Reconnaissance de symboles bruités à l'aide d'un treillis de Galois

Stéphanie Guillas, Karell Bertet, Jean-Marc Ogier

Reconnaissance de symboles bruités à l'aide d'un treillis de Galois

Stéphanie Guillas – Karell Bertet – Jean-Marc Ogier

Laboratoire L3I
Université de La Rochelle, Pôle Sciences et Technologie, 17042 La Rochelle Cedex 1, FRANCE

{sguillas,kbertet,jmogier}@univ-lr.fr

Résumé : *Ce papier présente une méthode de classification supervisée basée sur l'utilisation d'un graphe particulier appelé* treillis de Galois *ou* treillis des concepts *comme classifieur. L'application expérimentale est réalisée sur les symboles bruités de la base GREC2003 [GRE 03]. Le processus de reconnaissance est décrit en détail afin de préciser les éléments paramétrables. Il est classiquement réalisé en 2 étapes : l'apprentissage au cours duquel le treillis de Galois est construit, et la classification qui utilise le graphe obtenu pour la reconnaissance des symboles bruités. L'expérimentation sur les symboles de GREC2003 nous amène à comparer le treillis de Galois à 2 autres classifieurs très connus dans le domaine : le classifieur* k-ppv *et le classifieur* bayésien*. Les taux de reconnaissance du treillis sont relativement proches de ceux obtenus par les autres classifieurs alors que sa reconnaissance est basée sur une faible quantité de caractéristiques sélectionnées au sein de la signature. De plus, le treillis apporte en lisibilité (pas d'effet boite noire), ce qui permet notamment de comprendre et suivre pas à pas le cheminement au sein du graphe lors de la reconnaissance et d'évaluer les paramètres. Outre cette comparaison expérimentale, nous intégrons dans cette nouvelle étude les bases de la logique floue lors du processus de classification.*

Mots-clés : Treillis de Galois, Classification supervisée, Reconnaissance de symboles bruités

1 Introduction

Le travail présenté dans cet article se place dans le contexte de la rétroconversion automatique de documents techniques et propose d'exploiter les treillis de Galois pour la reconnaissance d'objets graphiques, comme les caractères ou les symboles, sous la contrainte classique de multi-orientation et multi-échelle.

Parmi les diverses techniques utilisées en classification, celles basées sur un treillis de Galois ont fait l'objet d'études comparatives dans de récents travaux [MEP 05]. De par les expérimentations pratiques qui y sont comparées, il y apparait clairement que le treillis de Galois offre un cadre intéressant en classification, malgré une complexité théorique exponentielle dans le pire des cas, mais polynomiale en pratique.

Dans une première étude [GUI 06], nous avons réalisé le réglage des paramètres de la méthode et une comparaison avec celle très proche de l'arbre de décision. Nous avons montré que la structure de l'arbre de décision est incluse dans celle du treillis de Galois. Les résultats expérimentaux ont montré que la taille plus importante du treillis (par rapport à celle de l'arbre) lui offre une meilleure robustesse au bruit. En effet, la multitude de chemins possibles pour atteindre une classe dans le treillis sont autant de scénarii de classification qui permettent d'aboutir à la reconnaissance des classes.

Le treillis de Galois est un graphe particulier dont la structure est proche de celle de l'arbre de décision. Dans l'arbre de décision, un seul chemin de la racine vers une feuille permet d'atteindre une classe, alors que dans le treillis tous les chemins possibles pour atteindre une classe sont définis et représentés. De même, plusieurs caractéristiques de la signature penvent être examinées en même temps pour progresser d'un nœud à l'autre contrairement à l'arbre de décision où une seule caractéristique est traitée à la fois. Ils intègrent tous les deux la possibilité de traiter des données numériques et symboliques.

L'objectif de cet article est tout d'abord d'enrichir le formalisme concernant la mesure de distance utilisée au cours de la classification en intégrant les bases de la logique floue. L'autre objectif de ce papier est de réaliser une comparaison expérimentale du treillis et de deux classifieurs reconnus dans le domaine : les classifieurs k-ppv et bayésien basée sur la reconnaissance des symboles bruités de la base GREC2003. A travers cette étude expérimentale, nous souhaitons améliorer les performances de reconnaissance du treillis.

Dans la partie 2, nous décrivons l'apprentissage des données, étape qui comprend la discrétisation et la construction du treillis de Galois. La classification, et notamment la navigation dans le treillis de Galois sont ensuite décrites dans la partie 3. En partie 4, l'expérimentation propose une comparaison avec les classifieurs k-ppv et bayésien. Enfin, la conclusion et les perspectives sont proposées en partie 5.

2 Apprentissage

La phase d'apprentissage consiste à organiser l'information extraite d'un ensemble d'objets sous forme d'un treillis. Dans notre cas, les objets sont décrits par des vecteurs numériques (ou signatures) normalisés extraits à partir d'images de symboles. L'apprentissage se décompose en 2 étapes :

- une étape de *discrétisation* des données numériques : où les données sont réparties dans des intervalles disjoints. Cette étape est essentielle à la construction du treillis de Galois et se paramètre par un *critère de coupe* d'un intervalle.
- une étape de *construction du treillis* à partir des données discrétisées ne nécessitant aucun paramétrage.

2.1 Discrétisation

La discrétisation consiste à organiser l'ensemble de ces données numériques en intervalles discrets afin d'obtenir une caractérisation spécifique de chaque classe d'objets. Elle est réalisée à partir des signatures $p = (p_i)_{i \leq n}, p \in O$ que l'on peut organiser sous forme d'une table de données à double entrée (tab. 1). Au début, on construit pour chaque caractéristique $i \leq n$ de la signature un intervalle $x = I_i$ regroupant l'ensemble V_x des valeurs p_i des objets $p \in O$. Notons qu'après cette étape d'initialisation, chaque objet $p \in O$ est en relation avec l'unique intervalle $x = I_i$ pour chaque caractéristique i. Il s'agit ensuite de sélectionner un *intervalle* x à couper, et un *point de coupe* $v_j \in V_x$ dans cet intervalle parmi les n valeurs $V_x = v_1, \ldots, v_n$ triées par ordre croissant, puis de couper l'intervalle x en deux intervalles $x' = \{v_1, \ldots v_j\}$ et $x'' = \{v_{j+1}, \ldots v_n\}$ avec $V_{x'} = v_1 \leq \ldots \leq v_j$ et $V_{x''} = v_{j+1} \leq \ldots \leq v_n$. Chaque objet sera alors en relation d'appartenance R avec l'un des deux intervalles créés, ce qui permet de différencier les deux sous-ensembles d'objets formés. On réitère le procédé de découpage des intervalles tant que l'on n'est pas capable de distinguer chacune des classes. Dans notre cas, le critère d'arrêt de la discrétisation est donc la séparation des classes. La sélection de l'intervalle à couper est dépendante d'un *critère de coupe* à définir.

Lorsque chaque classe est caractérisable par un ensemble d'intervalles qui lui est propre, on obtient une table discrétisée (tab. 2) contenant l'ensemble des objets $p \in O$ et des intervalles $I = I_1 \times I_2 \times \ldots \times I_n$ avec I_i l'ensemble des intervalles obtenus pour chaque caractéristique $i = 1 \ldots n$. Pour toute caractéristique k n'ayant jamais été sélectionnée, I_k contient un seul intervalle ($|I_k| = 1$) en relation avec tous les objets, elle n'est donc pas discriminante, et peut être retirée de la table discrétisée. Par le biais de la discrétisation, le système permet donc de sélectionner les caractéristiques d'une signature les plus pertinentes et de supprimer les autres et rejoint ainsi la problématique de la sélection de variables. La table discrétisée obtenue peut se représenter par une relation d'appartenance R entre objets O et intervalles I, et permet de déduire les intervalles associés à un objet $p = (p_1, \ldots, p_n) \in O$, où p_i est la valeur pour la caractéristique $i = 1 \ldots n$.

Exemple 1 *Prenons l'exemple de la table 1 qui présente les données normalisées de 10 objets répartis suivant 4 classes. La signature caractérisant chaque objet comprend 3 caractéristiques (a, b et c). Après discrétisation par un critère de coupe défini à partir d'un calcul d'entropie, on obtient la table 2. Chacune des caractéristiques a été sélectionnée et découpée au moins une fois, elles sont donc toutes conservées.*

2.1.1 Critère de coupe

De nombreux critères permettent de sélectionner l'intervalle à diviser et son point de coupe. Le choix de ce critère est déterminant pour l'apprentissage. Nous recherchons un intervalle $x \in I$ (dont les valeurs $V_x = (v_1 \ldots v_n)$ sont triées par ordre croissant) qui *maximise* un critère, pour une valeur v_j donnée. L'intervalle sera coupé entre v_j et v_{j+1}.

Classe	*Ident.*	*Signature*		
		a [0-20]	*b* [0-20]	*c* [0-20]
1	1	1	4	15
	2	0	0	18
2	3	1	12	13
	4	0	16	15
	5	3	12	11
3	6	8	16	15
	7	6	20	20
	8	15	12	15
4	9	18	4	0
	10	20	12	2

FIG. 1 – Signatures des 10 objets avant discrétisation

Classe	*Ident.*	*Intervalles*					
		a1 [0-3]	*a2* [6-20]	*b1* [0-4]	*b2* [12-20]	*c1* [0-2]	*c2* [11-20]
1	1	X		X			X
	2	X		X			X
2	3	X			X		X
	4	X			X		X
	5	X			X		X
3	6		X		X		X
	7		X		X		X
	8		X		X		X
4	9		X	X		X	
	10		X		X	X	

FIG. 2 – Signatures des 10 objets après discrétisation

Une précédente étude comparative a montré que le coefficient de Hotelling permettait d'obtenir de meilleurs résultats que l'entropie et la distance maximale (pour plus d'information concernant les critères de coupe utilisés, voir [GUI 06]). Le *coefficient de Hotelling* choisit l'intervalle qui maximise la distance entre les différentes classes (i.e. la variance interclasse) et minimise l'éparpillement au sein de chacune des classes (i.e. la variance intra-classe). Il prend en compte les classes ainsi que leur organisation, et notamment des distances qui les séparent.

Tout critère supervisé (i.e. prenant en compte l'information de classe) s'annule lorsque les classes sont séparées. Ainsi, lorsque la table discrétisée sépare les classes, le critère de Hotelling est nul, alors qu'un critère non supervisé peut encore être utilisé. En poursuivant la discrétisation, la table obtenue contiendra plus d'intervalles et une description plus fine des classes, mais contiendra des données plus corrélées. A l'inverse, un critère d'arrêt autre que la séparation entre classes pourrait être utilisé (dans un contexte de classification hiérarchique) pour que la discrétisation s'arrête avant que les classes ne soient séparées.

Notons la possibilité d'intégrer des données symboliques aux données numériques. L'intégration de ces données

consiste à calculer une extension de la relation d'appartenance R pour ensuite ajouter ces données au treillis. Cette intégration peut aussi être réalisée au cours de l'initialisation de la relation R, avant la discrétisation ; et ainsi servir à affiner le critère de coupe.

2.2 Construction du treillis de Galois

Après la phase de discrétisation vient la construction du treillis. Cette étape est totalement déterminée par la relation d'appartenance R entre objets O et intervalles I issue de la discrétisation. Aucun critère ou paramètre n'est à prendre en compte pour la construction du graphe étant donné qu'il représente toutes les combinaisons possibles de sous-ensembles d'objets et d'intervalles.

Plus précisément, un treillis de Galois est composé d'un ensemble de *concepts* reliés par inclusion, formant ainsi un graphe possédant les propriétés d'un treillis. Un *concept* est un couple objets-intervalles en relation selon R. Plus formellement, c'est un couple (A, B) avec $A \subseteq O$, $B \subseteq I$, $f(A) = B$ et $g(B) = A$ avec : $f(A)$ l'ensemble des intervalles en relation avec les objets de $A \subseteq O : f(A) = \{x \in I \mid pRx \,\forall\, p \in A\}$ et $g(B)$ l'ensemble des objets associés aux intervalles de $B \subseteq I : g(B) = \{p \in O \mid pRx \,\forall\, x \in B\}$.
Les fonctions f et g ainsi définies entre objets et intervalles forment une *correspondance de Galois*.

Deux concepts (A, B) et (A', B') sont reliés par inclusion dans le treillis de Galois si et seulement si $B \subseteq B'$ (ce qui est équivalent à $A' \subseteq A$). Le concept minimal du treillis contient tous les objets O : il s'agit du concept $(O, f(O) = \emptyset)$. Notons que lorsque les caractéristiques non sélectionnées lors de la discrétisation ne sont pas supprimées, elles sont partagées par tous les objets et se retrouvent dans $f(O)$. Duallement, le concept maximal est $(g(I), I)$.

Exemple 2 *La figure 3 représente le treillis de Galois de l'exemple* 1 *obtenu à partir de la table discrétisée 2. Le concept* $(\emptyset, I = \{a_1, a_2, b_1, b_2, c_1, c_2\})$ *est le concept maximal du treillis car* $g(I) = \emptyset$ *(aucun objet n'est en relation avec tous les intervalles à la fois). A l'inverse, si on applique la fonction* f *sur l'ensemble* O *de tous les objets, on obtient un ensemble vide d'intervalles et le concept* $(O, \emptyset)$ *est le concept minimal.*

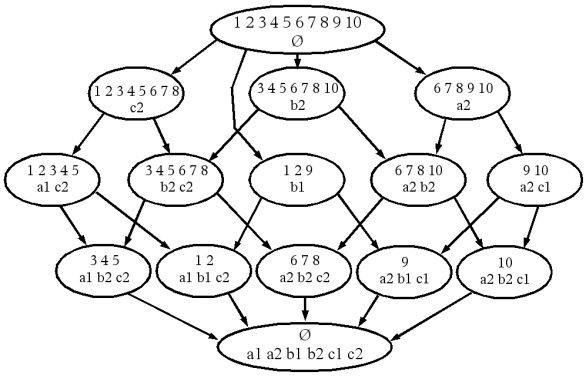

FIG. 3 – Treillis de Galois

La principale limite de l'utilisation du treillis de Galois est dûe à son coût à la fois en temps et en espace. Il existe de nombreux algorithmes de génération du treillis de Galois : Bordat [BOR 86], Ganter [GAN 84], Godin et al. [GOD 91], Nourine et Raynaud [NOU 99] ... Ces algorithmes ont une complexité polynomiale (au mieux quadratique dans [NOU 99]) par concept généré, et dépendent donc de la taille du treillis qui est bornée par $2^{|O+I|}$ dans le pire des cas, et par $|O + I|$ dans le meilleur des cas. Notons cependant que sa taille reste raisonnable en pratique, comme l'illustrent les expérimentations qui en ont déjà été faites [MEP 05]. Dans notre expérimentation, nous utilisons une contruction de treillis de Galois non optimale, mais simple qui se rapproche de l'algorithme de Bordat.

Afin de limiter cette complexité exponentielle, notons la possibilité de générer non pas le treillis lui-même, mais une *représentation* du treillis. Parmi les nombreuses représentations proposées dans la littérature, citons la représentation par un *système de règles d'implication* et plus précisément par la *base Guigues-Duquenne* [OBI 03] ou encore la *base canonique directe* [BER 04] que l'on retrouve notamment en analyse de données [TAO 01]. Une telle représentation permet, outre sa propriété de représentativité condensée du treillis, d'éviter la génération complète du treillis en proposant une *génération à la demande* des concepts du treillis qui sont nécessaires au cours de la phase de classification.

3 Classification

3.1 Classification par navigation dans le treillis

La classification consiste à déterminer la classe de nouveaux objets plus ou moins détériorés par navigation dans le treillis. Le treillis de Galois peut être vu comme un espace de recherche dans lequel on évolue par validation de caractéristiques. La navigation débute à partir du *concept minimal* $(O, f(O))$ où toutes les classes sont *candidates* à la reconnaissance et aucun intervalle n'est validé. Il s'agit ensuite de progresser concept par concept au sein du treillis de Galois par validation de nouveaux intervalles et par conséquent réduction de l'ensemble d'objets, jusqu'à un *concept final* où les objets restants, qui sont en relation avec tous les intervalles validés durant le parcours du graphe, sont tous de la même classe. La navigation est réalisée dans le diagramme de Hasse, c'est à dire la réduction réflexive et transitive du treillis de Galois.

3.2 Description d'une étape élémentaire de classification

Une étape élémentaire de classification consiste à partir d'un concept courant à sélectionner un ensemble d'intervalles S et de faire un choix parmi ces intervalles afin de progresser vers un nouveau concept courant. Plus précisément, S est une famille d'intervalles obtenue à partir des n successeurs $(A_1, B_1), \ldots, (A_n, B_n)$ du concept courant (A, B) par : $S = \bigcup_{i=1}^{n} B_i \backslash B = \{X_1, \ldots, X_n\}$

Les ensembles de S vérifient les propriétés suivantes :

– Ils sont disjoints : $X_i \bigcap X_j = \emptyset,\ \forall i, j \leq n, i \neq j$

– $X_i (i \leq n)$ ne peut contenir 2 intervalles issus d'une même caractéristique $j (j \leq n)$: $|X_i \bigcap I_j| \leq 1$.

Il s'agit ensuite de choisir un ensemble d'intervalles parmi X_i selon un *critère de choix*. Plus précisément : *Choisir X_i parmi $S = \{X_1, \ldots X_n\}$*. Ce choix est essentiel à toute étape élémentaire de classification, base de la navigation dans le treillis et dépend des données. Plus précisément, il est déterminé par un *critère de choix* défini à partir d'une *mesure de distance* entre la signature d'un objet à reconnaître et un intervalle.

Exemple 3 *Dans l'exemple (fig. 3), le choix à partir du concept minimal doit être fait parmi* $S = \{\{c_2\}, \{b_1\}, \{b_2\}, \{a_2\}\}$. *Notons qu'il existe dans le treillis de Galois plusieurs chemins (scénarii de reconnaissance) permettant d'aboutir à une même classe, propriété intéressante dans le cadre d'une classification d'objets détériorés. D'autre part, plusieurs concepts finaux peuvent représenter la même classe. Par exemple, en observant la primitive* b *pour la classe* 4, *on peut voir qu'elle sépare les objets* 9 *et* 10, *pourtant de la même classe. La conséquence pour le treillis, est que deux concepts finaux représentent la classe* 4.

3.3 Mesure de distance

Faire un choix parmi S nécessite l'utilisation d'une *mesure de distance* entre la valeur s_i du symbole s à classifier et un intervalle $x \in I_i$. Une première idée de distance d toute simple pourrait être :

$$d(s_i, x) = \begin{cases} 0 \text{ si } s_i \in x \\ 1 \text{ sinon} \end{cases}$$

Une amélioration permet d'intégrer l'éloignement entre la valeur s_i et le centre de l'intervalle x. Dans notre expérimentation, nous avons choisi d'utiliser une mesure de distance basée sur le ratio entre la distance euclidienne de s_i au centre de l'intervalle et la demi-longueur de celui-ci :

$$d(s_i, x) = \frac{\sqrt{(s_i - x_{milieu})^2}}{\sqrt{(x_{borneInf} - x_{milieu})^2}}$$

Par abus de notation, notons $d(s, x)$ au lieu de $d(s_i, x)$ cette mesure de distance, afin de pouvoir l'étendre à un ensemble d'intervalles $X \subseteq I$: $d(X) = \frac{1}{|X|} \sum_{x \in X} d(s, x)$.

Dans notre cas expérimental, les symboles à reconnaître sont bruités, ce qui entraîne des modifications des valeurs de leur signature qui risquent de ne plus être incluses dans les intervalles correspondants à leur classe. Il est donc intéressant de rendre les bornes des intervalles plus souples et d'intégrer cette dérive au sein du processus de reconnaissance par l'utilisation non pas d'intervalles, mais de nombres flous. Dans ce cas, la mesure de distance peut correspondre à la *fonction d'appartenance* du nombre flou.

Rappelons qu'un nombre flou A sur un univers U est défini par une *fonction d'appartenance* μ_A (ou μ) qui précise le *degré de vraisemblance* de l'assertion $x \in A$:

$$\begin{array}{rcl} \mu : U & \rightarrow & [0, 1] \\ x & \mapsto & \mu_A(x) \end{array}$$

Un nombre flou A est habituellement défini par un trapèze $[a, b, c, d]$ de *support* $[a, d]$ et de *noyau* $[b, c]$.

Il existe quelques cas particuliers de nombres flous : les triangulaires où $b = c$, les rectangulaires où $a = b$ et $c = d$, et les symétriques de noyau = $[m - t, m + t]$ et de support = $[m - s*t, m + s*t]$, avec m le milieu du nombre flou, $t = |m - a| = |m - d|$ et $s = |m - b| = |m - c|$. Une représentation des caractéristiques par des intervalles correspond en logique floue à l'utilisation de nombres flous rectangulaires.

Plusieurs extensions d'un intervalle (issu de la discrétisation) en un nombre flou sont envisageables. Nous en avons retenus 2 illustrées par les Fig. 4 et 5, qui prennent en compte la distribution des valeurs de l'intervalle.

FIG. 4 – Exemple 1 de nombre flou formé à partir d'un intervalle

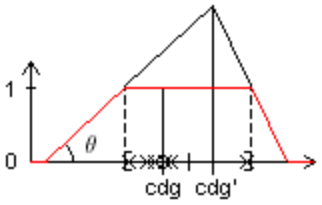

FIG. 5 – Exemple 2 de nombre flou formé à partir d'un intervalle

Dans les figures 4 et 5, un intervalle x est symbolisé par ses 2 bornes et par un ensemble de croix représentant la distribution de ses valeurs V_x. Dans ces 2 exemples, nous nous basons sur l'utilisation du centre de gravité de la distribution cdg. Le nombre flou de l'exemple 4 est triangulaire et symétrique. La pente est déterminée par la borne de l'intervalle la plus éloignée de cdg. Pour l'exemple 5, il faut calculer cdg' le symétrique de cdg par rapport au milieu de l'intervalle. Les pentes sont alors obtenues à partir de cdg'. Cet exemple de nombre flou est paramétrable selon un degré de flou θ : plus θ est petit, plus le nombre flou aura un suppport large ; et plus θ est grand, plus le nombre flou sera proche d'un nombre flou rectangulaire.

3.4 Critères de choix

Etant donnés $S = \{X_1, \ldots, X_n\}$ la famille d'intervalles sélectionnés, et $d(s, X)$ la *mesure de distance* entre l'objet s à classifier et un ensemble d'intervalles $X_i \subseteq I$, il s'agit de mettre en place un critère de choix pour sélectionner X_i parmi S. De nombreux critères de choix peuvent être définis, dont une liste exhaustive n'est pas envisageable. Voici quelques exemples simples :

1 choisir i tel que $d(s, X_i)$ est minimal.

2 choisir i tel que $|X_i \bigcap I_k| = |\{x \in X_i \bigcap I_k\}|$ est maximal, avec I_k l'ensemble des k premiers intervalles de S triés par ordre croissant selon la distance $d(s, x)$.

3 choisir i tel que $|\{x \in X_i$ tel que $d(s, x) < d_c\}|$ est maximal, avec d_c une constante que l'on peut assimiler à un degré de flou.

4 appliquer le critère n°3 avec $d_c = 1$ (ce qui équivaut à un nombre flou rectangulaire dont le support est égal aux bornes de l'intervalle). Puis en cas de choix multiples, appliquer le critère n°3 avec $d_c = 1,1$ (le support du nombre flou rectangulaire est alors élargi au-dela des bornes de l'intervalle de manière proportionnelle à sa taille). Puis en cas de choix multiples, appliquer le critère n°1 (ce qui équivaut à un nombre flou symétrique dont le centre est le milieu de l'intervalle).

Le critère n°1, défini de manière globale sur chaque X_i, possède l'inconvénient de noyer le bruit. Le second critère suit le principe du k-ppv, et le critère n°3 est un cas particulier du second. Pour l'expérimentation, nous avons retenu le critère de choix n°4 qui nous donne les meilleurs résultats [GUI 06].

4 Résultats expérimentaux

Une première étude expérimentale nous a permis de définir les différents paramètres de la méthode [GUI 06] : la signature décrivant les symboles, le critère de coupe et le critère de choix. Le but étant de déterminer pour chaque paramètre les choix les plus favorables à une bonne reconnaissance des symboles. La bonne lisibilité du treillis de Galois nous a notamment permis d'évaluer l'efficacité de la mesure de distance et du critère de choix. Nous avons ainsi choisi d'utiliser la signature de Radon [TAB 03] pour caractériser les symboles, le critère de coupe de Hotelling pour effectuer la discrétisation, et le critère de choix n°4 pour la classification.

Dans cette nouvelle étude expérimentale, nous comparons le treillis de Galois à 2 autres classifieurs : le k-ppv (k plus proches voisins) et le classifieur bayésien. Nous proposons une étude comparative sur l'évolution du taux de reconnaissance en fonction de la taille de l'ensemble d'apprentissage.

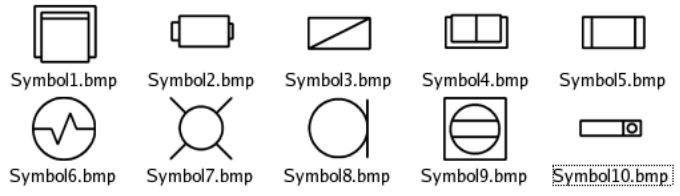

FIG. 6 – Exemples de symboles extraits de la base GREC2003

Nous avons réalisé nos tests sur 2 séries de 10 classes (à savoir les classes 1-10 et les classes 11-20). Sur ces 2 séries de classes, nous avons construit des ensembles d'apprentissage contenant de 5 à 25 symboles par classe. Les symboles de GREC2003 [GRE 03] (voir Fig. 6) contiennent pour chaque classe : 1 symbole parfait et 90 symboles bruités. Pour former un ensemble d'apprentissage contenant par exemple 5 symboles par classe, nous prenons au hasard 4 symboles parmi les 90 symboles bruités, ainsi que le symbole parfait. Les symboles bruités pris au hasard sont retirés de l'ensemble de test. Pour chaque taille de l'ensemble d'apprentissage, nous avons mis en place 5 tests. Nous présentons les résultats moyens obtenus à ces 5 tests. Nous avons testé l'évolution des taux de reconnaissance des 3 méthodes pour les 10 premières caractéristiques de la signature de Radon, ainsi que sur la totalité de la signature (voir Fig. 7 et 8).

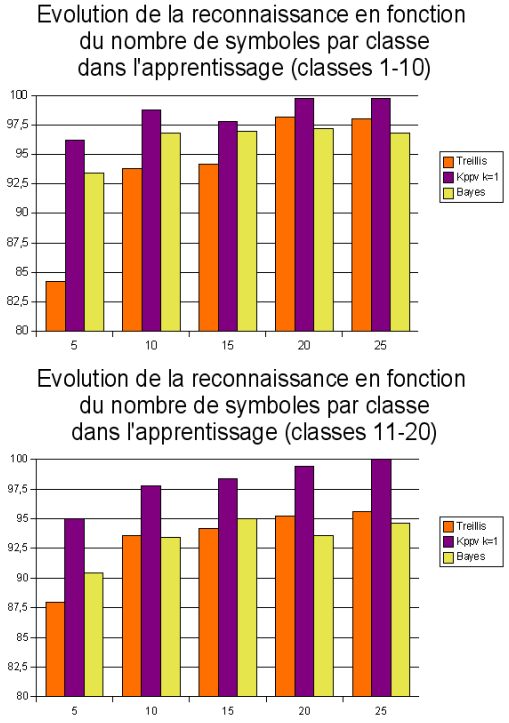

FIG. 7 – Evolution de la reconnaissance en fonction du nombre de symboles par classe dans l'ensemble d'apprentissage (signature de Radon sur 10 caractéristiques)

Les résultats obtenus fluctuent selon les données utilisées, la taille de l'ensemble d'apprentissage et celle de la signature. Cependant, on peut observer une augmentation des taux de reconnaissance lorsque la taille de l'ensemble d'apprentissage augmente. Le treillis de Galois donne des taux proches de ceux des autres classifieurs bien qu'il n'utilise pas la totalité de la signature. En effet, sur les 10 ou les 50 caractéristiques de la signature, seules 6 à 8 sont sélectionnées par le critère de Hotelling et utilisées pour construire le treillis. L'inconvénient majeur du classifieur k-ppv est qu'il requiert le stockage de l'ensemble des données de la base d'apprentissage. De son côté, le classifieur bayésien nécessite de faire une hypothèse (normale, uniforme,...) sur la distribution des données qui implique la réalisation d'une étude préalable des données. Le treillis quant à lui n'est pas restreint par ces 2 limites. D'autre part, il possède l'avantage tout comme l'arbre de décision d'une bonne lisibilité, avec en plus une meilleure robustesse au bruit : plusieurs scenarii de classification per-

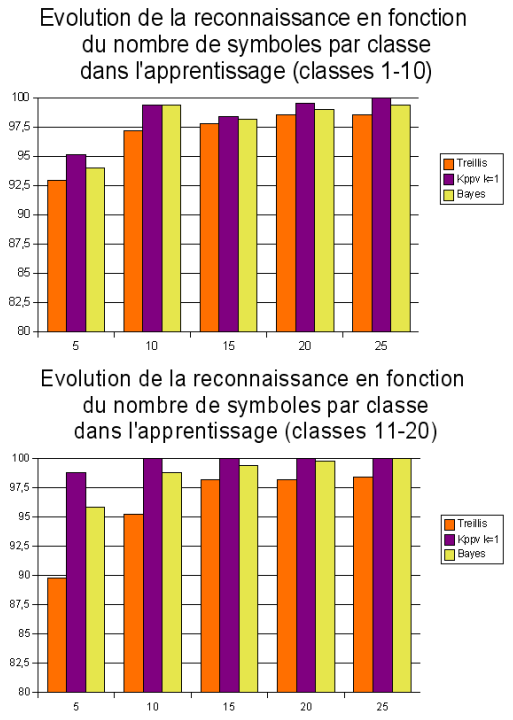

FIG. 8 – Evolution de la reconnaissance en fonction du nombre de symboles par classe dans l'ensemble d'apprentissage (signature de Radon sur 50 caractéristiques)

mettent d'aboutir à la reconnaissance d'une même classe.

5 Conclusion

L'aptitude de sélection de caractéristiques du treillis de Galois le rend difficile à comparer avec les autres classifieurs, et une étude comparative plus formelle serait nécessaire. Nous aimerions entre autre tester ces 3 classifieurs sur les symboles de la base GREC2005 qui sont plus bruités que ceux de la base GREC2003.

En ce qui concerne l'utilisation de la théorie du flou, nous souhaiterions approfondir nos recherches afin de l'intégrer au mieux à notre système. Plusieurs pistes sont envisageables : transformer les intervalles utilisés en sous-ensembles flous, construire des sous-ensembles flous en fonction de la distribution des valeurs de l'apprentissage, ou encore étudier la construction des arbres de décision flous et l'adapter à notre graphe.

Nous développons également une signature structurelle que nous allons intégrer aux signatures statistiques (telle que la signature de Radon) dans le treillis de Galois. D'autre part, nous souhaitons une expérimentation plus poussée des fonctions d'appartenances de nombres flous présentées dans la partie 3.3.

Enfin, nous pensons mettre en place un système de classification hierarchique basée sur l'utilisation successive de plusieurs treillis. Le but d'un tel système hierarchique est d'affiner la recherche de la classe à chaque étape par ajout de nouvelles caractéristiques dynamiquement selon le scénario de classification en cours. Une telle approche est possible par ajout d'un critère d'arrêt de la discrétisation autre que la séparation de toutes les classes. Ainsi, le treillis contruit ne permettrait plus de reconnaître une classe parmi les autres, mais un ensemble de classes candidates, parmi les autres ensembles de classes. Cet ensemble de classes candidates serait alors traité par un nouveau treillis de Galois. Cette succession de classifications de plus en plus détaillées permettrait d'accélérer le processus de reconnaissance car les treillis construits à chaque étape seraient plus petits, de traiter une très grande quantité de classes, mais aussi de diminuer considérablement le nombre d'erreurs de classification.

6 Biliographie

Références

[BER 04] BERTET K., NEBUT M., Efficient algorithms on the family associated to an implicationnal system, *DMTCS*, vol. 6, n° 2, 2004, pp. 315-338.

[BOR 86] BORDAT J., Calcul pratique du treillis de Galois d'une correspondance, *Math. Sci. Hum.*, vol. 96, 1986, pp. 31-47.

[GAN 84] GANTER B., Two basic algorithms in concept analysis, *Technische Hochschule Darmstadt (Preprint 831)*, , 1984.

[GOD 91] GODIN R., MISSAOUI R., , ALAOUI H., Learning algorithms using a Galois lattice structure, *Third International Conference on Tools for Artificial Intelligence, San Jose, Calif.*, , 1991, pp. 22-29, IEEE Computer Society Press.

[GRE 03] GREC, www.cvc.uab.es/grec2003/SymRec-Contest/index.htm, 2003.

[GUI 06] GUILLAS S., BERTET K., , OGIER J. M., Comment utiliser le treillis de Galois en reconnaissance d'images ?, *Atelier ECOI, Conférence EGC 2006*, , 2006, pp. 31-36.

[MEP 05] MEPHU NGUIFO E., NJIWOUA P., Treillis des concepts et classification supervisée, *Technique et Science Informatiques (à paraître), RSTI)*, Hermès - Lavoisier, Paris, France, 2005.

[NOU 99] NOURINE L., RAYNAUD O., A fast algorithm for building lattices, *Third International Conference on Orders, Algorithms and Applications*, Montpellier, France, august 1999.

[OBI 03] OBIEDKOV S., DUQUENNE V., Incremental Construction of the Canonical Implication Basis, *Fourth International Conference Journée de l'Informatique messine*, 2003, pp. 15–23, submitted to Discrete Applied Mathematics.

[TAB 03] TABBONE S., WENDLING L., Recherche d'images par le contenu à l'aide de la transformée de Radon, *Technique et Science Informatiques*, , 2003.

[TAO 01] TAOUIL R., BASTIDE Y., Computing proper implications, *Proceedings of ICCS-2001 Internationnal Workshop on Concept Lattices-Based Theory, Methods and tools for Knowledge Discovery in Databases*, , 2001, pp. 290-303.