Les treillis de Galois : un outil pour la sélection de primitives ?

Concept lattices : a tool for primitives selection ?

Stéphanie Guillas, Karell Bertet et Jean-Marc Ogier

L31 Université de La Rochelle, av M. Crépeau,
17042 La Rochelle Cedex 1, France
stephanie.guillas@univ-lr.fr, kbertet@univ-lr.fr, jmogier@univ-lr.fr

Manuscrit reçu le 19 mai 2004

Résumé et mots clés

Dans ce papier, nous présentons la problématique de la reconnaissance d'images détériorées et plus particulièrement l'étape de sélection de primitives au sein d'un traitement de classification supervisée. Cette étape de sélection a lieu après que la segmentation et l'extraction des descripteurs statistiques sur des images documentaires aient été réalisées. Nous exposons en détail l'utilisation d'un arbre de décision, afin de l'harmoniser puis la comparer avec une approche moins étudiée utilisant un treillis de Galois.

Reconnaissance de symboles, treillis de Galois, arbre de décision, images détériorées, descripteurs statistiques.

Abstract and key words

In this paper, we present the problem of noisy images recognition and in particular the stage of primitives selection in a classification process. This selection stage appears after segmentation and statistical describers extraction on documentary images are realized. We describe precisely the use of decision tree in order to harmonize and compare it with another less studied method based on a concept lattice.

Symbol recognition, Concept lattice, Decision tree, Noisy images, Statistical describers.

1. Introduction

À ce jour, une part importante des documents « papier » n'a pas intégré un système informatisé de circulation et de stockage de l'information. C'est particulièrement vrai pour les dessins techniques, les graphiques et les données cartographiques pour lesquels la numérisation est très coûteuse en temps et en argent. Filpski [FF92] indique en 1992 que 3,5 milliards de documents techniques sont disponibles sur support papier (les USA et le Canada), et 26 millions de documents « papier » sont créés chaque année. Cette estimation même ancienne montre l'ampleur de la tâche d'intégration compte tenu de l'actualité des systèmes d'information documentaire.

Si on s'intéresse plus particulièrement à la dématérialisation de cette information, aucun logiciel automatique de rétroconversion n'offre à notre connaissance d'outils génériques capables de transformer ces documents « papier » vers un format numérique facilement manipulable. En fait, le problème est principalement dû à la grande diversité des supports et des représentations graphiques qui vont du schéma de pièces automobiles au plan d'un gazoduc, en passant par le dessin d'une parcelle sur un plan cadastral.
Sur un plan scientifique, la communauté travaillant sur ces sujets rencontre encore un certain nombre de verrous liés à la multi-représentation de symboles, aux contraintes de multi-orientation et multi-échelle, et à la très grande densité de certains documents, rendant très difficile la caractérisation des objets à reconnaître.

Le travail présenté dans cet article se place dans le contexte de la rétroconversion automatique et propose une première approche théorique concernant l'exploitation des treillis de Galois (aussi appelé treillis des concepts) pour la reconnaissance automatique d'objets graphiques, comme les caractères ou les symboles, sous la contrainte classique de multi-orientation et multi-échelle.
Parmi les techniques diverses et variées utilisées en classification supervisée, celles basées sur un treillis de Galois ont fait l'objet d'études comparatives dans de récents travaux [KO01, MNN05]. 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.
Nous nous proposons dans cet article de décrire de façon générale l'utilisation de cette structure en classification, puis d'apporter quelques éléments de comparaison avec les techniques bien reconnues en classification basées sur un arbre de décision.
Cet article se décompose en différentes parties : la première partie présente une synthèse bibliographique sur la reconnaissance de texte et de symboles. La seconde partie est une description précise de l'utilisation d'un arbre de décision. Cette description servira de comparaison à une approche basée sur un treillis de Galois décrite dans la troisième partie. Chacune de ces deux parties décrit une approche basée sur une structure pour la sélection de primitives, et pour un processus de classification. Nous finirons par une dernière partie apportant des éléments de comparaison entre ces deux approches, et quelques résultats expérimentaux.

2. Contexte et classification supervisée

2.1 Introduction

L'analyse et la gestion des informations textuelles et symboliques est fondamentale dans le cadre de la lecture d'un document, cette couche d'informations étant sémantiquement très riche. Chacune des informations textuelles représente une part d'information exploitable par le lecteur du document technique notamment lorsqu'il s'agit de plans ou de schémas (cotations sur les schémas mécaniques, toponyme sur une carte routière, etc), et ceci en adéquation avec la symbologie adoptée sur le schéma. La connaissance précise de cette information représente donc un enjeu important pour les systèmes d'interprétation de documents.
La littérature et les produits commerciaux font aujourd'hui une large place aux systèmes de reconnaissance optique des caractères (*Optical Character Recognition*) dans un cadre classique : caractères et alignements horizontaux, fontes connues. Cependant, la généralisation à d'autres orientations et d'autres alignements des caractères réduit d'autant le nombre de propositions. Il est même possible de dire qu'aujourd'hui, il n'existe pas de produits industriellement fiables permettant de reconnaître les caractères et les symboles acceptant des variabilités intrinsèques. Or, le contexte de la reconnaissance des éléments textuels et symboliques présents sur des documents techniques fait référence à ce type d'outils. Dans la plupart des cas, les caractères et les symboles sont écrits manuellement, avec contraintes, et avec une orientation relative à celle de l'objet décrit. Ainsi, le nom d'une route suivra l'orientation de cette dernière.
Concernant cette problématique du traitement automatique des informations textuelles, lorsque celles-ci sont facilement « séparables » sur le document (*i.e.* lorsque les caractères, symboles ou objets linéaires sont sur des couches bien déconnectées les unes des autres), différentes techniques sont généralement utilisées de manière séquentielle. Celles-ci concernent la segmentation des caractères, leur reconnaissance individuelle, puis leur regroupement en mots ou toponymes.
Concernant la représentation des symboles, la problématique est souvent plus difficile dans la mesure où ceux-ci sont très fréquemment connectés physiquement à d'autres objets sur l'image, ajoutant une problématique de détection/segmentation suivant la stratégie de reconnaissance adoptée.
Ces constatations mettent en évidence le fait qu'un « verrou méthodologique » voire « scientifique » important demeure au sujet de la reconnaissance des informations textuelles et symboliques qui sont inter-connectées ou connectées à d'autres objets. De manière générale, les problématiques de reconnaissance de caractères et de symboles sont considérées différemment, à l'exception de quelques contributions qui considèrent le texte comme une symbolique particulière [AOC+01].

2.2 Approches classiques de reconnaissance

En matière de reconnaissance de symboles sur des documents graphiques, il est possible de distinguer deux grandes catégories d'approches, suivant que les formes à reconnaître sont originellement connectées ou non à d'autres objets sur l'image.
D'une part, la littérature propose un ensemble très important d'approches à base de reconnaissance structurelle. D'autre part, on trouve également de nombreuses contributions à base de signatures discrètes et/ou statistiques des formes, suivant des schémas en appui sur des classifieurs plus ou moins élaborés.
Nous proposons ci-dessous un panorama synthétique de ces différentes approches.

Approches à base de description structurelle

La représentation/reconnaissance structurelle est une technique fréquemment utilisée en reconnaissance de symboles, comme en attestent de nombreuses publications [LVSM01, AS98,

PN93]. Cependant de nombreuses autres problématiques s'appuient également sur l'analyse structurelle ; citons entre autre l'apprentissage, l'indexation, la structuration des données, etc… Plus proche de la structure du symbole, elle possède cependant l'inconvénient de ne pas être robuste au bruit.

Approches à base de descriptions statistiques

Elles s'appuient généralement sur un calcul de signature, introduite dans un système de classification. En fonction du contexte, les chercheurs sont souvent amenés à préalablement segmenter les objets, notamment lorsque ceux-ci sont connectés à d'autres objets sur l'image, soulevant ainsi le classique paradigme segmentation/reconnaissance. Dans ce paragraphe, nous abordons la problématique de la segmentation, puis celle de la caractérisation de formes.
Lorsque les objets sont connectés à d'autres formes (lignes, symboles, caractères) sur l'image, une étape préalable de *segmentation* est souvent préconisée par les approches statistiques, en appui sur des stratégies dépendant du contexte de représentation des objets (densité, taille des symboles, …) [TOM96].
Cette étape vise à isoler les caractères des autres éléments du document. La plupart des auteurs proposent de détecter les caractères en isolant les composantes connexes de taille prédéfinie [TTJ97, FK98, LU98, SHA+92, LK94, LCD97].
Après segmentation, il est possible de passer à l'étape de *caractérisation des formes*. La reconnaissance de formes répondant aux contraintes d'invariance vis à vis de transformations telles que les similitudes est un point essentiel dans la conception de systèmes fiables pour l'interprétation de documents techniques. Dans ce champ de recherche, nous avons montré dans une contribution précédente que l'on peut distinguer trois principales catégories d'approches [AOC+01].
La première suggère un calcul préliminaire de l'orientation de la forme, et essaie, par l'intermédiaire d'étapes de normalisation et de rotation, d'obtenir une forme dans une position de référence, qui peut ainsi être introduite dans un système de classification. Une seconde approche classique consiste à utiliser un classifieur neuronal prenant en entrée l'image de la forme et qui rend lui même, lors de sa phase d'apprentissage, le problème invariant aux transformations [FOTK92]. La dernière approche, qui est probablement la plus utilisée, consiste à extraire de la forme un ensemble de descripteurs invariants aux transformations, avant d'alimenter un système de classification. De très bonnes descriptions de l'état de l'art dans ce domaine de la description de formes peuvent être trouvées dans quelques contributions [TJT96, AOC+01]. Adam [AOC+01] propose une catégorisation des vecteurs utilisés pour décrire les formes indépendamment de leur position, de leur taille et de leur orientation suivant deux catégories.

Descripteurs basés sur l'aspect global de la forme.

De nombreuses caractéristiques peuvent être utilisées pour décrire l'aspect global d'une forme. Depuis les travaux de Hu en 1961, les moments invariants [HU62], qui sont basés sur des combinaisons de moments réguliers, ont été très fréquemment utilisés [RSV96]. Parmi ceux-ci, on peut citer les moments de Zernike [TEA80, KH90B, KH90A, LP98] qui restent une référence dans le domaine, les pseudo-moments de Zernike [TEA80], les moments de Bamieh [BDF86], ou les moments de Legendre [CLD96].
Ces moments invariants offrent généralement des propriétés de reconstructibilité, ce qui permet d'assurer que les primitives extraites contiennent la plus grande partie de l'information incluse dans la forme étudiée. De bonnes études comparatives concernant ces moments invariants peuvent être trouvées dans [TC88] et [BSA91], chacune d'elle soulignant la supériorité des moments de Zernike quant aux performances de reconnaissance. Néanmoins, ces études prouvent également que les approches à base de moments sont sensibles au bruit et qu'elles sont coûteuses en terme de temps de calcul, et ce, malgré les nombreuses méthodes d'optimisation et/ou de réduction de complexité qui existent dans la littérature [DBN92, LS91].

Descripteurs basés sur une approche locale.

À l'inverse des approches présentées ci-dessus, une description invariante géométrique peut aussi être effectuée en utilisant des primitives qui localement, sont théoriquement très informatives. Par exemple, les contours sont fréquemment utilisés pour obtenir des descriptions invariantes des formes par utilisation des descripteurs de Fourier [PL92] ou des descripteurs elliptiques de Fourier [LIN87]. Taxt [TOD90] a mené une étude comparative entre ces descripteurs qui montre leur intérêt potentiel, surtout en ce qui concerne leur simplicité et leur robustesse. En particulier, Taxt [TOD90] propose de retenir les moments elliptiques de Kuhl [KG82] lorsque l'orientation du caractère est connue. Ce type de technique est utilisé également par Trier [TTJ97] pour la reconnaissance des caractères appliquée à des cartes hydrographiques.
Les primitives circulaires qui sont, par définition, bien adaptées à la reconnaissance invariante à la rotation, ont été utilisées dans [KIT92]. Celles-ci sont basées sur l'analyse de la forme par l'intermédiaire d'un ensemble de cercles. Lefrère [LEF99] propose en particulier un ensemble de sondes circulaires permettant de décrire la forme à partir de son centre de gravité. Ce type d'approche reste néanmoins très sensible aux variabilités et à la présence de bruit. Une étude comparative, disponible dans [DBM77], montre que ce type de primitives circulaires permet l'obtention de meilleurs résultats que les moments de Hu.

2.3 Problématique de classification

Chacune des techniques issues des approches classiques propose généralement un schéma de reconnaissance, en appui sur des principes de classification et/ou de combinaison de classifieurs, après une éventuelle sélection de primitives pertinentes.
En effet, l'analyse de la littérature met en évidence le besoin criant d'adapter les processus de reconnaissance à la notion de

contexte, s'appuyant sur des stratégies adaptées de systèmes coopérants, et/ou sur des jeux de primitives sélectionnées à partir d'algorithmes d'optimisation ([AOC+99]).
De manière générale, une catégorisation possible des classifieurs est de les organiser en fonction de la nature de leurs sorties [GML]. En effet, selon les méthodes employées, les sorties sont relatives à :
- une mesure de distance entre l'objet et les prototypes qui peut être explicite (distance entre chaînes, Mise En Correspondance (M.E.C.), Mise En Coincidence (M.E.Co.) ou implicite (Multi-Layers Perceptron (MLP), Learning Vector Quantization (LVQ), …)
- une mesure de distance sur les densités de probabilité (Kppv, Noyaux de Parzen …)
- une mesure de probabilité (*a posteriori* : Classifieur Bayésien, ou bien associée à des règles de production stochastiques)
L'objectif général du classement d'une forme consiste à affecter une interprétation à un vecteur de caractéristiques d'une forme. Une interprétation éventuellement associée à une mesure de similarité. Dans certains types de situations, on évoque également la notion de rejet, lorsque le classifieur considère que le vecteur analysé ne peut être associé à aucune classe connue.
Suivant le contexte, le classement peut être réalisé suivant des schémas supervisés ou non supervisés. Ces schémas sont généralement précédés d'une phase de sélection de primitives, visant à ne retenir que les primitives pertinentes au regard du contexte analysé, ou ayant pour but de faire face au fameux problème de la malédiction de la dimensionnalité.

2.4 Problématique de sélection de primitives

Comme nous l'avons évoqué dans les paragraphes précédents, les schémas de reconnaissance de formes sur les documents s'appuient souvent sur une étape préalable de sélection de primitives, basée sur des techniques d'optimisation de critères. Le lecteur pourra d'ailleurs trouver de bons états de l'art dans les références [ZJ96] ou [SEM04]. On peut classer les méthodes d'optimisation de critères (recherche d'extrema) en quatre principales catégories [GOL89] :
Les méthodes fondées sur le calcul. De nombreuses méthodes de ce genre existent (moindre carrés, gradient…). Elles peuvent être subdivisées en deux classes principales. Les méthodes indirectes qui cherchent à atteindre les extrema locaux en résolvant des systèmes d'équations qui annulent le gradient ; et les méthodes directes qui recherchent quant à elles les optima locaux en émettant des hypothèses sur la fonction et en se déplaçant dans une direction dépendante de son gradient. Ces méthodes sont simples à mettre en œuvre mais présentent deux inconvénients majeurs. D'une part, elles nécessitent l'existence de dérivées de la fonction et d'autre part, elles ont la fâcheuse tendance à s'enfermer dans les extrema locaux.
Les procédures énumératives. Leur idée générale est simple. L'algorithme examine les valeurs de chacun des points de l'espace un par un et retient le meilleur. Bien que la simplicité de telles procédures soit attirante, elles sont abandonnées, dans la majeure partie des cas pratiques, à cause de la taille de l'espace des possibilités à explorer.
Les algorithmes de recherche aléatoire. Ces méthodes explorent aléatoirement l'espace des solutions et mémorisent le meilleur élément. Elles sont elles aussi peu robustes et manquent d'efficacité dans la plupart des cas.
Les méthodes dites pseudo-aléatoires. Elles utilisent également le hasard en tentant de le contrôler par le but à atteindre et non de manière totalement désordonnée. On retrouve dans cette catégorie la méthode du recuit simulé mais aussi les algorithmes génétiques.

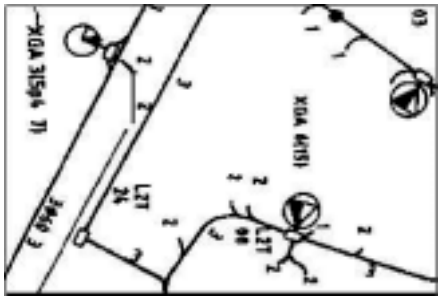


Figure 1. Exemple d'image originale.

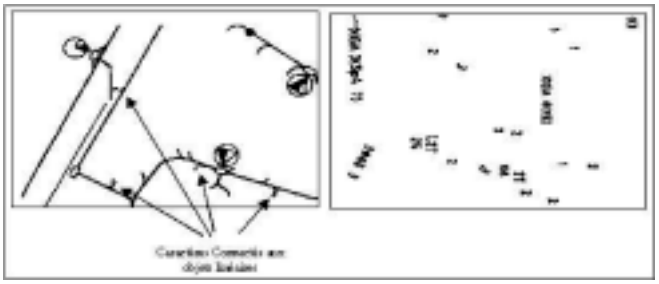


Figure 2. Caractères connectés au réseau (à gauche) Caractères isolés (à droite).

2.5 Bilan concernant les problèmes de reconnaissance de symboles

Dans le domaine de la reconnaissance de formes invariante à différentes transformations géométriques, la plupart des chercheurs [TJT96] s'accordent pour dire que chaque étape du processus est importante. Un grand nombre de types de primitives est rapporté dans la littérature, de même qu'il existe de nombreuses techniques de sélection de primitives, de classification de primitives et de combinaisons de classifieurs.
Néanmoins l'analyse de la littérature met en évidence plusieurs difficultés auxquelles l'ensemble de ces approches tentent de répondre plus ou moins partiellement.
Une des premières difficultés est l'adaptation à la notion de contexte, visant à définir des scénarii de reconnaissance adéquats à une problématique particulière, tout en intégrant la pos-

sibilité d'évolution des dispositifs (définition de nouvelles classes par exemple, prises en compte de façon incrémentale par le dispositif sans qu'il ne soit remis en cause). Une autre difficulté réside dans la problématique de combinaison de schéma de reconnaissance, intégrant des descriptions structurelles et statistiques des formes, sans « distortion » préalable des susdits schémas.

Enfin, la problématique de la sélection des primitives pertinentes, adaptée à un contexte particulier, en adéquation avec des systèmes évolutifs reste un problème ouvert et pas toujours explicite. En effet, les schémas d'optimisation généralement utilisés pour cette étape essentielle de sélection de primitives ne permettent pas toujours de rendre explicite les choix des primitives retenus, du fait de l'effet « boite noire » de ces dispositifs. Leur lisibilité est en effet un critère de comparaison important car la facilité à définir les paramètres du dispositif en dépend.

Nous proposons dans cet article une première contribution concernant la reconnaissance de symboles sur des documents graphiques, sur la base de l'utilisation des treillis de Galois.

En effet, les treillis semblent apporter des réponses intéressantes à l'ensemble des difficultés évoquées précédemment, de par leur possibilité quasi-naturelles d'intégration de descriptions statistiques/structurelles, et également grâce à leurs facultés de validation de primitives pertinentes au regard d'un contexte particulier. Le caractère évolutif de l'approche proposée, permettant une adaptation dynamique à des évolutions du paysage des classes, offre également des perspectives intéressantes.

3. Arbre de décision

Afin d'étudier l'intérêt d'une approche basée sur un treillis de Galois, il nous a paru important de l'harmoniser avec celle très proche basée sur un arbre de décision. En effet, ces deux approches regroupent à la fois l'étape de sélection de primitives et celle de classification, et diffèrent essentiellement par la structure qui y est manipulée. Cette harmonisation nous permet d'identifier des éléments de comparaisons importants, et nécessite une description précise de l'arbre de décision, description qui fait l'objet de cette partie.

Les arbres de décision sont largement utilisés dans le domaine de la classification. Ils ont notamment fait l'objet de nombreuses recherches par le passé et sont encore actuellement un axe de réflexion important [BK99, CF04].

3.1 Description et principe de sélection

Le principe de l'arbre de décision est de diviser récursivement les objets d'un ensemble par des tests sur les primitives, jusqu'à obtenir des sous-ensembles ne contenant que des objets appartenant à une même classe. Chaque objet est décrit par l'ensemble des primitives. Les primitives peuvent être de type binaire, qualitatif (prend sa valeur parmi un ensemble fini), continu (valeur réelle).

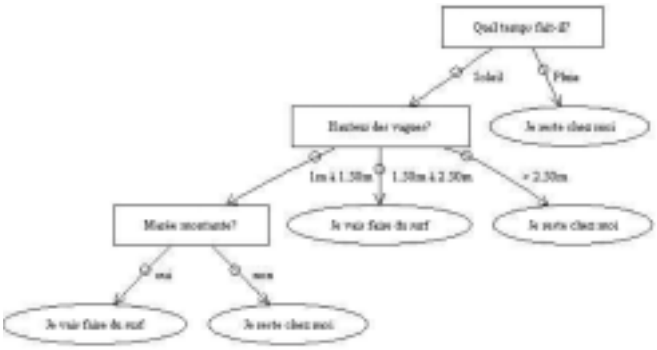


Figure 3. Exemple d'arbre de décision.

Un exemple d'arbre de décision est présenté en figure 3. Il s'agit de déterminer si la personne va faire du surf ou bien rester chez elle.

Les nœuds d'un arbre de décision sont appelés *nœuds de décision* car chaque nœud décrit un test sur l'une des primitives qui doit permettre de partitionner les objets. On peut ainsi associer à chaque nœud de décision l'ensemble D de primitives à tester. Le test doit donc être applicable à l'ensemble des objets. À chaque réponse au test est associée une *branche* qui conduit vers un autre nœud de décision ou bien vers une feuille. Les *feuilles* sont les nœuds terminaux de l'arbre. Chaque feuille correspond à une classe, c'est-à-dire à une prise de décision. Plusieurs feuilles peuvent représenter la même classe.

Lors d'une procédure de classification, on parcourt l'arbre à partir de sa racine (le premier nœud de décision), puis on progresse vers d'autres nœuds de décision en fonction des réponses aux tests, jusqu'à atteindre une feuille, c'est-à-dire trouver la classe de l'objet. Un nœud de décision correspond à une étape élémentaire de classification qui se caractérise complètement par :

- l'ensemble D des *primitives sélectionnées* pour le test, et associées au nœud,
- l'ensemble V des *primitives validées*, depuis l'étape initiale, la racine,
- l'ensemble C des *classes candidates* à la classification, ensemble qui se déduit de l'ensemble N des objets testés au nœud courant. Si l'indication de la classe d'un objet x est donnée par $c(x)$, on peut étendre cette indication de classe à un ensemble d'objets N, où $c(N)$ est l'ensemble des classes des objets de N

À l'étape initiale de classification, aucune primitive validée n'est associée au nœud ($V = \emptyset$) et toutes les classes sont candidates ($|C|$ est maximal). À l'étape finale, toutes les primitives nécessaires à la classification de l'objet sont associées au nœud, ainsi que la classe obtenue ($|C| = 1$).

3.2 Discrétisation

En général, dans les arbres de décision, les tests effectués au niveau des nœuds sont conçus pour des primitives de type qualitatif. Il s'agit par exemple de qualifier la primitive saison (qui prend ses valeurs parmi un ensemble de cardinal fini : printemps, été, automne ou hiver). Cependant, il est parfois nécessaire d'utiliser des primitives continues, qui évoluent dans le domaine des réels. Le découpage en intervalles disjoints des valeurs prises par une primitive continue permet sa catégorisation et donc son utilisation dans les arbres de décision. Ce phénomène de découpage est appelé discrétisation.
Les méthodes de discrétisation peuvent être organisées selon trois axes :
- *supervisées/non-supervisées* : dans le cas de la discrétisation supervisée, on tient compte des classes des objets et de leurs similarités pour effectuer le découpage, alors que pour la discrétisation non-supervisée seules les similarités entre les objets sont considérées.
- *globales/locales* : pour la discrétisation globale, les intervalles sont contruits et fixés avant de réaliser la construction de l'arbre de décision. Au contraire, en discrétisation locale, le découpage est réalisé au fur et à mesure de la construcion de l'arbre. Ainsi à chaque nœud, on divise les ensembles d'objets concernés en sous-ensembles.
- *statiques/dynamiques* : la discrétisation statique est la stratégie la plus utilisée. Il s'agit de traiter les primitives indépendamment les unes des autres. À l'inverse, la discrétisation dynamique intègre l'information de toutes les primitives simultanément pour construire les intervalles, ce qui permet la prise en compte d'une éventuelle corrélation entre les primitives.
Dans [DKS95], les auteurs réalisent une comparaison expérimentale entre les méthodes supervisées et non-supervisées. Dans leurs tests d'évaluation, ils étudient les résultats sur des jeux de données réelles pour trois méthodes : le découpage en intervalles égaux, l'algorithme 1RD (One-Rule Discretizer) de Holte [HOL93] et le découpage récursif minimisant l'entropie [FI93]. La première méthode est non-supervisée et les deux autres sont supervisées. La comparaison des méthodes est effectuée en appliquant deux classifieurs différents (C4.5 de [QUI93], et un classifieur bayésien : $\mathcal{MLC}++$ de [KJL+94]) à la suite de l'étape de discrétisation. Les classifieurs permettent de déterminer la classe d'appartenance de chaque objet d'un ensemble. En utilisant ces classifieurs, les auteurs peuvent donc évaluer la précision de la classification par rapport à celle obtenue directement par les classifieurs sans effectuer de discrétisation au préalable. Ils concluent que les trois méthodes donnent des résultats à peu près similaires, avec un léger avantage pour les méthodes supervisées. La meilleure performance moyenne est obtenue pour le découpage récursif minimisant l'entropie associé au classifieur bayésien.
La thèse de Ricco Rakotomalala [RAK97] comporte une partie sur la discrétisation des primitives continues. Dans ce chapître, il explique notamment combien il est difficile de comparer les méthodes globales et locales, étant donné qu'elles possèdent toutes deux des avantages et des inconvénients. La discrétisation locale, avec un partitionnement en un nombre fixe d'intervalles, peut notamment conduire à un fractionnement de données, et la construction de l'arbre sera donc plus en profondeur. Cependant, ce procédé à l'avantage de prendre en compte les intéractions entre les primitives. S'agissant de la discrétisation globale, elle induit la construction d'un arbre plus en largeur. De plus, elle permet une réduction de la dimension à la fois au niveau des primitives (horizontalement) mais aussi au niveau des individus (verticalement) ce qui améliore la rapidité de traitement. Les tests effectués par [QUI96] et [DKS95] ne permettent pas de trancher entre les deux stratégies. Il est nécessaire de savoir ce que l'on souhaite obtenir pour utiliser la méthode la plus en adéquation avec les attentes.
En conclusion, il paraît intéressant de préférer l'utilisation d'une méthode de discrétisation supervisée, statique (les plus utilisées), basée sur un critère de minimisation de l'entropie et enfin de choisir entre stratégie globale et locale en fonction de l'objectif souhaité.

3.3 Construction de l'arbre

Pour construire un arbre de décision, il est nécessaire de passer par deux phases successives : l'expansion de l'arbre, puis son élagage. L'étape d'expansion consiste à sélectionner les tests à associer aux différents nœuds de décision et ainsi fixer l'ordre dans lequel ils seront effectués. Il est également important de déterminer la position des feuilles et d'attribuer une classe à chacune d'elles. Cette étape est réalisée sur un ensemble d'apprentissage. Ensuite vient la phase d'élagage, qui est faite à partir d'un ensemble test. Elle permet de réduire la taille de l'arbre. Le principe d'élagage est le suivant : en partant des feuilles de l'arbre, on remonte vers la racine en remplaçant les nœuds de décision par des feuilles. On construit ainsi tous les arbres élagués candidats et pour chacun d'eux on calcule l'erreur de classification sur l'ensemble test. L'arbre retenu est celui qui minimise l'erreur.

Expansion

Il n'est pas envisageable de construire l'ensemble des arbres de décision pour ensuite sélectionner le meilleur, étant donné l'important nombre de combinaisons possibles. Il faut savoir que pour n primitives prenant en moyenne chacune m valeurs, il existe $\prod_{i=1}^{n} i^{m^{n-i}}$ arbres possibles. Par exemple, dans le cas de 4 primitives prenant chacune 3 valeurs, on peut obtenir 55 296 arbres différents. On cherche donc à construire l'arbre intelligemment en suivant une méthode descendante. En observant l'algorithme 1 de construction d'un arbre binaire, on peut constater que la phase d'expansion requiert la définition de trois opérateurs :
1. Sélectionner une primitive discriminante et donc un test à associer au nœud

2. Décider si un nœud est terminal
3. Affecter une classe à un nœud

Le premier opérateur nécessite l'utilisation d'une fonction caractérisant le degré de mélange des classes. En effet, pour savoir si une primitive est discriminante, il faut vérifier si elle sépare au mieux les objets de classes différentes, et donc il est obligatoire de pouvoir estimer dans quelle proportion les objets sont mélangés. Généralement, on utilise la fonction de Gini ou bien la fonction entropie. Toutes deux ont la particularité d'atteindre leur maximum lorsque les objets de classes différentes sont mélangés de manière homogène et valent zéro s'il n'y a aucun mélange. On cherche donc à trouver la primitive qui possède le plus grand pouvoir discriminant, c'est-à-dire celle qui minimise la fonction de Gini ou d'entropie.

À partir de ces fonctions, on peut introduire la notion de gain qui correspond à la réduction d'entropie attendue suite à la partition de N suivant une primitive V qui peut prendre c valeurs. Ce calcul du gain est réalisé pour l'ensemble des primitives. On choisira d'affecter au nœud de décision la primitive qui possède un gain maximal.

S'agissant du second opérateur : Décider si un nœud est terminal, il est possible d'utiliser différents critères. Le plus simple est de considérer que l'on crée une feuille losrqu'il ne reste que des objets appartenant à une même classe. Cependant, ce critère entraîne un partitionnement relativement conséquent des objets de l'ensemble d'apprentissage. Il existe d'autres critères permettant d'arrêter plus rapidement la construction de l'arbre tel que un seuil de profondeur maximal à ne pas dépasser, un seuil minimal d'objets, un faible gain d'information, ou encore un seuil de pureté du nœud en terme de classes au-delà duquel il faut interrompre le partitionnement des objets.

Enfin, pour le troisième opérateur, on affecte la classe majoritaire au nœud.

Nom: `ConstruireArbre(E, A)`
Donnees : E = un ensemble d'apprentissage, A = un arbre (vide au début)
début
  Si *le critère de terminaison de l'arbre est vérifié* **alors**
    Créer une feuille du nom de la classe dominante dans E ; Ajouter la feuille à A ;
  sinon
    Sélectionner la primitive la plus discriminante ;
    Créer un nœud comportant un test sur la primitive afin de diviser E en 2 sous-ensembles ;
    Ajouter le nœud à A ;
    ConstruireArbre(Egauche, A) ;
    ConstruireArbre(Edroit, A) ;
fin

Algorithme 1. Construction d'un arbre binaire récursivement.

Elagage

Il est réalisé sur l'arbre obtenu par l'algorithme de construction. Lors de la phase d'expansion, l'arbre est généralement développé jusqu'à obtenir des feuilles pures, il contient donc un nombre conséquent de nœuds de décision. Sa complexité doit nécessairement être limitée. La phase d'élagage a pour objectif la simplification de l'arbre qui se traduit par la réduction du nombre de nœuds de décision. L'élagage est effectué progressivement en remontant des feuilles vers la racine. À chaque itération, on sélectionne le nœud qui minimise un critère dans le but de le transformer en feuille. Le critère doit être choisi de manière à ce que le bénéfice de l'élagage soit un bon compromis entre la complexité de l'arbre et l'erreur de classification obtenue sur un ensemble test.

Parmi les critères existants, on peut citer : le Minimal Cost-Complexity Pruning (MCCP) ([BRE84]), le Minimum Error Pruning (MEP) ([NB86]), le Pessimistic Error Pruning (PEP) ([QUI87]) et le Error-Based Pruning (EBP) ([QUI93]).

Lorsqu'un nœud est sélectionné par le critère, il est transformé en feuille. On obtient alors un nouvel arbre qui est conservé pour être comparé par la suite à tous les autres arbres ainsi créés. L'arbre retenu est celui qui possède la plus petite erreur de classification.

Nom : `ElaguerArbre`($A_{\max}$)
Donnees : A_{max} = l'arbre binaire obtenu par l'algorithme d'expansion
Résultat : l'un des arbres binaires de l'ensemble ($A_{\max}$, A_1, …, A_n)
début : $k := 0$;
  $A_k := A_{\max}$;
  tant que A_k *possède plus d'un nœud* **faire**
    pour *chaque nœud n de* A_k **faire**
      Calculer le critère $c(A_k,n)$ sur l'ensemble d'apprentissage ;
    Choisir le nœud n_m pour lequel le critère est minimum ;
    A_{k+1} se déduit de A_k en y remplaçant n_m par une feuille ;
    $k := k + 1$;
  Choisir dans l'ensemble des arbres obtenus ($A_{\max}$, A_1, …, A_n) celui qui possède la plus petite erreur de classification sur l'ensemble test ;
fin

Algorithme 2. Élagage d'un arbre binaire.

3.4 Exemple de construction d'un arbre de décision

L'exemple suivant va permettre d'illustrer le processus de construction d'un arbre de décision. Nous sommes partis de la table 1 contenant cinq objets répartis selon deux classes. Les objets sont décrits par trois primitives b_1, b_2 et b_3, pour lesquelles les intervalles de discrétisation ont été créés. La formule de l'équation (1) représente le calcul de l'entropie d'un ensemble d'objets N, alors que la formule de l'équation (2) est l'entropie conditionnelle des sous-ensembles de N correspondants aux primitives de la table 1. Pour chaque primitive de la table, on calcule la valeur de l'entropie conditionnelle.

Table 1. Exemple.

	b_1	b_{21}	b_{22}	b_{23}	b_{31}	b_{32}	Classe	
Intervalles	[2-6]	[4-10]	[40-44]	[60-60]	[2-6]	[20-30]	C_1	C_2
a_1	X	X			X		X	
a_2	X	X			X		X	
a_3	X		X			X	X	
a_4	X		X		X			X
a_5	X			X		X		X

$$f(N) = -\sum_{i=1}^{nbClasses} \frac{n_i}{n} \log_2(\frac{n_i}{n}) \quad (1)$$

Avec N l'ensemble des objets testés au nœud courant, $nbClasses$ le nombre de classes, n le cardinal de N et n_i le nombre d'objets de N appartenant à la classe i.

$$f(C_1|b_.) = \sum_{j=1}^{nbIntervalles} \frac{n_{b_{.j}}}{n} \left(-\sum_{i=1}^{nbClasses} \frac{n_i}{n_{b_{.j}}} \log_2 \left(\frac{n_i}{n_{b_{.j}}} \right) \right) \quad (2)$$

Avec $nbIntervalles$ le nombre d'intervalles de discrétisation de la primitive $b_.$ (par exemple, 3 intervalles pour b_2 : b_{21}, b_{22} et b_{23}), $n_{b_{.j}}$ le nombre d'objets dans l'intervalle $b_{.j}$, n le nombre total d'objets, $nbClasses$ le nombre de classes et n_i le nombre d'objets de la classe i appartenant à $b_{.j}$.

$$f(C_1|b_1) = \frac{5}{5}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right) = 0{,}971 \quad (3)$$

$$f(C_1|b_2) = \frac{2}{5}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 0{,}4 \quad (4)$$

$$\begin{aligned} f(C_1|b_3) &= \frac{3}{5}\left(-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right) \\ &\quad + \frac{2}{5}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) \\ &= 0{,}951 \end{aligned} \quad (5)$$

D'après (3), (4) et (5), l'entropie est minimale pour b_2. La racine de l'arbre comportera donc un premier nœud de décision avec un test sur cette primitive. Les trois nœuds fils correspondent aux trois intervalles de la primitive b_2. Si $b_2 = b_{21}$, alors les objets sont de classe C_1, si $b_2 = b_{22}$ alors les objets sont de classe C_1 ou C_2 et enfin si $b_2 = b_{23}$, les objets sont de classe C_2. L'ambiguïté est donc située sur la primitive b_{22}, car elle ne permet pas de prendre de décision sur la classe des objets. Il faut donc créer un nouveau nœud de décision pour classer ces objets (a_3 et a_4). Lorsque b_{22} est validé, la table 1 se réduit à la table 2. On calcule à nouveau l'entropie sur les deux primitives restantes b_1 et b_3 (équations (6) et (7)).

$$f(C_1|b_1) = \frac{2}{2}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 1 \quad (6)$$

$$f(C_1|b_3) = 0 \quad (7)$$

Table 2. Exemple (suite)

	b_1	b_{31}	b_{32}	Classe	
Intervalles	[2-6]	[2-6]	[20-30]	C_1	C_2
a_3	X		X	X	
a_4	X	X			X

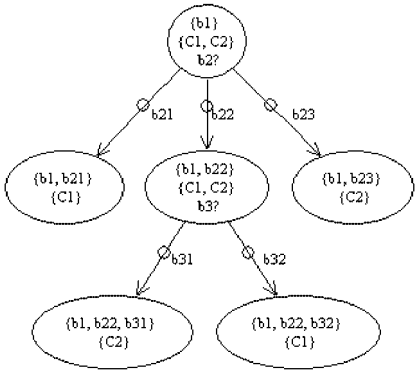


Figure 4. Arbre obtenu après construction basée sur l'entropie.

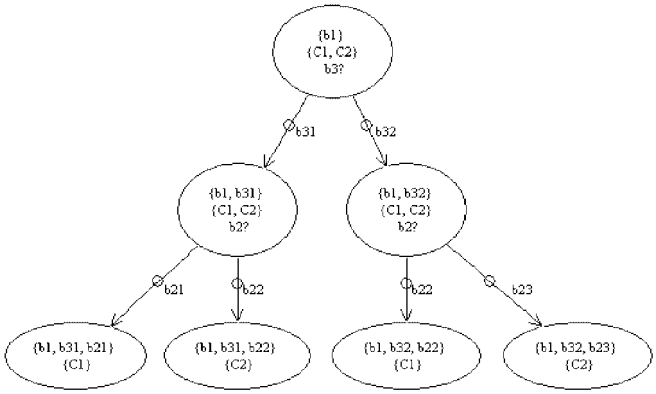


Figure 5. Arbre obtenu suivant un critère non-optimal.

C'est b_3 qui minimise l'entropie, donc le nouveau test portera sur cette primitive. On observe que la primitive b_1 n'apporte pas d'information pour le classement des objets. On peut donc valider cette primitive dès le premier test. On obtient alors l'arbre de la figure 4. En utilisant un autre critère, il est possible d'obtenir une construction d'arbre très différente (figure 5). Ici, nous avons choisi un critère non-optimal de l'entropie (sélection inverse des primitives par rapport à l'arbre précédent).

3.5. Conclusion

En général, les arbres de décision donnent de bons résultats dans la pratique. Cette structure est intéressante car elle est facilement compréhensible par l'utilisateur. De plus, elle permet une

traduction immédiate en règles de décision. Ces règles sont d'ailleurs mutuellement exclusives, c'est-à-dire qu'il n'est pas possible d'avoir deux feuilles différentes possédant les mêmes primitives de description.
Cependant, les arbres de décision possèdent quelques inconvénients. Tout d'abord, les méthodes de construction sont non optimales ; elles ne permettent pas de créer le meilleur arbre parmi tous les possibles. De plus, les choix dans la construction ne sont jamais remis en question (pas de backtracking) et l'ordre dans lequel les primitives sont étudiées est figé. L'utilisation d'un ensemble d'échantillonnage peut entraîner un problème de biais inductif. En effet, si l'ensemble d'échantillonnage n'est pas assez représentatif, il est très difficile d'estimer l'erreur commise sur la classification. Enfin, étant donné les nombreuses heuristiques à définir pour la construction de l'arbre, cet outil est difficile à paramétrer.

4. Treillis de Galois

Après avoir été l'objet de premiers travaux formels en théorie des graphes et des structures ordonnées [Bi67,BM70,DP91], le treillis de Galois, ou encore treillis des concepts a été introduit en analyse de données et classification [Wil82,GD86] où il a rapidement montré son utilité : la structure de treillis, basée sur la notion de *concept*, permet de décrire les données tout en conservant leur diversité, mais aussi leur complexité. L'*analyse formelle des concepts (AFC)* [GW99] a ainsi été introduite pour fournir un cadre théorique au treillis de Galois et à ses applications nombreuses. Une étude récente [MN04] recense quelques méthodes de classification basées sur un treillis de Galois, et apporte des éléments de comparaisons : les résultats y sont semblables voire meilleurs que ceux d'approches plus classiques.
Nous nous proposons dans une première partie de définir le treillis de Galois, puis de décrire l'utilisation qui peut en être faite en classification et sélection de primitives. Dans une deuxième partie, nous aborderons la problématique de sa génération, dont la complexité théorique est exponentielle dans le pire des cas, mais intéressante en pratique. Nous finirons par une illustration de cette approche avec un exemple, avant de conclure et d'apporter les éléments de comparaisons qui nous semblent important entre le treillis de Galois et l'arbre de décision.

4.1. Description et principe de sélection

Un treillis de Galois est composé d'un ensemble de concepts reliés par inclusion, et qui forment ainsi un graphe possédant les propriétés d'un treillis. Les concepts y sont définis à partir de données organisées sous la forme d'une table discrétisée appelée *contexte formel*.

Définition 1. (Contexte formel) *Un* contexte formel $C = (G, M, I)$ *est la donnée d'un ensemble* G *(*ensemble d'objets*), d'un ensemble* M *(ensemble de* primitives*), et d'une relation d'incidence* I *entre* G *et* M.

Définition 2. (Correspondance de Galois) *On associe à un ensemble* $A \subseteq G$ *l'ensemble* $f(A)$ *des primitives communes aux objets de* A *:*

$$f(A) = \{m \in M \mid (g,m) \in I \ \forall\, g \in A\} \qquad (8)$$

Duallement, pour $B \subseteq M$*, on definit l'ensemble* $g(B)$ *des objets communs aux primitives de* B *:*

$$g(B) = \{g \in G \mid (g,m) \in I \ \forall\, m \in B\} \qquad (9)$$

Ces deux fonctions f et g définies entre objets et primitives forment une correspondance de Galois.

Définition 3. (Concept formel) *Un* concept formel *du contexte* C *est un couple* (A,B) *avec* $A \subseteq G$, $B \subseteq M$, $f(A) = B$ *et* $g(B) = A$.

Définition 4. (Treillis des concepts, treillis de Galois) *Le* treillis de Galois *ou* treillis des concepts *d'un contexte formel* $C = (G, M, I)$ *est une paire* $(\beta(C), \leqslant)$ *où :*
- $\beta(C)$ *est l'ensemble de tous les concepts de* C.
- $\leqslant$ *est une relation d'ordre sur* $\beta(C)$ *définie, pour* (A_1, B_1) *et* (A_2, B_2) *deux concepts de* $\beta(C)$ *par :*

$$(A_1, B_1) \leqslant (A_2, B_2) \text{ ssi } A_2 \subseteq A_1 \text{ (équivalent à } B_1 \subseteq B_2) \qquad (10)$$

La relation $\leqslant$ *étant une relation d'ordre, on peut lui associer sa relation de couverture que l'on notera* $\prec$. $(\beta(C), \prec)$ *est alors le diagramme de Hasse*[1] *du treillis des concepts* $(\beta(C), \leqslant)$.

Considérons l'exemple représenté par la table1 présentant la répartition discrétisée de 5 objets selon deux classes, sans l'information relative aux classes donnée par les deux dernières colonnes. La figure 6 représente son treillis des concepts. On remarque sur cet exemple qu'il n'existe pas de concept contenant un ensemble vide de primitives car la primitive b_1 est partagée par tous les objets ($g(G) = \{b_1\}$). Elle appartient alors à tous les concepts.

Principe de sélection

Le treillis de Galois ainsi défini peut alors s'utiliser comme un espace de recherche permettant de sélectionner des primitives pour classifier un objet x : la signature du symbole dégradé à reconnaître est placée en entrée du graphe, puis nous progressons à l'intérieur du graphe en fonction des primitives validées jusqu'à atteindre un nœud correspondant à une classe. Une primitive est validée si la valeur de la signature du symbole est

1. Représentation d'une relation d'ordre sans ses relations de réflexivité et de transitivité.

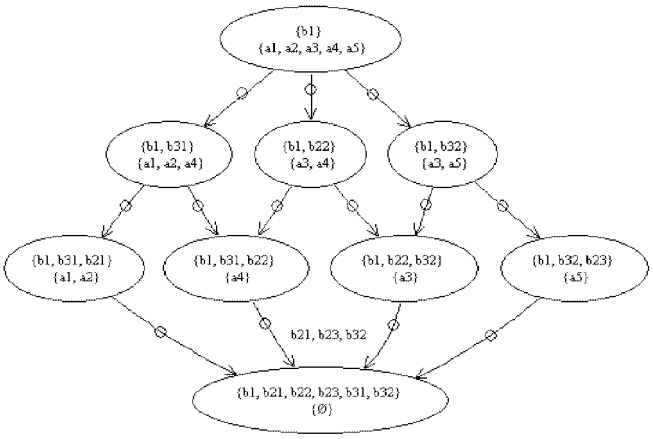


Figure 6. Le treillis de Galois du contexte de la table 1.

comprise dans l'intervalle correspondant à la primitive. Il est cependant important de noter que le treillis n'est définit que pour des données qui s'organisent en contexte, c'est-à-dire de données discrétisées. Un concept (A,B) d'un treillis de Galois correspond à l'ensemble maximal des objets A partageant les primitives B. On associe à un concept (A,B) une étape de classification où B est l'ensemble des primitives déjà validées pour x, et $c(A)$ l'ensemble des classes candidates pour classifier x (rappelons que $c(A)$ est l'ensemble des classes des objets de A). On retrouve ainsi avec un concept (A,B) les informations caractéristiques d'une étape élémentaire de classification définies pour un arbre de décision, à savoir :

- l'ensemble B des *primitives validées*, ensemble constitutif du concept,
- l'ensemble $c(A)$ des *classes candidates* à la classification et
- l'ensemble P des *primitives sélectionnées* et à tester qui se déduit des concepts correspondant aux étapes suivantes.

Par conséquent, classifier un objet x consiste en une succession d'étapes de classification jusqu'à une étape finale qui correspond à un concept (A,B) où $c(A)$ ne propose qu'une seule classe candidate. L'étape initiale correspond au plus petit concept $(G,g(G))$ (au sens de la relation $\leqslant$) où un ensemble $g(G)$, généralement vide, de primitives est validé, et toutes les classes $c(G)$ sont candidates. Une étape de classification élémentaire permet alors d'atteindre (A_1,B_1) à partir d'un concept $(A,B) \leqslant (A_1,B_1)$ par la validation des primitives $B_1 \setminus B$ pour x : l'ensemble des primitives validées augmente (car $B \subseteq B_1$) alors que l'ensemble des classes candidates décroit (car $A \supseteq A_1$ et par conséquent $c(A) \supseteq c(A_1)$).

La validation de primitives pour x nécessite l'utilisation d'un *critère de similarité*. On peut définir de nombreux critères de similarité, qu'ils soient basés sur des calculs de distances ou d'entropie. Une telle validation de primitives correspond bien à la notion de *décision* définie dans les approches basées sur un arbre de décision et décrite dans la partie précédente.

4.2 Construction du treillis de Galois

Discrétisation

Le treillis de Galois n'est défini que pour des données qui s'organisent en contexte, c'est-à-dire des données discrétisées. L'utilisation du treillis de Galois en classification supervisée nécessite donc une phase préalable de discrétisation des données, problématique complexe dont les différentes approches ont été présentées dans la section précédente. Une discrétisation globale est plus appropriée car les données sont discrétisées dans une phase de prétraitement. Notre problématique se situant dans le domaine de la classification supervisée, il s'agira donc d'une technique de discrétisation supervisée, c'est-à-dire introduisant l'indication de la classe d'un objet x par $c(x)$.

Construction

Le treillis de Galois a dans le pire des cas une taille exponentielle en la taille des données à classifier (c'est-à-dire du contexte initial) alors qu'il reste linéaire dans le meilleur des cas. Des études de complexité en moyenne sont extrêmement difficiles à mener, car la taille du treillis dépend à la fois de la taille des données à classifier, mais aussi de leur organisation et de leur diversité.

Une comparaison [KO01] a récemment été menée entre plusieurs algorithmes de génération du treillis et sur un même jeu de données. Parmi ces algorithmes citons les algorithmes de Bordat [Bor86], de Godin [GM93], ainsi que celui de Nourine et Raynaud [NR99] qui a la meilleure complexité théorique (complexité quadratique par éléments du treillis produit). Il y est constaté que les complexités théoriques des algorithmes testés sont de loin supérieures aux résultats observés en pratique. De récents travaux [GELY05] proposent un algorithme générique permettant à la fois d'unifier les algorithmes existants dans un même cadre, mais aussi de les comparer en fonction des propriétés des données, et par conséquent du treillis.

Réduction

Il est possible de définir des *critères de validité* permettant d'invalider certains concepts pour ne pas les générer et ainsi réduire la taille du treillis. On retrouve ainsi la notion d'élagage introduite pour un arbre de décision, mais aussi l'opérateur défini pour déterminer si un nœud est terminal ou non, les critères proposés peuvent également s'appliquer ici.

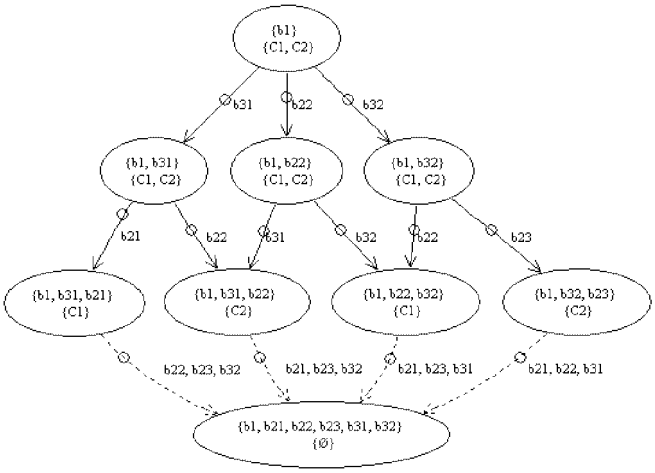
De tels critères permettent d'invalider des concepts jugés trop spécifiques et source d'erreurs, et qui sont généralement proches (au sens de la relation $\prec$) de l'élément maximal du treillis. Ces heuristiques génèrent une structure issue du treillis dont des concepts « maximaux » ont été supprimés (il s'agit généralement d'un inf-demi-treillis) souvent appelé Iceberg. Un premier critère de validité simple et immédiat consiste à valider les concepts

proposant une ou plusieurs classes candidates, et ainsi à invalider le seul concept maximal $(M,\emptyset)$ où $c(\emptyset)$ ne contient pas de classes candidates.

Il est également possible de réduire le nombre de concepts du treillis au cours du pré-traitement de discrétisation supervisée des données. Ainsi, sur l'exemple proposé dans la section précédente il apparaît que les objets a_1 et a_2 de la table 1 sont définis par un même ensemble de primitives. Dans le treillis associé, représenté par la figure 6, on remarque que a_1 et a_2 se retrouvent dans les mêmes concepts. En effet, il n'est pas nécessaire de différencier ces deux objets car ils appartiennent à la même classe : $c(a_1) = c(a_2) = C_1$. Avec une méthode de discrétisation non supervisée, ou ne tenant pas compte de l'indication de classe, ces deux objets auraient été différenciés via deux ensembles de primitives différents. Le treillis correspondant aurait alors été de taille supérieure.

4.3 Exemple de treillis de Galois

Reprenons l'exemple de la table 1 pour illustrer l'approche basée sur un treillis de Galois. La figure 6 représente le treillis de Galois défini pour cette table.



La figure 7 quant à elle représente ce même treillis où, pour chaque concept (A,B), l'ensemble des objets A est remplacé par l'indication de classe $c(A)$, information nécessaire à chaque étape de classification. Les primitives sélectionnées sont quant à elles indiquées sur chaque arc du treillis. Les arcs entrants du concept minimal sont quant à eux représentés en pointillés afin d'indiquer la possibilité de réduire ce treillis en un inf-demi-treillis par suppression du concept maximal qui ne contient aucune indication de classe.

Ces quelques modifications permettent d'harmoniser la structure de treillis avec celle de l'arbre de décision. On retrouve ainsi pour chacune de ces deux structures les trois informations nécessaires à une étape élémentaire de classification : l'ensemble des primitives sélectionnées, l'ensemble de primitives validées et l'ensemble des classes candidates.

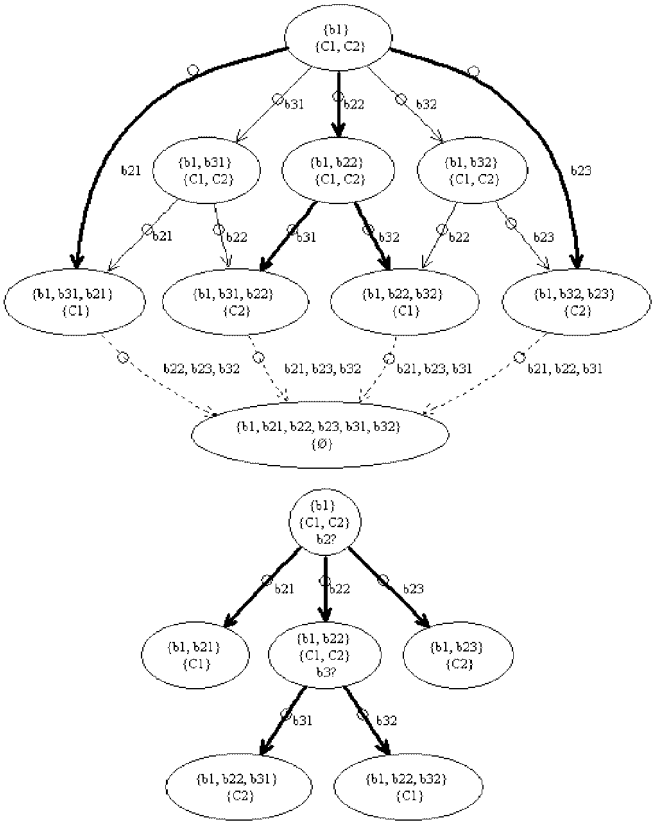


Figure 8. Comparaison du treillis et de l'arbre de décision construit sur un critère « optimal ».

Les figures 8 et 9 reprennent quant à elles les deux arbres de décisions présentés dans la section précédente, l'un optimal et l'autre non optimal, selon un critère d'entropie. Il y apparaît clairement que le treillis est une structure qui apporte la même lisibilité que l'arbre mais de l'information supplémentaire (l'information de ces deux arbres de décisions est présente dans le treillis).

4.4. Conclusion de l'approche avec un treillis de Galois

L'utilisation d'un treillis pour sélectionner des primitives au cours d'un processus de classification supervisée est extrêmement proche de celle d'un arbre de décision : en effet, on associe facilement à ces deux structures un traitement de classification défini par une séquence d'étapes élémentaires que l'on retrouve à l'identique dans les deux approches. Le treillis conserve donc l'intérêt de lisibilité apporté par les arbres de décision.

Une différence entre ces deux structures se situe cependant au niveau de leur taille : l'arbre de décision même s'il n'est pas toujours optimal, est plus petit que le treillis dont la taille est exponentielle dans le pire des cas. On remarque même sur des exemples que l'arbre de décision est inclus dans le treillis pour un même jeu de données discrétisées. Ceci implique que la construction du treillis coûte cher : de complexité théorique exponentielle dans le pire des cas, bien qu'elle soit jugée inté-

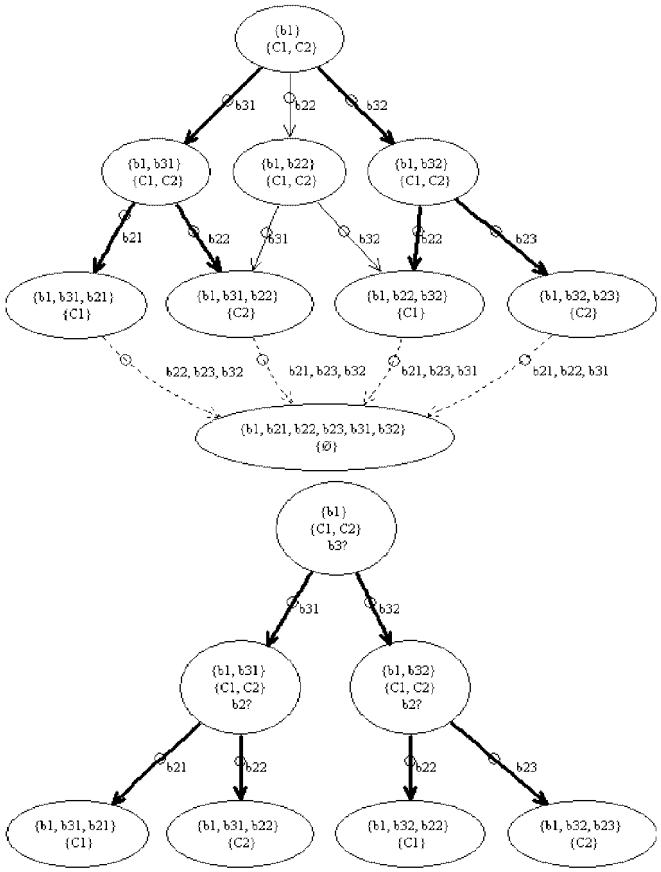


Figure 9. Comparaison du treillis et de l'arbre de décision construit sur un critère « non optimal ».

ressante et abordable par plusieurs études expérimentales. Cette différence de taille vient essentiellement du fait que le treillis propose un grand nombre d'étapes élémentaires de classification, et de séquences de sélection de primitives qui se regroupent. On peut cependant noter que c'est une structure qui s'adapte parfaitement à des données bruitées ou détériorées pour lesquelles des décisions ne peuvent pas être prises, ou risquent d'être invalides. Il est cependant important de remarquer que le treillis est défini formellement et de façon unique à partir d'un contexte, et sa construction ne nécessite aucun paramétrage. Contrairement à l'arbre de décision qui, défini à partir de nombreuses heuristiques, reste difficile à paramétrer, et par conséquent n'est pas unique.

Nous terminerons cette section de description du treillis de Galois par rapporter les résultats d'une étude récente portant sur l'utilisation du treillis de Galois en classification [MNN05] : les résultats faits sur plusieurs jeux de données sont semblables voire meilleurs à ceux obtenus par des techniques plus classiques de classification. Plusieurs méthodes basées sur un treillis y sont décrites et comparées. Ces méthodes varient selon plusieurs critères dont l'étape de discrétisation, mais aussi le treillis obtenu (souvent un treillis réduit à partir d'un critère de validité des concepts), ou encore les algorithmes de génération utilisés. Le principal inconvénient de cette approche reste le temps de calcul inhérent à la construction du treillis.

5. Expérimentation

Nous tenons à préciser que cette expérimentation a été réalisée dans le but de comparer l'arbre de décision et le treillis de Galois, et non pour obtenir de bons résultats en terme de reconnaissance. Nous avons utilisé les symboles de GREC 2003 disponibles sur le site [GREC03]. Dans un souci de simplification, nous avons travaillé sur 10 symboles constituant les 10 classes à reconnaître. Pour les caractériser, une signature contenant 33 invariants de Fourier-Mellin a été calculée. Pour chaque modèle, nous disposions sur le site de 90 symboles bruités, répartis par groupe de 10 suivant une échelle de dégradation allant de 1 à 9. La méthode de génération du bruit pour des images binaires a été réalisée par Kanungo *et al.* [KHB+94]. Le bruit doit ressembler aux dégradations obtenues par des opérations d'impression, de photocopie ou de numérisation. Le principe de la méthode est d'inverser la couleur des pixels noirs et blancs en considérant pour chaque pixel candidat la distance qui le sépare de la région la plus proche de couleur inverse.

Nous avons utilisé deux méthodes de discrétisation : discrétisation par entropie et par la distance maximale. La première a été décrite précédemment. La seconde est une méthode non supervisée qui consiste à rechercher la primitive qui possède l'écart maximal entre deux valeurs consécutives (v_i et v_{i+1}) lorsque les valeurs de l'intervalle sont classées par ordre croissant. On discrétise alors les valeurs de la primitive sélectionnée en coupant l'intervalle en deux au niveau de l'écart maximal. Le premier nouvel intervalle ainsi créé aura pour borne maximale v_i et le second aura pour borne minimale v_{i+1}. Cette méthode a été choisie pour sa simplicité.

Les 2 tables discrétisées obtenues (tables 3 et 4) nous ont alors chacune permis de construire l'arbre de décision et le treillis de Galois. Nous présentons seulement les graphes obtenus avec la table discrétisée par l'entropie (fig. 10 et 11). À partir des graphes construits, nous avons calculé la signature (33 invariants de Fourier-Mellin) des symboles dégradés sans procéder à aucun prétraitement pour supprimer le bruit. Pour la reconnaissance, nous avons procédé de la manière suivante : en entrée du graphe nous avons placé la signature du symbole dégradé à reconnaître, puis nous avons progressé à l'intérieur du graphe en fonction des primitives validées jusqu'à atteindre un nœud correspondant à une classe. Une primitive est validée si la valeur de la signature du symbole est comprise dans l'intervalle correspondant à la primitive. Dans le tableau 5, nous présentons le nombre de symboles dégradés bien classés sur les 900 symboles testés. Les tables 6 et 7 sont les matrices de confusion obtenues pour l'arbre de décision et le treillis de Galois en cumulant les résultats de reconnaissance des différents niveaux de dégradation.

Dans cette expérimentation, nous avons voulu mettre en place une comparaison entre l'arbre de décision et le treillis de Galois.

Tableau 3. Table discrétisée par l'entropie et simplifiée (suppression des colonnes entièrement remplies) pour les 10 symboles traités.

Intervalles d'origine	0	1	2	3	4	5	6
Intervalles pertinents	0 33	1 34	2 35	3 36	4 37	5 38	6 39
1	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X
3	X	X	X	X	X X	X	X
4	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X

Tableau 4. Table discrétisée par distance maximale et simplifiée (suppression des colonnes entièrement remplies) pour les 10 symboles traités.

Int. d'origine	0	1	2	8	9	12	17	18	23	27	28
Int. pertinents	0 34 40	1 33 37	2 43	8 36 41	9 35 39 42	12 48	17 45	18 38	23 46	27 44	28 47
1	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X	X	X	X

Tableau 5. Nombre de symboles dégradés bien classés par l'arbre de décision et le treillis de Galois sur les 900 symboles testés.

	Discrétisation entropie	Discrétisation distance maximale
Arbre de décision	369 (41 %)	408 (45 %)
Treillis de Galois	401 (45 %)	401 (45 %)

Nous avons pour cela utilisé la même structure de base (la table discrétisée) pour construire les deux graphes, ainsi que le même dispositif de reconnaissance. D'après le tableau 5, les deux structures obtiennent des résultats comparables en terme de reconnaissance, et le treillis de Galois est légèrement plus efficace lorsque la table est discrétisée par l'entropie. En observant les matrices de confusion 6 et 7, on constate que l'arbre de décision et le treillis de Galois classent correctement et incorrectement à peu près les mêmes symboles. Ainsi les trois symboles les mieux reconnus sont identiques avec les deux méthodes que ce soit pour la discrétisation par l'entropie (symboles 7, 6 et 2)

Figure 10. Arbre de décision construit à partir de la table discrétisée par l'entropie (table 3).

Tableau 6. Matrices de confusion de l'arbre de décision (à gauche) et du treillis de Galois (à droite) avec la table discrétisée par l'entropie.

	1	2	3	4	5	6	7	8	9	10	Somme
1	14	6	3	16	13	1		10		10	73
2	46	58	19	7	35	7	19				191
3	10		46	30		14				10	110
4				15					41	20	76
5		5			15	1		20		20	61
6	20	20	14	18	20	62		10	10		174
7		1	7		7	5	71		30		121
8			1					50			51
9									8		8
10				4					1	30	35

	1	2	3	4	5	6	7	8	9	10	Somme
1	39			2						12	59
2	21	61	42	27	43	17	15	10	7	10	250
3		1	12	33	2	5			5		57
4	10			24					6	1	41
5		4			27	4		18	2	18	73
6		19	21		12	55		10	32		149
7			12		1	5	75	2	26		121
8	20	5						50		2	74
9						3			11		14
10			3	4	5	1			1	47	62

Tableau 7. Matrices de confusion de l'arbre de décision (à gauche) et du treillis de Galois (à droite) avec la table discrétisée par la distance maximale.

	1	2	3	4	5	6	7	8	9	10	Somme
1	60	20			30		10		10	31	161
2		69	1	9	10	6					95
3			45	2		59				11	117
4			25	68		5			30		128
5					28					20	48
6						0					0
7	30	1	19	11	22	20	80	90	20		293
8								0			0
9									30		30
10										28	28

	1	2	3	4	5	6	7	8	9	10	Somme
1	50	18			5		10				83
2		55	6	9	12	18					100
3			36	1		20					57
4		2	29	69	23	46	27	19	51	10	276
5					32					10	42
6						6					6
7	40	14	19	11	3		53	51	9	10	210
8		1						20		10	31
9									30		30
10					15					50	65

ou celle par la distance maximale (symboles 7, 2 et 4). De plus, les erreurs de classement sont souvent faites en choisissant les mêmes symboles. Par exemple, avec la discrétisation par l'entropie, l'arbre de décision et le treillis de Galois choisissent souvent les symboles 2, 6 et 7 et avec la discrétisation par la distance maximale, ce sont les symboles 7 et 4. On peut constater que les symboles les mieux reconnus sont ceux qui sont les plus souvent choisis par les deux graphes.

En terme de complexité, la génération du treillis de Galois est bien évidemment plus coûteuse que celle de l'arbre de décision, mais cependant la reconnaissance est équivalente. Nous avons évalué la complexité de chacune de ces méthodes en fonction des paramètres suivants :

- p : le nombre de symboles modèles ($p = 10$)
- c : le nombre de classes avec $c \leq p$ (ici $c = p$)
- n : le nombre de symboles à reconnaître ($n = 900$)
- l : la longueur de la signature ($l = 33$ invariants de Fourier-Mellin)

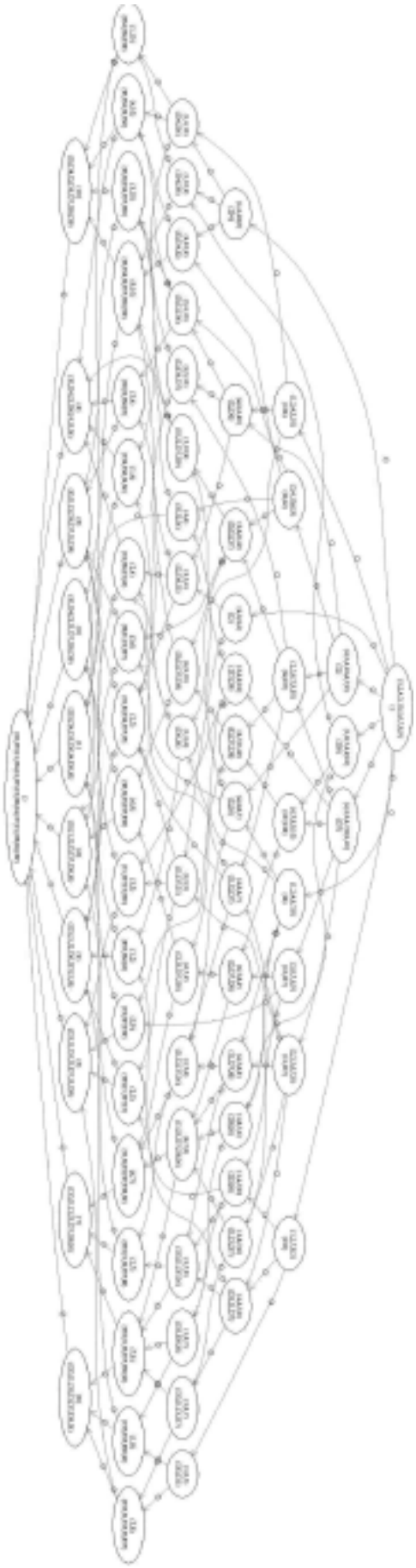


Figure 11. Treillis de Galois construits à partir de la table discrétisée par l'entropie (table 3).

- m : le nombre d'intervalles dans la table discrétisée avec $l < m$ (m = 40 pour la discrétisation par l'entropie et m = 49 pour la discrétisation par la distance maximale)
- m' : le nombre d'intervalles pertinents, c'est-à-dire le nombre d'intervalles de la table qui ne sont pas entièrement remplis de croix avec $m' \leq m$ (m' = 14 pour la discrétisation par l'entropie et m' = 27 pour la discrétisation par la distance maximale)

Il est possible de représenter la complexité de chacune des méthodes sous la forme générique suivante : $O(P + nR)$, avec P la complexité du prétraitement et R la complexité de la reconnaissance d'un symbole. Les complexités des méthodes sont présentées dans le tableau 8.

Tableau 8. Complexité des différentes méthodes.

Méthodes	P	R
Arbre de décision	Table : $O(pm')$ + Arbre : $O(pm')$	$O(cm')$
Treillis de Galois	Table : $O(pm')$ + Treillis : $O(2^{c+m'})$	$O(cm')$

De plus, le processus de reconnaissance de ces deux méthodes est moins coûteux qu'une recherche exhaustive parmi toutes les signatures des symboles modèles, ce qui constitue un point important lorsque l'on doit traiter des bases contenant plusieurs milliers de symboles. D'autre part, il est envisageable de générer le treillis de Galois à la demande grâce à l'utilisation de règles [GD86,BN04}, la complexité de sa construction en serait ainsi diminuée.

Pour aller plus loin dans la comparaison, nous avons testé la reconnaissance des symboles en réduisant la taille des signatures. Les graphiques de la figure 12 présentent l'évolution du nombre de symboles classés correctement par les méthodes lorsque la signature (les invariants de Fourier-Mellin) varie de 3 à 33 valeurs. Avec une discrétisation par la distance maximale, on peut remarquer que le treillis donne les meilleurs résultats quand la signature est inférieure à 21 invariants. Pour une signature plus grande, c'est l'arbre de décision. Il est aussi intéressant de constater que le maximum de reconnaissance n'est pas forcément obtenu lorsque la signature est la plus grande. En effet, le treillis de Galois atteint un maximum de 56 % de symboles correctement classés pour une signature de 12 invariants. Certaines primitives sont donc porteuses de plus d'information pertinente que les autres. D'autre part, on peut observer que les résultats de reconnaissance de l'arbre de décision sont disposés en palier. Nous avons obtenus ces résultats équivalents car les arbres de décision générés étaient identiques. La construction de l'arbre de décision est modifiée seulement lorsqu'un invariant porteur d'une information prépondérante est introduit dans la signature, alors que pour le treillis tous les invariants correspondants à des intervalles pertinents modifient sa contruction. Ainsi, on peut

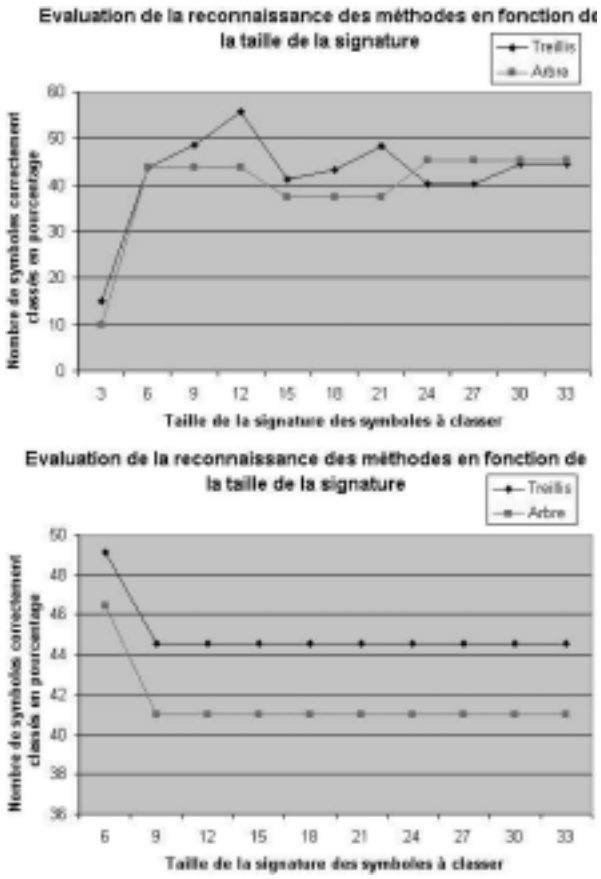


Figure 12. Évaluation de la reconnaissance des méthodes en fonction de la taille de la signature. Tests réalisés sur 900 symboles avec une discrétisation par la distance maximale (en haut) et par l'entropie (en bas).

observer qu'avec une discrétisation par l'entropie, les 7 premiers invariants de la signature sont porteurs de la totalité de l'information, car l'ajout des autres invariants de la signature ne modifie pas la construction des graphes et donc n'améliore pas la reconnaissance.

Cette expérience révèle l'importance d'une étude préliminaire sur les données issues du contexte afin d'utiliser les primitives les plus pertinentes pour la construction des graphes.

7. Conclusion et perspectives

Dans ce papier, nous avons décrit la problématique de la reconnaissance d'images détériorées ainsi que les différentes approches proposées. Parmis ces approches, nous nous sommes intéressés à celles basées sur une étape de sélection de primitives au sein d'un traitement de classification supervisée après segmentation et analyse statistique d'images documentaires. Nous avons ainsi présenté en détail une telle approche utilisant un arbre de décision,

afin de l'harmoniser avec une approche moins étudiée utilisant un treillis de Galois. Cette harmonisation permet ainsi de souligner le fait que ces deux structures proposent les mêmes étapes élémentaires de classification supervisée. Il apparaît cependant que l'arbre de décision, de taille plus petite que le treillis, permet d'optimiser le traitement, mais peut aussi entrainer des erreurs de classification dues au bruit engendré au cours des différentes étapes du traitement global (segmentation, puis descripteur statistique, discrétisation, sélection de primitives et enfin classification). L'approche avec un treillis propose un plus grand nombre de séquences de classification et semble plus appropriée pour des images détériorées, mais au détriment de sa grande taille. D'où un coût théorique exponentiel de l'approche par treillis, mais intéressant de par les quelques expérimentations qui en ont déjà été faites. L'étude expérimentale en cours nous encourage dans cette direction, à savoir une étude comparative entre les structures d'arbre et de treillis pour la sélection de primitives.

Nous avons montré dans ce papier que pour sélectionner des primitives au cours d'un processus de classification, une approche basée sur un treillis de Galois est plus adaptée à des données déteriorées qu'à des données supposées fiables, et pour lesquelles une méthode basée sur un arbre de décision est nettement suffisante. C'est la raison pour laquelle il nous semble important d'orienter nos travaux vers une étude de la robustesse au bruit, problématique qui s'adapte tout à fait à des données issues des descripteurs statistiques d'images documentaires.

En terme de perspectives sur ces phases expérimentales, nous envisageons l'exploitation des treillis pour intégrer des descriptions statistico-structurelles des formes, apportant ainsi une réponse au cloisonnement fort entre description structurelle et statistique.

Il nous apparaît également important de travailler sur la discrétisation supervisée des données afin d'adapter cette étape essentielle à la fois au treillis, mais aussi aux données dont on connait le mode de génération par descripteurs statistiques.

Il nous semble également intéressant d'exploiter les propriétés structurelles du treillis afin de limiter le coût de sa construction, notamment l'utilisation d'une représentation canonique et non exponentielle d'un treillis par un système de règles [GD86,BN04] qui permettrait de générer le treillis à la demande, c'est-à-dire de générer les étapes de sélection seulement si cela est nécessaire lors de la reconnaissance.

Références

[AOC+99] S. ADAM, J.M. OGIER, C. CARIOU, R. MULLOT, J. GARDES and Y. LECOURTIER, Multi-scaled and multi oriented character recognition : An original strategy, *ICDAR'99*, pages 45-48, Septembre 1999.

[AOC+01] S. ADAM, J.M. OGIER, C. CARIOU, R. MULLOT, J. GARDES and Y. LECOURTIER, Utilisation de la transformée de Fourier-Mellin pour la reconnaissance de forme multi-orienté et multi-échelle : application l'analyse automatique de documents tecniques. *Traitement du Signal*, 18(1):17-33, 2001.

[AS98] C. AH SOON, *Analyse de plans architecturaux*. PhD thesis, Institut National Polytechnique de Lorraine, 1998.

[BDF86] R. BAMIEH and R. DE FIGUEIREDO, A general moments invariants/attributed graph method for the three dimensional object recognition from a single image. *IEEE Journal of Robotics Automation*, 2 : 240-242, 1986.

[BFOS84] L. BREIMAN, J.H. FRIEDMAN, R.A. OLSHEN and C.J. STONE, *Classification and regtession trees*. Wadsworth Inc., Belmont, California, 1984.

[BIR67] G. BIRKHOFF, *Lattice theory*, volume 25. American Mathematical Society, 3rd edition, 1967.

[BK99] E. BAUER, R. KOHAVI, An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. Machine Learning, 36((1-2) : 105-139, 1999.

[BM] M. BARBUT, B. MONJARDET, *Ordre et classification, Algèbre et combinatoire*.

[BN04] K. BERTET, M. NEBUT, Efficient algorithms on the family associated to an implicationnal system. *Discrete Mathematical and Theoretical Computer Science (DMTCS)*, 6(2) : 315-338, 2004.

[BOR86] J.P. BORDAT, Calcul pratique du treillis de Galois d'une correspondance. *Math. Sci. Hum.*, 96 : 31-47, 1986.

[BSA91] S.O. BELKASIM, M. SHRIDAR and M. AHMADI, Pattern recognition with moment invariants : a comparative study and new results. *Pattern Recognition*, 24 : 1117-1138, 1991.

[CF04] D.R. CARVALHO and A.A. FREITAS, A hybrid decision tree/genetic algorithm method for data mining. *Information Science*, 163(1-3) : 13-35, June 2004.

[CLD96] Y. CHEN, N.A. LANGRANA, and A.K. DAS, Perfecting vectorized mechanical drawings. Computer Vision and Image understanding, 63(2) : 273-286, 1986.

[DBM77] S.A. DUDANI, K.J. BREDDING and R.M. MCGHEE, Aircraft identification by moment invariants. *IEEE Trans on Computers*, 26 : 39-45, 1977.

[DBN92] M. DAI, P. BAYLOU, and M. NAJIM, An efficient algorithm for computation of shape moments from run-length codes or chain codes. *Pattern Recognition*, 25 : 1119-1128, 1992.

[DKS95] J. DOUGHERTY, R. KOHAVI and M. SAHAMI, *Supervised and unsupervised discretization of continuous features*. Morgan Kaufman, 1995.

[DP91] B.A. DAVEY and H.A. PRIESTLEY, *Introduction to lattices and orders*. Cambridge University Press, 2nd edition, 1991.

[FF92] A.J. FILIPSKI and R. FLANDRENA, Automated conversion of engineering drawings to cad form. *Proc. IEEE*, 80(7) : 1195-1209, 1992.

[FI93] U.M. FAYYAD and K.B. IRANI, *Multi-interval discretization of continuous-valued attributes for classification learning*. Morgan Kaufman, 1993.

[FK] L.A. FLETCHER and R. KASTURI, A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. on PAMI*, 10(6) : 910-918, 1998.

[FOTK92] M. FUKUMI, S. OMATU, T. TAKEDA and T. KOSAKA, Rotation invariant neural pattern recognition system with application to coin recognition. *IEEE Trans. on Neural Networks*, 3 : 272-279, 1992.

[GD86] J.L. GUIGUES and V. DUQUENNE, Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 95 : 5-18, 1986.

[GEL05] A. GELY, A generic algorithm for generating closed sets of binary relation. Formal Concept Analysis, *Third Int. Conf., ICFCA 2005*, pages 223-234, February 2005.

[GM93] R. GODIN and H. MILI, Building and maintening analysis-level class hierarchies using Galois lattices. *OOPSLA*, pages 394-410, 1993.

[GML] V. GUNES, M. MENARD and P. LOONIS, A multiple classifier system using ambiguity rejection for clustering-classification cooperation. *IJUFKS, Worl Scientific*.

[GOL89] D.E. GOLDBERG, Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company, 1989.

[GRE03] GREC, www.cvc.uab.es/grec2003/symreccontest/index.htm, 2003.

[GW99] B. GANTER and R. WILLE, *Format concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 1999.

[HOL93] R.C. HOLTE, Very simple classification rules perform well on most commonly used datasets. *Machines Learning*, 11 : 63-90, 1993.

[HU62] M.K. HU, Visual pattern recognition by moment invariants. *IRE Trans. On Information Theory*, 8 : 179-187, 1962.

[KG82] F.P. KUHL and C.R. GIARDINA, Elliptic Fourier features of closed contour. *Computer Vision, Graphics and Image Processing*, 18 : 236-258, 1982.

[KH90a] A. KHOTANZAD and Y.H. HONG, Invariant image recognition by zernike moments. *IEEE Trans. on PAMI*, 12(5) : 489-497, 1990.

[KH90b] A. KHOTANZAD and Y.H. HONG, Invariant image recognition using features selected via a systemaic method. *Pattern Recognition*, 23 : 1089-1101, 1990.

[KHB+94] T. KANUNGO, R.M. HARALICK, H.S. BAIRD, W. STUETZLE and D. MADIGAN, Document degradation models : parameter estimation and model validation. In *IAPR Workshop on machine vision applications*, Kawasaki (Japan), pages 552-557, 1994.

[KIT92] N. KITA, Object locating based on concentric circular description. *Proc. 11th IEEE International Conference of Pattern Recognition, The Hague*, 1 : 637-641, 1992.

[KJL+94] R. KOHAVI, G. JOHN, R. LONG, D. MANLEY and K. PFLEGER. *MLC++ : A machine learning library in C++*. IEEE Computer Society Press, 1994.

[KO01] S. KUZNATSOV, S. OBIEDKOV, Comparing performance of algorithms for generating concept lattices. In *Proceedings of ICCS'01 workshop on Concept Lattices for Knowledge Discovery in Databases (CLKDD)*, volume 42, pages 35-47, July 2001.

[LCD97] N.A. LANGRANA, Y. CHEN and A.K. DAS, Feature identification from vectorized mechanical drawings. *Computer Vision and Image Understanding*, 68(2) : 127-145, 1997.

[LEF99] L. LEFRERE, *Contribution au Développement d'Outils pour l'Analyse Automatique de Documents Cartographiques*. PhD thesis, Université de Rouen, 1999.

[LIN87] C.H. LIN, New forms of shape invariants from elliptic Fourier descriptors. *Pattern Recognition*, 20 : 535-545, 1987.

[LK94] C.P. LAI and R. KASTURI, Detection of dimension sets in engineering drawings. *IEEE Trans. on PAMI*, 16(8) : 848-855, 1994.

[LP98] S.X. LIAO and M. PAWLAK. On the accuracy of zernike moments for image analysis. *IEEE Trans. on PAMI*, 20(12)) : 1358-1364, 1998.

[LS91] B.C. LIN and J. SHEN, Fast computation of moment invariants. *Pattern Recognition*, 24) : 807-813, 1991.

[LU98] Z. LU, Detection of text regions from digital engineering drawings. *IEEE Trans. on PAMI*, 20(4)) : 431-439, 1998.

[LVSM01] J. LLADOS, E. VALVENY, G. SANCHEZ and E. MARTI, Symbol recognition, current advances and perceptives. *Les actes de IAPR International Workshop on Graphics RECgnition (GREC)*, Kingston, Canada, pages 109-129, 2001.

[MNN05] E. MEPHU NGUIFO, P. NJIWOUA, Treillis des concepts et classification supervisée. In *Technique et Science Informatiques (à paraître), RSTI*. Herm-Lavoisier, Paris, France, 2005.

[NB86] T. NIBLETT, I. BRATKO, *Learning decision rules in noisy domains*. Cambridge University Press, 1996.

[NR99] L. NOURINE, O. RAYNAUD, A fast algorithm for building lattices. In *Third International Conference on Orders*, Algorithms and Applications, Montpellier, august 1999.

[PL92] S.C. PEI, C.N. LIN, Normalisation of rotationally symmetric shapes for pattern recognition. *Pattern Recognition*, 25 : 913-920, 1992.

[PN93] B. PASTERNAK and B. NEUMANN, Adik : An adaptable drawing interpretation Kernel. *Les actes de International Joint Conference on Artificial Intelligence (IJCAI), Avignon*, 1 : 531-540, 1993.

[QUI87] J.R. QUILAN, Simplifying decision trees. *International Journal of ManMachine Studies*, 27 : 221-234, 1987.

[QUI93] J.R. QUILAN. *C4.5 : Programs for Machine Learning*. Morgan Kaufman, Los Altos, California, 1993.

[QUI96] J.R. QUINLAN, *Bagging, boosting and C4.5*. AAAI Press, Menlo Park, CA, 1996.

[RAK97] R. RAKOTOMALALA, *Graphes d'induction*. PhD thesis, Université Claude Bernard, Lyon I, Décembre 1997.

[RSV96] I. ROTHE, H. SUSSE and K. VOSS, The method of normalization to determine invariants. *IEEE Trans. on PAMI*, 18(4) : 366-379, 1996.

[SEM04] D. SEMANI, *Sélection de variables pour la caractérisation d'objets déformables en mouvement – Application à la reconnaissance temps réels de Poissons dans des séquences vidéos, Projet Aq@thèque*. PhD thesis, Université de La Rochelle, 2004.

[SHA+92] S. SHIMOTSUJI, O. HORI, M. ASANO, K. SUSUKI, F. HOHINO and T. ISHII, A robust recognition system for a drawing superimposed on a map. *IEEE Computer magazine*, 25(7) : 56-64, 1992.

[TC88] C. TEH and R. CHIN, On image analysis by the method of moments. *IEEE Trans. on PAMI*, 10 : 496-512, 1988.

[TEA80] M. TEAGUE, Image analysis via the general theory of moments. *Journal of Optical Society of America*, 70:920-930, 1980.

[TJT96] O.D. TRIER, A.K. JAIN and T. TAXT, Features extraction methods for character recognition – a survey. *Pattern Recognition*, 29 : 641-662, 1996.

[TOD90] T. TAXT, J.B. OLAFSDOTTIR, M. DAEHLEN, Recognition of hand-written symbols. *Pattern Recognition*, 23 : 1155-1166, 1990.

[TOM96] K. TOMBRE, Quelques contributions à l'interprétation de documents techniques, habilitation à diriger des recherches, 1996.

[TT97] O.D. TRIER, T. TAXT and A.K. JAIN, Recognition of digits in hydrographic maps : binary versus topographic analysis. *IEEE Trans. on PAMI*, 19(4) : 399-404, 1997.

[WIL82] R. WILLE, Restructuring lattice theory : an approach based on hierarchy on contexts. *Ordered sets*, pages 445-470, 1982.

[ZJ96] D. ZONGER and A. JAIN, Algorithms for feature selection : An evluation. *Proceedings if ICPR 96*, 2 :18-22, 1996.

Jean-Marc **Ogier**

Jean-Marc Ogier est professeur au sein du laboratoire L3I de l'université de La Rochelle. Les thématiques de recherche abordées au cours de ces 4 dernières années concernent principalement :

- le développement d'outils bas niveau pour l'interprétation des images et des documents,
- La représentation et l'acquisition des différentes catégories de connaissances impliquées dans des dispositifs d'interprétation,
- Le développement d'architectures logicielles génériques pour la compréhension des signaux et des images,
- L'indexation de documents composites, la recherche de signature pertinentes pour l'indexation dans des document de niveau de structuration variable.

Stéphanie **Guillas**

Stéphanie Guillas effectue sa thèse au laboratoire L3I de l'université de La Rochelle. La problématique qui l'intéresse concerne la reconnaissance d'objets détériorés ou les objets sont des images, plus particulièrement des images de symboles issus de documents numériques. Il s'agit d'étudier la faisabilité d'utiliser une structure de treillis de Galois pour guider une telle reconnaissance.

Karell **Bertet**

Karell Bertet est maître de conférences, membre du laboratoire L3I de l'université de La Rochelle. Ses travaux de recheche concernent les systèmes implicatifs, ou systèmes de règles qui offrent un cadre méthodologique complet et robuste pour exprimer et traiter efficacement des liens entre des données sous forme de règles. La pertinence d'un tel cadre méthodologique provient essentiellement du lien formel entre les systèmes implicatifs et la théorie des treillis. Au cours de ces 4 dernières années, ses contributions sont :

- orientées modèle (étude algorithmique des systèmes implicatifs)
- orientées modélisation (aide à la reconnaissance d'images détériorées à l'aide d'un treillis de Galois)