}(S)$ and S : $\Sigma \subseteq \mathcal{P}(S) \times S$. An ordered pair $(A, b) \in \Sigma$ is called a *Σ-implication* whose *premise* is A and *conclusion* is b. It is written $A \rightarrow b$, meaning "A implies b". A subset $X \subseteq S$ *respects* a Σ-implication $A \rightarrow b$ when $A \subseteq X$ implies $b \in X$ (i.e. "if X contains A then X contains b"). $X \subseteq S$ is *Σ-closed* when X respects all Σ-implications, i.e $A \subseteq X$ implies $b \in X$ for every Σ-implication $A \rightarrow b$. The set of all Σ-closed sets forms a closure system $\mathbb{F}_\Sigma$ on S :

$$\mathbb{F}_\Sigma = \{X \subseteq S \,|\, X \text{ is } \Sigma\text{-closed}\} \tag{6}$$

Then we can associate to Σ a closure operator $\varphi_\Sigma = \varphi_{\mathbb{F}_\Sigma}$ which defines the closure of a subset $X \subseteq S$

$$\varphi_\Sigma(X) = \pi_\Sigma(X) \cup \pi^2_\Sigma(X) \cup \pi^3_\Sigma(X) \cup \ldots \text{ where} \tag{7}$$

$$\pi_\Sigma(X) = X \cup \bigcup \{b \,|\, A \subseteq X \text{ and } A \rightarrow b\} \tag{8}$$

An UIS Σ is a *generating system* of the closure system $\mathbb{F}_\Sigma$ (using Eq. (4)), and thus for the induced closure operator φ, and the induced lattice $(\mathbb{F}_\Sigma, \subseteq)$. When some UISs Σ and Σ' on S are generating systems for the same closure system, they are called *equivalent* (i.e. $\mathbb{F}_\Sigma = \mathbb{F}_{\Sigma'}$).

An UIS Σ is called *direct* or *iteration-free* if for every $X \subseteq S$, $\varphi(X) = \pi_\Sigma(X)$ (see Eq. (8)). An UIS Σ is *minimal* or *non-redundant* if $\Sigma \setminus \{X \rightarrow y\}$ is not equivalent to Σ, for all $X \rightarrow y$ in Σ. It is *minimum* if it is of least cardinality, i.e. if $|\Sigma| \leq |\Sigma'|$ for all UIS Σ' equivalent to Σ. A minimum UIS is trivially non-redundant, but the converse is not true in general. A minimal UIS is usually called a *basis* for the induced closure system (and thus for the induced lattice), and a *minimum basis* is then a basis of least cardinality.

The *canonical direct basis* described in [17] and denoted Σ_{cd}, has three main properties : directness, canonical and minimality properties.

In the literature, an implicational system (IS for short) Σ can also be defined as a binary relation on $\mathcal{P}(S)$. A Σ-implication is then an ordered pair $(A, B) \in \Sigma$, written $A \rightarrow B$, with $A, B \in \mathcal{P}(S)$. Generating systems (also called *covers*) and bases can be also defined for IS. In this case, there exists an unique minimum basis, called the *canonical basis*, also denoted the *Guigues and Duquenne basis* ([15]), enabling to get all the others minimum basis.

Other definitions and bibliographical remarks can be found in the survey of Caspard and Monjardet in [5].

Bordat's algorithm. One of the first algorithm generating the concept lattice is due to Chein [6] : concepts are generating from the inital context using submatrix computation algorithm. These first algorithms has then been improved using efficient methods for testing wether a concept has been already generated. The initial algorithms are Norris's algorithm [23], Next Closure of Ganter [11], Bordat's algorithm [3]. Norris's algorithm is an incremental algorithm, as algorithms in [13] and [27]. Next Closure algorithm defines the *lectic order* (extended the inclusion order) between concept for testing wether a concept has been already generated. Bordat's agorithm computes the Hasse diagram of the concept lattice, as algorithms in [4] and [27]. Nourine and Raynaud [24] use a sophisticated tree data structure to generated concepts with the best theoretical complexity.

Name : `Concept-Lattice`
Input: A context $K = (G, M, I)$
Output: The cover relation $(\mathbb{C}, \prec)$ of the concept lattice of K
begin
    $\mathbb{C} = \{(f(G), G))\}$;
    foreach $(A, B) \in \mathbb{C}$ *not marked* **do**
        F=`cover` $(K, (A, B))$;
        foreach $B' \in F$ **do**
            $A' = g(B')$;
            if $(A', B') \notin \mathbb{C}$ **then** add (A', B') in $\mathbb{C}$;
            add the cover relation $(A, B) \prec (A', B')$
        end
        mark (A, B)
    end
    return $(\mathbb{C}, \prec)$
end

For our classification problem, Bordat's algorithm [3], or any algorithm generating the Hasse diagram, is the more appropriate since it generates the cover relation between concepts. Bordat's algorithm is issued from Theorem 1 that is redefined in Corollary 1.

Theorem 1 (Bordat [3]). *Let* (A, B) *and* (A', B') *two concepts of a context* (G, M, I)*. Then* $(A, B) \prec (A', B')$ *if and only if* B' *is inclusion-maximal in the following set system* F_A *defined on* G *:*

$$F_A = \{f(a) \cap B \ : \ a \in M - A\} \tag{9}$$

Corollary 1 (Bordat [3]). *Let* (A, B) *be a concept. There is a one-to-one mapping between* $Cover((A, B))$ *and the inclusion-maximal subsets of* F_A *where :*

$$Cover((A, B)) = \{(A', B') \ : \ (A, B) \prec (A', B')\} \tag{10}$$

Bordat's algorithm in Algorithm `Concept-Lattice` computes all the concepts of $\mathbb{C}$ by computing $cover(A, B)$ for each concept (A, B), starting from the bottom concept $\bot = (f(G), G)$, until all concepts are generated. It is in $O(|\mathbb{C}||M|^{\alpha})$, with $2,5 \leq \alpha \leq 3$, since each concept is issued from the computation of $cover((A, B))$ that is in $O(|M|^{\alpha})$.

Algorithm `cover-objects` describe the two steps of the computation of $cover((A, B))$: the set system F_A has first to be generated in a linear time since F_A can be computed in an incremental way ; then inclusion-maximal subsets of F_A can easily be computed in $O(|M|^3)$, but the inclusion-maximal subsets problem is known to be resolved in $O(M|^{2,5})$ using sophisticated data structures ([10,22]).

Name : `Cover-objects`
Input: A context K ; A concept (A, B) of K
Output: The inclusion-maximal subsets of F_A
begin

1. Compute F_A : $F_A = \{f(a) \cap B \ : \ a \in M - A\}$;
2. Compute F : the maximal-inclusion subsets of F_A;

return F
end

3 Extensions of Bordat's algorithm for recognition of noised symbols

Classification with a concept lattice. In our current works in [16], we use a concept lattice as classifier of noised symbol images. Each symbol X is giving by a vector of features $(x_i)_{i \leq n}$, denoted its *signature*, and extracted from the image of a symbol X using image analysis treatment. A class information $c(X)$ is also associated to each symbol.

The classification problem consists then in computing the class information of not classed and noised symbols. This problem is usually decomposed into two step : a *learning step* aiming at generating a classifier from a set of symbols as input ; a *classification step* aiming at classify noised symbols using the classifier. The noised symbol recognition problem takes as learning input a set of symbols given by their signature and their class ; and as classification symbols a set of noised symbols given by their signatures.

In the learning step, features of the signature are discretized into intervals in order to separate between symbols of different classes. The relation between the learning symbols (objects) and the features's intervals (attributes) forms a context $K = (G, M, I)$ and the classifier is then the cover relation of the concept lattice of K.

This concept lattice classifier is used to classify a noised symbol $X = (x_i)_{i \leq n}$ in a second step by navigation into the concept lattice. The navigation consists in a breath-first search in the concept lattice starting from the bottom concept, until a final concept is reached, using a choice criteria to choose according to the cover-relation between concepts. Class of the input noised symbol is then the class associated to the reached final concept, thus a concept is a final concept when it is composed of objects of the same class. A final concept is covered by the top concept. This classifier is based on three extensions of Bordat's algorithm :

1. We need attributes and not objects during the navigation in the concept lattice. Thus a first extension of Bordat's algorithm to compute the closure system on attributes instead if the concept lattice.
2. We explore by navigation only a small part of the concept lattice depending on the input symbols to classify. So, all concepts are not required. The second extension of Bordat's algorithm consists to on-line generate a concept only when it is reached by navigation.
3. Implication between attributes represents an efficient tools to describe attributes. The third extension of Bordat's algorithm is a generation of the canonical direct basis while generating concepts.

Generation of the closure system on attributes. In the context of recognition of noised symbols, the cover relation $(\mathbb{C}_M, \prec)$ of the closure system on attributes is sufficient for the navigation.

Let $A \in \mathbb{C}_M$ be a closed set. The cover of A in the lattice $(\mathbb{C}_M, \prec)$ is redefined as :

$$Cover(A) = \{A' : A' \prec A\} \tag{11}$$

Although the set system F_A is composed of subsets of objects of G, it is defined according to attributes of $M \setminus A$ (see 9). Let us consider the two following cases for a and a' two attributes of $M \setminus A$:

1. If $f(a) \cap B = f(a') \cap B$ then a and a' give raise to one subset $f(a) \cap B$ in F_A. Thus a and a' are equivalent in this case.
2. If $f(a) \cap B \neq f(a') \cap B$, then the two subsets $f(a) \cap B$ and $f(a') \cap B$ belong to F_A. The maximal-inclusion subsets of F_A are deduced from the inclusion relation between $f(a) \cap B$ and $f(a') \cap B$, and can be extended to a relation between a and a'.

To formalize the first case, we introduce an equivalence relation $\sim$ on attributes of $M \setminus A$. The second case can then be reformulated using an ordered relation $\lhd_A$ on the set of equivalence classes of $\sim$. This set is called the quotient set by $\sim$ and denoted $(M \setminus A)/_\sim$:

1. $\sim$ is an equivalence relation $\sim$ on $M \setminus A$ defined by

$$\forall a, a' \in M \setminus A,\, a \sim a' \iff f(a) \cap B = f(a') \cap B \tag{12}$$

2. $\lhd_A$ is the order relation defined on the quotient set $(M \setminus A)/\sim$ by :

$$\forall a, a' \in M \setminus A,\, \lfloor a \rfloor \lhd_A \lfloor a' \rfloor \iff f(a) \cap B \subseteq f(a') \cap B \tag{13}$$

Therefore, F_A can be be extended to an order relation on equivalence classes on attributes of $M \setminus A$. The following corollary extends Theorem 1 to an use of the only attributes, and gives raise to Algorithm `Closure-System` that generates the cover each closed set A using Algorithm `cover-attributes`.

Corollary 2. *Let $A \in \mathbb{C}_M$. There is a one-to-one mapping between $Cover(A)$ and the maximal elements of the poset $((M \setminus A)/_\sim, \triangleleft_A)$.*

Name : `Closure-System`
Input: A context $K = (G, M, I)$
Output: The cover relation $(\mathbb{C}_M, \prec)$ of the lattice $(\mathbb{C}_M, \subseteq)$

```
begin
    C_M = {f(G)};
    foreach A ∈ C_M not marked do
        F=cover-attributes (K, A);
        foreach X ∈ F do
            A' = A + X;
            if A' ∉ C_M then add A' in C_M;
            add a cover relation A ≺ A'
        end
        mark A
    end
    return (C_M, ≺)
end
```

Algorithm `cover-attributes` is very similar to Algorithm `cover-attributes`, and can also be resolved in $O(|\mathbb{C}||M|^\alpha)$ using sophisticated data structure. However, the maximal elements of the inclusion relation $\triangleleft_A$ can be updated in a incremental way in $O(|M|^3)$ since equivalence classes for a closed set A are included in those of closed set successors of A. Therefore, Algorithm `cover-attributes` is in $O(|M|^3)$, and Algorithm `Closure-System` is in $O(|\mathbb{C}||M|^3)$, as Bordat's algorithm.

A similar poset introduced by Morvan and Nourine and denoted G'_A in [20] or $Pred^*_A$ in [21] is defined according to $B \setminus f(a)$ instead of $f(A) \cap B$. They state the following equivalent theorem :

Theorem 2 (Morvan, Nourine [20]). *Let $A \in \mathbb{C}_M$. There is a one-to-one mapping between $Cover(A)$ and $min(G'_A)$.*

They derive from this theorem a generation algorithm of "minimal interval extensions" based on a one-to-one mapping between theses extensions and the maximal chains of a closure system ordered by inclusion (i.e. a lattice). In another paper [21], this algorithm has been simplified to the distributive case (case where the concept lattice is distributive) in $O(|\mathbb{C}|)$: in this case, the $\triangleleft_A$'s similar poset can be computed in a post-treatment, and thus has not to be updated for each closed set. This generation is called the *strong simplicial elimination scheme* and corresponds to the decomposition process of a distributive lattice in intervals described by Markowski in [19] ; as to the duplication process of intervals introduced by Day in [7] and generalized to other duplications in [8,9,1].

Name : `cover-attributes`
Input: A context K ; closed set $A \in \mathbb{C}_M$ of attributes
Output: The maximal elements of $((M \setminus A)/_\sim, \lhd_A)$.
begin

1. Compute $(M \setminus A)/_\sim$: the equivalences classes of $\sim$ on $M \setminus A$.
2. Compute $\lhd_A$: the inclusion relation on $(M \setminus A)/_\sim$;
3. Compute F : the maximal elements of $\lhd_A$

return F
end

Extension to an on-line generation. Algorithm `On-Line-Closure-System` is an extension of Bordat's algorithm to on-line generate closed sets according to a choice criteria to select a closed set between *cover*(A). Since Bordat's algorithm generates concepts with a breath-first strategy, it has been adapted to a death-first strategy in a recursive way, and has initially to be called with $\varphi(\emptyset)$ as first closed set.

Let us notice that a death-first generation of all closed sets would consists in replacing `Select` X `in` F by the loop `ForEach` X `in` F to consider all subsets of F in the same way, thus uselessness of the else statement.

Algorithm `On-Line-Closure-System` is in $(O(|\mathbb{C}^{on-line}||M|^3)$ where $\mathbb{C}^{on-line}$ is the set of closed set on-line generated. This set depends of the symbols to classify : $|M| \leq |\mathbb{C}^{on-line}| \leq |\mathbb{C}|$

Table 3 illustrates the interest of an on-line generation for recognition of noised symbols.

	Learning	Recognition	Number of concepts
Total generation	430,2 sec	2 sec	3185
On-line generation	0,5 sec	9,8 sec	282

Tab. 1. Recognition of 10 noised symbols ; Learning with 25 not noised symbols

Extension to generation of the canonical direct basis. It is possible to extend relation $\lhd_A$ to be defined on $M \setminus A$ instead on the quotient set $(M \setminus A)_\sim$:

$$\forall a, a' \in M \setminus A,\ a \lhd_A a' \iff f(a) \cap B \subset f(a') \cap B \qquad (14)$$

Name : `On-Line-Closure-System`
Input: A context $K = (G, M, I)$; a suborder $(\mathbb{C}'_M, \prec)$ of the cover relation $(\mathbb{C}, \prec)$; a closed set $A \in \mathbb{C}'_M$
begin
  if A *not marked* **then**
    F=`Cover-Attributes` (K, A);
    Select X in F according to the choice criteria;
    $A' = A + X$;
    add the cover relation $A \prec A'$;
    if $A' \notin \mathbb{C}'_M$ **then** add A' in $\mathbb{C}'_M$;
    mark A;
  end
  else
    Let $F = \{A' \setminus A \, : \, A' \in \mathbb{C}'_M$ and $A \prec A'\}$;
    Select X in F according to the choice criteria;
  end
  if $A' \neq M$ **then** `On-Line-Closure-System`$(K, \mathbb{C}'_M, \prec, A')$;
end

It's important to notice that the relation $\lhd_A$ defined on $M \setminus A$ isn't an ordered relation since $a \lhd_A a'$ and $a' \lhd_A a$ for two $\sim$-equivalent attributes a and a'.

It is stated in [**?**] that $Pred^*_A$ (i.e. relation $\lhd_A$) is equivalent to the *dependance relation* δ defined for a lattice, and introduced in [**?**] (see also [**?**]). The *dependence relation* δ_X is defined on S, with $x, y \in S$ and $X \subset S$, by :

$$x \delta_X y \text{ if and only if } x \notin \varphi(X),\ y \notin \varphi(X) \text{ and } x \in \varphi(X + y) \qquad (15)$$

In [**?**], the authors state the equality between the canonical basis and five implicational systems issued from different works and satisfying various properties. One of the five implicational system is the dependence relation's basis on S is issued from the *dependence relation* :

$$\Sigma_{cd} = \{X + y \rightarrow x \, : \, x \delta_X y \text{ and } X \text{ is minimal for this property}\} \qquad (16)$$

Therefore, using relation $\lhd_A$, it is possible to compute the canonical direct basis Σ_{cd} of the closure set $\mathbb{C}'_M$ using Eq.16 as done by Algorithm `Canonical-Direct-Basis`. In particular $\lhd_\emptyset$ corresponds to the inclusion relation on M, and to unitary implicational rules in Σ_{cd} (i.e. rules with a singleton as premise).

The canonical direct basis Σ_{cd} can be computed in $(O(|\mathbb{C}_M||M|^3)$, thus a complexity of Algorithm `On-Line-Closure-System` in $(O(|\mathbb{C}_M||M|^3)$, as Bordat's algorithm.

4 Conclusion

We present in this paper three extensions of Bordat's algorithm. Although these extensions have been introduced for the noised symbol recognition problem, it can be implemented with any concept lattice.

```
Name : Canonical-Direct-Basis
Input: A context K = (G, M, I)
Output: The Cover relation of the lattice (ℂ_M, ⊆) ; The canonical direct basis Σ_cd
begin
    ℂ_M = {f(G)};
    Σ_cd = ∅;
    Let ◁_∅ be the inclusion relation on M;
    foreach (a, a') ∈ M² such that a ◁_∅ a' do
        add a' → a in Σ_cd
    end
    foreach A ∈ ℂ_M not marked do
        Let F = cover-attributes (K, A);
        Let ◁_A = the inclusion relation on M \ A;
        foreach X ∈ F do
            A' = A + X;
            foreach (a, a') ∈ M \ A' such that a ◁_A' a' and a ⋪_A a' do
                add A + a' → a in Σ_cd
            end
            if A' ∉ ℂ_M then add A' in ℂ_M;
            add a cover relation A ≺ A'
        end
        mark A
    end
    return (ℂ_M, ≺) and Σ_cd
end
```

Acknowledgments to Philippe Sachot and Antoine Mercier for the implementation of the algorithms presented in this paper.

Références

1. K. Bertet and N. Caspard. Doubling convex sets in lattices : characterisations and recognition algorithms. *Order*, 19(2) :181–207, 2002.
2. G. Birkhoff and O. Frink. Representations of lattices by sets. *Transactions of the American Mathematical Society*, 64 :299–316, 1948.
3. J. Bordat. Calul pratique du treillis de galois d'une correspondance. *Math. Sci. Hum.*, 96 :31–47, 1986.
4. C. Carpineto and G. Romano. Galois : An order-theoretic approach to conceptual clustering. In *Proceedings of ICML'93*, pages 33–40, Amherst, July 1993.
5. N. Caspard and B. Monjardet. The lattice of closure systems, closure operators and implicational systems on a finite set : a survey. *Discrete Applied Mathematics*, 127(2) :241–269, 2003.
6. M. Chein. Algorithme de recherche des sous-matrices premières d'une matrice. *Bulletin Mathématique de la Sociologie Scientifique de la R.S. de Roumanie*, 1969.
7. A. Day. A simple solution of the word problem for lattices. *Canad. Math. Bull.*, 13 :253–254, 1970.

8. A. Day. Characterizations of finite lattices that are bounded-homomorphic images or sublattices of free lattices. *Canadian J. Math.*, 31 :69–78, 1979.
9. A. Day, J. Nation, and S. Tschanz. Doubling convex sets in lattices and a generalized semidistributivity condition. *Order*, 6 :175–180, 1989.
10. M. Fisher and A. Meyer. Boolean matrix multiplication and transitive closure. In 12^{th} *Annual Sympsosium on Switching and Automata Theory*, pages 129–131, 1971.
11. B. Ganter. Two basic algorithms in concept analysis. *Technische Hochschule Darmstadt (Preprint 831)*, 1984.
12. B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 1999.
13. R. Godin, R. Missaoui, and H. Alaoui. Learning algorithms using a Galois lattice structure. *Third International Conference on Tools for Artificial Intelligence, San Jose, Calif.*, pages 22–29, 1991.
14. J. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 95 :5–18, 1986.
15. J. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathematiques & Sciences Humaines*, 95 :5–18, 1986.
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. In *Fourth International Conference on Concept Lattices and their Applications (CLA'2006)*, pages 257–263, Tunisia, 2006.
17. B. M. K. Bertet. The multiple facets of the canonical direct basis, 2005.
18. D. Maier. *The Theory of Relational Databases*. Computer Sciences Press, 1983.
19. G. Markowski. Primes, irreducibles ans extremal lattices. *Order*, 9 :265–290, 1992.
20. M. Morvan and L. Nourine. Generating minimal interval extensions. Research Report 92-015, LIRMM, Montpellier, juin 1992.
21. M. Morvan and L. Nourine. Simplicial elimination shemes, extremal lattices and maximal antichains lattice. *Order*, 13(2) :159–173, 1996.
22. I. Munro. Efficient determination of the transitive closure of a directed graph. *Information Processing Letter*, pages 56–58, 1971.
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. In *Third International Conference on Orders, Algorithms and Applications*, Montpellier, France, august 1999.
25. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In L. S. Verlag, editor, *ICDT'99*, volume 1540, pages 398–416, 1999.
26. R. Taouil and Y. Bastide. Computing proper implications. *Proceedings of ICCS-2001 Internationnal Workshop on Concept Lattices-Based Theory, Methods and tools for Knowledge Discovery in Databases*, pages 290–303, 2001.
27. P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards constructing galois (concept) lattices. *Discrete Mathematics*, 3(256) :801–829, 2002.