MATHÉMATIQUES ET SCIENCES HUMAINES

ENGELBERT MEPHU NGUIFO

Une nouvelle approche basée sur le treillis de Galois, pour l'apprentissage de concepts

Mathématiques et sciences humaines, tome 124 (1993), p. 19-38

<http://www.numdam.org/item?id=MSH_1993__124__19_0>

Math. Inf. Sci. hum., (31e année, n°124, *1993*, pp.19-38)

UNE NOUVELLE APPROCHE BASÉE SUR LE TREILLIS DE GALOIS, POUR L'APPRENTISSAGE DE CONCEPTS

Engelbert MEPHU NGUIFO[1]

RÉSUMÉ — *L'apprentissage automatique à partir d'exemples consiste généralement à caractériser un ensemble d'objets dénotant un concept. Nous avons développé deux méthodes d'apprentissage symbolique, LEGAL et LEGAL-E, qui s'appuient sur le même modèle d'apprentissage, et utilisent une technique de généralisation descendante, basée sur la logique des propositions et sur la structure de treillis de Galois, pour produire un ensemble de descriptions structurées et ordonnées. Elles diffèrent dans leur approche de production de connaissances.*

Pour des raisons de complexité, seules deux variantes de LEGAL-E sont évaluées sur le problème de la prédiction de sites de jonctions introns-exons. Une comparaison à d'autres méthodes montrent que nos résultats sont meilleurs que ceux obtenus avec des méthodes symboliques, et sont relativement comparables à ceux des meilleures méthodes neuronales. Nous montrons enfin que LEGAL-E peut être vu comme un réseau de neurones multi-couches, simple et dynamique.

SUMMARY — A new approach based on Galois lattice for concept learning

The main goal of machine learning systems is to characterise a concept denoted by a set of examples. We have designed and implemented a symbolic-based method, LEGAL, which uses a top-down generalisation mecanism based on propositional logic and Galois lattice structure, to build a set of ordered and structured descriptions. Its major drawback relies on its time and space complexity when building learned knowledge.

Our goal in this paper is to present a new learning method LEGAL-E which uses a different approach allowing to reduce this drawback. Two variants of this method are tested onto the problem of splice junction sites prediction on primate genetic sequences. A comparison to others machine learning systems shows that our results are far better than those obtained with symbolic representation, and are as good as the best neural networks-based ones. We finally show that LEGAL-E can be assimilated to a simple and dynamic multi-layer neural network method.

1. INTRODUCTION

Les techniques d'Intelligence Artificielle, parmi lesquelles celles d'apprentissage, permettent de concevoir des systèmes dits "intelligents". L'apprentissage est la production et l'évaluation de modèles devant être exploités après validation par l'expert d'un domaine. Il existe plusieurs types d'apprentissage [Michalski & Kodratoff, 1990], [Mephu Nguifo, 1993]. Nous avons focalisé notre travail sur l'apprentissage inductif par détection de similarités à partir d'exemples (SBL: Similarity Based Learning), qui dénotent un concept.

[1] Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, UMR n°9928 CNRS-USTL 161 rue ada - 34392 Montpellier Cedex 5 ; e-mail: mephu@lirmm.fr ; Fax: 67 41 85 00 ; Tel: 67 41 85 83.

Dans le paradigme SBL, le problème de la formation des concepts est principalement lié à celui de la classification des objets. L'idée principale est de caractériser chaque sous-classe des objets de l'ensemble d'apprentissage. On pose l'*hypothèse forte selon laquelle les objets sont classifiables dans des sous-classes exclusives.* Ce type de méthodes SBL engendre une formule logique qui représente la connaissance apprise. Cette formule est une disjonction de conjonctions d'attributs binaires, où chaque conjonction est en même temps l'étiquette d'une sous-classe et la propriété qui exprime la relation d'appartenance à cette sous-classe d'objets. L'ensemble des objets est ainsi divisé en sous-classes disjointes. Dans ce cas, l'explication de la décision est donnée par une étiquette de classe, et la validation de la méthode est effectuée par une estimation statistique. Une méthode bien connue de ce type est la méthode ID3 développée par Quinlan [1986].

Dans notre approche, nous considérons qu'un objet doit pouvoir interagir presque de la même façon que son voisinage. Il devient par conséquent difficile d'exhiber une classification hiérarchique significative des objets. *Notre hypothèse est basée sur le fait qu'il y a différentes manières d'être considéré comme similaires, et qu'un objet peut être un élément de plusieurs sous-ensembles significatifs.* Les objets sont regroupés dans plusieurs sous-ensembles. Les méthodes d'apprentissage construisent dans ce cas un ensemble de conjonctions d'attributs où chaque conjonction est l'étiquette d'un sous-ensemble. Cependant, il n'y a pas de processus de décision particulière dérivant de la phase d'apprentissage, mais plutôt une certaine latitude est laissée pour élaborer plusieurs processus différents. LEGAL [Liquière & Mephu, 1990] est conçu à partir de ces idées, et le treillis de Galois est une des meilleures manières de produire ces recouvrements.

Le treillis de Galois ou treillis des concepts est une notion centrale dans un domaine de recherche relativement récent appelé Analyse Formelle des Concepts [Wille, 1982], basé sur un modèle ensembliste de concepts et de hiérarchies de concepts. Dans le modèle de Wille, on recherche des relations ou dépendances entre objets ou attributs, qui sont présentes dans le contexte initial. Alors que nous cherchons à caractériser un concept à travers le contexte initial qui le dénote et qui peut être simplement une vue partielle du concept, les données pouvant être incomplètes et erronées.

Le principal avantage du treillis de Galois réside dans son exhaustivité. De cet avantage, découle un inconvénient majeur: tous les algorithmes de construction du treillis ont une complexité exponentielle en fonction du nombre d'attributs et d'objets. Il est alors nécessaire d'introduire des heuristiques pour réduire cette complexité. Nous nous sommes appuyés sur le principe de la logique majoritaire déjà utilisée dans le système CALM [Quinqueton & Sallantin, 1986] pour définir ces heuristiques dans nos deux méthodes LEGAL et LEGAL-E. Les deux méthodes génèrent un sup-demi treillis de concepts, mais leur principale différence réside dans le fait que lors de la génération, LEGAL utilise tous les objets (exemples + contre-exemples) tandis que LEGAL-E se focalise seulement sur les exemples.

Deux variantes de la méthode LEGAL-E ont été implémentées et testées sur une application de biologie moléculaire (la prédiction de sites de jonctions d'épissage). Les résultats obtenus dans les deux cas montrent un comportement relativement identique des deux variantes. Cependant, une comparaison effectuée avec les résultats obtenus par d'autres méthodes d'apprentissage sur le même jeu de données, montre que LEGAL-E se comporte relativement mieux que certaines méthodes symboliques, et aussi bien que les méthodes neuronales.

Après un bref rappel sur la notion de treillis de Galois, nous allons introduire la méthode LEGAL au travers d'un petit exemple (figure 1). Nous définissons ensuite la méthode LEGAL-E et deux de ses variantes. Ces deux variantes de LEGAL-E sont évaluées sur un problème de biologie moléculaire, et comparées à d'autres méthodes d'apprentissage. Nous montrons enfin les relations existant entre LEGAL et réseau de neurones.

2. TREILLIS DE GALOIS

Nous introduisons la notion de treillis de Galois en définissant ses notions de base et en l'illustrant sur un exemple (figure 1 et 2). La représentation duale du treillis de Galois possède une utilité potentielle. En effet, après Barbut et Monjardet [1970], Wille [1992] montre que ce modèle est une nouvelle approche pour l'analyse de données, et pour les méthodes de représentation formelle de connaissance conceptuelle. Duquenne & Guigues [1986] s'intéressent également à la recherche de familles non redondantes d'implications contextuelles à partir de la structure de treillis de Galois. Godin et ses collègues [1986] utilisent ce modèle pour représenter une base d'informations dans laquelle on peut facilement naviguer pour rechercher des données.

O = {1,2,3,4,5,6,7} **A = {a,b,c,d,e,f,g,h}**

O \ A	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			1
7	1		1			1		

Figure 1 : Exemple de contexte.

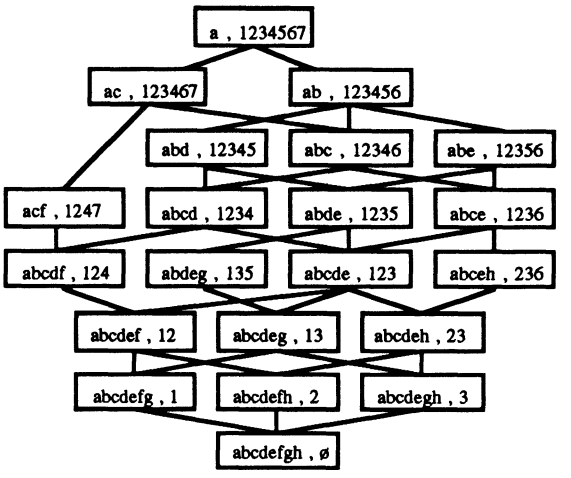


Figure 2 : Treillis de Galois de la figure 1.

Si l'objet o ∈ **O** vérifie l'attribut a ∈ **A** alors la case correspondante à la ligne o et à la colonne a est égale à 1.
Un rectangle matérialise un concept, et une arête entre 2 concepts représente la relation d'ordre entre les concepts.

Contexte :
Un *contexte* K est un triplet (**O**, **A**, **I**), où **O** (resp. **A**) est un ensemble fini d'objets (resp. d'attributs), et **I** ⊆ **O** x **A** est une relation binaire entre **O** et **A**, définie comme suit :
L'objet o ∈ **O** vérifie l'attribut a ∈ **A** ssi le couple (o, a) ∈ **I**. On a donc : (o, a) ∈ **I** ↔ o I a.

L'expression en *compréhension* d'un sous-ensemble d'objets X ⊆ **O** est définie par :
$X' = \{a \in \mathbf{A} \mid o\ I\ a \ \ \forall\ o \in X\}$ exemple : $X = \{1,2,7\} \Rightarrow X' = \{a,c,f\}$
De façon duale, l'expression en *extension* d'un sous-ensemble d'attributs Y ⊆ **A** est :
$Y' = \{o \in \mathbf{O} \mid o\ I\ a \ \ \forall\ a \in Y\}$ exemple : $Y = \{d,e\} \Rightarrow Y' = \{1,2,3,5\}$

Correspondance de Galois:
Soient f et g, deux applications définies respectivement de P(**O**) dans P(**A**) et de P(**A**) dans P(**O**) et qui respectivement à X associe X' et à Y associe Y'. A l'évidence f et g sont des applications monotones décroissantes satisfaisant les propriétés suivantes :

Soient $X_1, X_2, X \subseteq \mathbf{O}$, et $Y_1, Y_2, Y \subseteq \mathbf{A}$, $f : X \mapsto X'$ $g : Y \mapsto Y'$

(1) $X_1 \subseteq X_2 \Rightarrow f(X_1) = X_1' \supseteq X_2' = f(X_2)$. (2) $X \subseteq X''$ et $X' = X''' = f(g(X'))\ \forall\ X \subseteq \mathbf{O}$.

(1') $Y_1 \subseteq Y_2 \Rightarrow g(Y_1) = Y_1' \supseteq Y_2' = g(Y_2)$. (2') $Y \subseteq Y''$ et $Y' = Y''' = g(f(Y'))\ \forall\ Y \subseteq \mathbf{A}$.

L'application composée h = f•g (resp. h' = g•f) est une fermeture dans l'ensemble des parties $\mathcal{P}(\mathbf{O})$ (resp. $\mathcal{P}(\mathbf{A})$) [Barbut & Monjardet, 1970]. Par définition, le couple (f, g) forme une *correspondance de Galois* entre l'ensemble des parties $\mathcal{P}(\mathbf{O})$ et l'ensemble des parties $\mathcal{P}(\mathbf{A})$.

Concept :
Un *concept* dans (**O**, **A**, **I**) est un couple (X, Y), $X \subseteq \mathbf{O}$ et $Y \subseteq \mathbf{A}$, où X'=Y et Y'=X.
Autrement dit, X est l'ensemble de tous les objets qui vérifient tous les attributs de Y (extension), et Y est l'ensemble des attributs communs à tous les objets de X (compréhension).

La notion de concept est donc ici formée de deux parties: une *compréhension* constituée de tous les attributs qui sont à la fois nécessaires et suffisants pour l'appartenance au concept[2], et une *extension* formée de tous les objets vérifiant ces attributs. Dans la figure, un concept peut être vu comme un *rectangle maximal* de '1' au sens de l'inclusion [Guénoche, 1990].

Relation d'ordre :
Tous les concepts d'un contexte $\mathcal{K}$ sont ordonnés par la relation de *sous-concept / sur-concept.*

Soient $\mathcal{L}$ l'ensemble de tous les concepts de $\mathcal{K}$, $\mathcal{L} = \{(X,Y) \in \mathcal{P}(\mathbf{O}) \times \mathcal{P}(\mathbf{A}) \mid X = Y' \text{ et } Y = X'\}$

et $\leq$ une *relation d'ordre* sur $\mathcal{L}$ définie par : $(X_1, Y_1) \leq (X_2, Y_2)$ ssi $X_1 \subseteq X_2$ (ou $Y_1 \supseteq Y_2$)

On dit que: (X_1, Y_1) est un sous-concept (spécialisation) de (X_2, Y_2).
(X_2, Y_2) est un sur-concept (généralisation) de (X_1, Y_1).

Treillis de Galois d'une correspondance binaire :
L'ensemble $\mathcal{L}$ muni de la relation d'ordre $\leq$, $(\mathcal{L}, \leq)$, possède la structure d'un treillis complet, et est appelé *Treillis de concept* du contexte $\mathcal{K}$ et noté $\mathcal{L}(\mathcal{K})$, ou encore ***Treillis de Galois***, T, de la relation binaire I sur $\mathbf{O} \times \mathbf{A}$ (figure 2).

Il existe plusieurs algorithmes de construction des éléments du treillis, parmi lesquels on peut citer ceux de Chein, Norris, Ganter, et Bordat (voir [Guénoche, 1990]). Tous ces algorithmes génèrent les concepts du treillis, mais le dernier permet de construire plus facilement le treillis [Bordat, 1986]. C'est celui que nous utilisons dans nos deux méthodes. L'algorithme de Bordat construit les éléments du treillis et les arêtes de son graphe de Hasse[3]. Cette construction est basée sur la *relation de couverture* de la relation d'ordre $\leq$ [Bordat, 1986] [Liquière & Mephu, 1990].

La construction d'un treillis de Galois est exponentielle en fonction du nombre d'attributs, car le nombre de sommets obtenus peut être de la forme 2^n. Le nombre d'éléments du treillis construit est "exponentiel" en fonction de la taille des données. Dans la pratique, les exemples traités gardent souvent une taille raisonnable pour que l'on s'y intéresse de près.

En fait, la complexité est aussi fonction du contenu du tableau binaire, par conséquent le problème, dans la pratique, peut être résolu plus rapidement dans certains cas [Godin, 1989]. De plus, la complexité en temps et en mémoire est fortement réduite si le tableau initial est trié.

Remarque : Le Treillis peut être généré en largeur ou en profondeur. Nous avons implémenté la première : génération en *largeur* ou par *niveaux* (en anglais: "in breadth-first order").

[2] Elle se distingue avec la notion plus large de "concept" utilisé en Apprentissage pour se referer à un principe abstrait qui est en relation avec un sujet particulier. Une propriété d'un concept est qu'il est la généralisation d'un ensemble d'objets.

[3] C'est le graphe dont l'ensemble des sommets est l'ensemble des concepts du treillis et les arêtes correspondent à la relation de couverture de la relation d'ordre $\leq$.

3. TREILLIS DE GALOIS ET APPRENTISSAGE

Nous présentons ici les deux méthodes d'apprentissage LEGAL et LEGAL-E, et les illustrons sur l'exemple d'application de la figure 1.

Le treillis de Galois est construit à partir d'un ensemble d'objets décrits par un ensemble d'attributs binaires. Le processus de construction du treillis part du concept le plus général (décrivant tous les objets), et par spécialisations successives, aboutit à un ensemble de concepts. C'est bien une méthode descendante, qui est facilement exploitable en apprentissage. De plus, le treillis de Galois offre une représentation géométrique intuitive et facilement représentable en machine, sur laquelle on peut explorer les conjonctions d'attributs caractéristiques possibles, et apprendre aisément. La structure du treillis de Galois offre le plus large espace d'exploration des régularités.

Le treillis de Galois permet également d'engendrer facilement les implications d'attributs [Duquenne & al., 1986] - appelées règles - liées au contexte étudié. Ces règles sont généralement utilisées pour la construction de bases de connaissances, par exemple les règles de production dans les systèmes experts. Dans l'exemple précédent, on obtient à partir du treillis (figure 2) :

$b \Rightarrow a$	"b implique a"	$f \bullet g(\{b\}) = f(\{1,2,3,4,5,6\}) = \{a,b\}$
$f \Rightarrow a \wedge c$	"f implique a et c"	$f \bullet g(\{f\}) = f(\{1,2,4,7\}) = \{a,c,f\}$

Autrement dit - et on le constate sur la figure 1 - :

- "tout objet vérifiant la propriété b vérifie aussi la propriété a";
- "tout objet vérifiant la propriété f vérifie aussi les propriétés a et c".

L'apprentissage par l'intermédiaire du treillis limitera le nombre d'implications aux seules règles pertinentes du contexte, dans la mesure où elles seront vérifiées par beaucoup d'exemples. La recherche d'un système de règles (implications entre attributs) dans le système d'apprentissage Charade [Ganascia, 1988] est assimilable à la recherche d'implications contextuelles dans la structure du treillis de Galois qu'effectuent Duquenne & al. [1986]. Mais la structure du cube de Hilbert présente le désavantage de contenir des descriptions non maximales, et donc redondantes car elles n'apportent pas d'informations utiles pour l'exploitation des propriétés engendrées.

Nous avons ainsi conçu et implémenté une méthode inductive simple, conceptuelle, et empirique, d'apprentissage par détection de similarités, basée sur la notion de treillis de Galois pour construire un ensemble ordonné de faits (ou régularités) à partir des descriptions d'objets caractérisant le concept à apprendre. L'avantage principal de notre méthode est son exhaustivité bien que cela ait l'inconvénient de ralentir considérablement l'exécution dans le cas où il y a un grand nombre de données. Les heuristiques de choix des concepts pertinents du treillis réduisent considérablement cette complexité. Ces heuristiques sont basés sur les critères de validité et de quasi-cohérence.

3.1. Critères de validité et de quasi-cohérence

Le but recherché ici est la réduction de la construction du treillis aux seuls **concepts valides**. Nous définissons ci-après les notions de *complétude,* de *cohérence,* de *validité*, de *quasi-cohérence* et de *pertinence* liées à un concept. Mais avant, on peut remarquer que le treillis ne sera pas regénéré en entier. On va obtenir un *demi-treillis* qui représente notre *espace des hypothèses (concepts).* Cette réduction diminuera considérablement le temps de calcul.

Notations

$\mathbf{O}^+$: ensemble des exemples ; $\mathbf{O}^-$: ensemble des contre-exemples ; $\mathbf{O} = \mathbf{O}^+ + \mathbf{O}^-$

$\mathbf{U}$: l'ensemble des concepts valides ;

$\mathbf{C}$: l'ensemble des concepts valides et quasi-cohérents ;

$\mathbf{P}$: l'ensemble des descriptions pertinentes ;

α : nombre minimum d'exemples liés à un concept valide ;

β : nombre maximum de contre-exemples liés à un concept valide et quasi-cohérent ;

(X_0, Y_0) : le concept le plus général: $X_0 = \mathbf{O}$ et $Y_0 = \emptyset$.

Hypothèses

α et β sont les deux paramètres fixés par l'utilisateur, et respectivement appelés *seuil de validité* et de *quasi-cohérence*, avec les contraintes : (1) $0 \leq \alpha \leq |\mathbf{O}^+|$ (2) $0 \leq \beta \leq |\mathbf{O}^-|$

Complétude et Cohérence d'un concept :

• Un concept $(X, Y) \in \mathbf{L}$ est **complet** si sa description Y reconnaît tous les exemples d'objets du contexte ; autrement dit: $\mathbf{O}^+ \subseteq X$. On dit également que la description est complète.

• Un concept $(X, Y) \in \mathbf{L}$ est **cohérent** si sa description Y rejette tous les contre-exemples; autrement dit: $\mathbf{O}^- \cap X = \emptyset$. On dit aussi que la description est cohérente.

Ces deux critères peuvent parfois apparaître trop sévères, et contraignants. De plus, l'incomplétude de la description des objets peut induire une construction d'une connaissance erronée. Nous avons introduit deux nouveaux critères moins contraignants.

Validité et Quasi-cohérence d'un concept :

• Un concept $(X, Y) \in \mathbf{L}$ est **valide** si sa description Y reconnaît *suffisamment* d'exemples d'objets du contexte étudié. Sa première composante contient suffisamment d'objets qui sont des exemples. La description liée à ce concept est également valide.

$\mathbf{U} = \{(X, Y) \in \mathbf{L} \mid |X^+| \geq \alpha \text{ et } X \neq X_0\}$, autrement dit un concept (X, Y) est **valide** si *au moins* α exemples d'objets appartiennent à X.

• Un concept valide $(X, Y) \in \mathbf{L}$ est **quasi-cohérent** si sa description Y reconnaît *très peu* de contre-exemples. La description liée à ce concept est aussi valide et quasi-cohérente.

$\mathbf{C} = \{(X, Y) \in \mathbf{L} \mid |X^+| \geq \alpha,\ X \neq X_0 \text{ et } |X^-| \leq \beta\} = \{(X, Y) \in \mathbf{U} \mid |X^-| \leq \beta\}$,

autrement dit un concept valide (X, Y) est **quasi-cohérent** si *au plus* β contre-exemples sont des éléments de X.

Par définition, $\mathbf{C} \subseteq \mathbf{U}$.

Concept minimal, concept maximal :

La minimalité ou maximalité au sens de l'inclusion, est liée aux notions précédentes. Elle est définie sur des concepts de même type : complétude, cohérence, validité, et quasi-cohérence. Un concept valide $(X, Y) \in \mathbf{L}$ est **minimal** s'il ne couvre aucun autre concept valide $(X_1, Y_1) \in \mathbf{L}$.

Un concept valide et quasi-cohérent $(X, Y) \in \mathbf{L}$ est **maximal** s'il n'est couvert par aucun autre concept valide et quasi-cohérent $(X_1, Y_1) \in \mathbf{L}$.

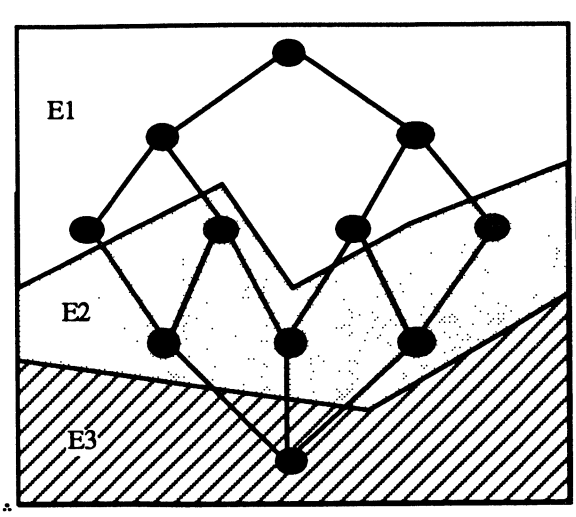


Figure 3 : Espace des concepts explorés.
E1 ∪ E2 = espace exploré; E3 = espace non exploré;
E1 = Sup demi-treillis engendré.

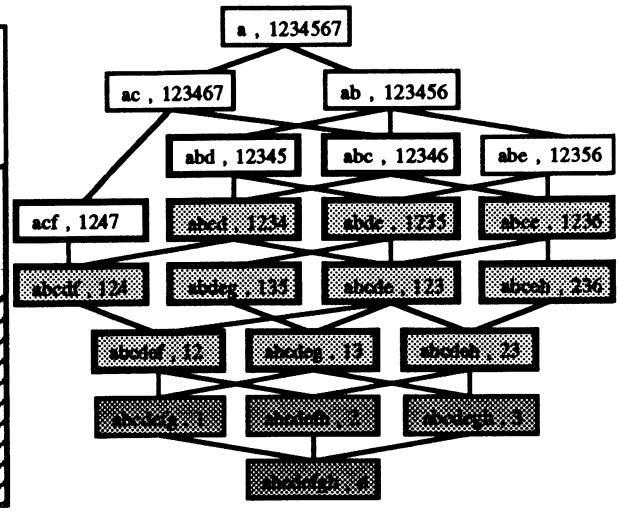
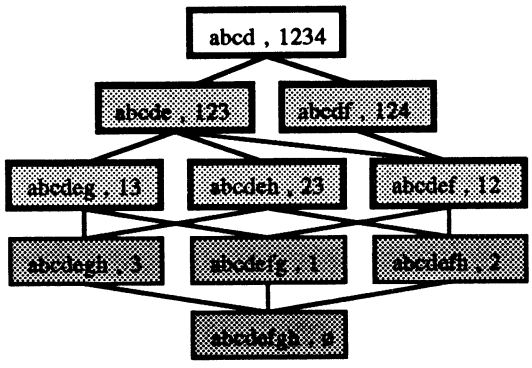


Figure 4 : Sup-demi treillis de LEGAL.
(voir légende ci-dessous)



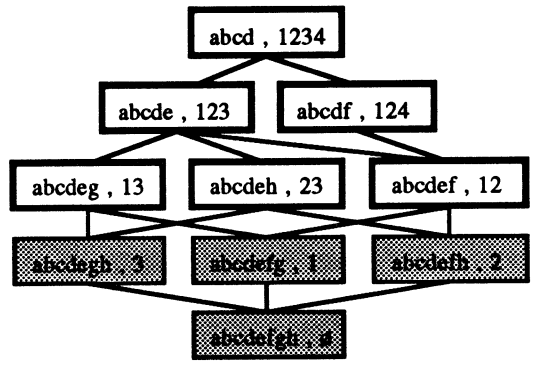


Figure 5 : Sup-demi treillis de LEGAL-E-2. **Figure 6 : Sup-demi treillis de LEGAL-E-1**

Légende : Concept valide ; Concept quasi-cohérent ; Concepts non engendrés

$O^+ = \{1,2,3,4\}$ $O^- = \{5,6,7\}$ $a = 2$ $b = 1$

Un rectangle représente un concept, et une arête entre 2 concepts matérialise la relation de sous-concept/sur-concept.

Pertinence d'un concept :
Les concepts pertinents sont ceux qui sont sélectionnés dans le treillis pour la phase de décision. Alors, quels concepts choisir ? Autrement dit, quelles heuristiques implémentées pour extraire les concepts pertinents ? Les différents critères mentionnés précédemment montrent bien l'existence d'une multitude de choix possibles. Des heuristiques peuvent également être conçues en combinant ces critères entre eux ou avec d'autres critères comme le rang d'un concept, ou les concepts feuilles du demi-treillis, ou même encore sur la longueur des descriptions de concepts (en terme de nombre d'attributs de la description).

Une *description pertinente* est liée à un concept pertinent. Dans la suite nous désignerons souvent une description d'un concept par le terme **régularité**.

En construisant un demi-treillis (voir figure 3), nous cherchons à déterminer l'ensemble **P** des descriptions - ou régularités - pertinentes. La phase de décision est élaborée à partir de **P**.

Pour étayer notre illustration, nous fixons les conventions suivantes :

• l'ensemble **O** des objets est découpé en un ensemble $\mathbf{O}^+$ de quatre exemples et un ensemble $\mathbf{O}^-$ de trois contre-exemples : $\mathbf{O}^+ = \{1,2,3,4\}$ $\mathbf{O}^- = \{5,6,7\}$

• un concept valide contient au moins deux exemples : $\alpha = 2$

• un concept valide et quasi-cohérent contient au plus un contre-exemple : $\beta = 1$

3.2. LEGAL - LEarning with GAlois Lattice [Liquière & Mephu Nguifo, 1990]

Nous résumons la méthode d'apprentissage LEGAL, et l'illustrons sur l'exemple précédent de la figure 1. LEGAL s'appuie sur le tableau binaire correspondant à tous les objets (exemples et contre-exemples) pour construire le Sup-demi treillis.

Le principe de l'algorithme d'apprentissage est celui de Bordat modifié tel que seuls les concepts valides sont générés. Bordat définit une structure de file des éléments du treillis, initialisée avec l'élément suprémum $\mathbf{O} \times \emptyset$, et dont les opérations associées sont :

- INIT : création de la file à partir de son 1er élément;
- FRONT : donne l'élément situé en tête de la file;
- SUPPRIMER : supprime l'élément situé en tête de la file;
- INSERER : insère un nouvel élément en queue de la file;
- FILEVIDE : booléen testant si la file est vide.

Une procédure RECHERCHE renvoie la valeur *vrai*, par la variable booléenne *trouvé*, si le nouvel élément (successeur) généré existe déjà dans le treillis, et *faux* sinon. Cette procédure effectue réellement la double opération de recherche-insertion dans le treillis. Les concepts valides et quasi-cohérents non maximaux ne sont pas insérés dans le treillis.

Nous introduisons une nouvelle fonction VALIDITE qui, pour un concept, retourne une des valeurs "valide", "valide et quasi-cohérent", ou "non valide" qui lui correspond.

Heuristique :
L'heuristique de LEGAL dépend du type de problème que l'on essaie de résoudre, à savoir assimilation (pas de contre-exemples) ou discrimination (exemples et contre-exemples) :

• En Assimilation, tous les concepts valides sont **pertinents**. Tous ces concepts sont aussi quasi-cohérents car il n'y a pas de contre-exemples. **P** contient les descriptions de **U**.

• En Discrimination, seuls les concepts valides, quasi-cohérents et maximaux sont **pertinents**. **P** contient les descriptions maximales de **C**.

Algorithme :

```
INIT(File, O × Ø);
repéter
        X × Y := FRONT(File);
        SUPPRIMER(File);
        pour tout X1 × Y1 successeur[4] de X × Y faire début
            si VALIDITE(X1) = "valide" alors
                RECHERCHE(X1 × Y1, trouvé);
                si non trouvé alors
                    si (VALIDITE(X1) ≠ "valide-coherent") alors INSERER(File, X1 × Y1);
        fin;
jusqu'à FILEVIDE;
```

[4] Est identique à: "couvert par".

Remarque 1 :

• La réduction du treillis aux seuls concepts valides peut diminuer considérablement la complexité de sa construction comme le montre la figure 4. Par conséquent, elle accroît l'intérêt de notre système pour des cas pratiques.

• Avec l'exemple d'application précédent, LEGAL explore 9 concepts et construit un demi-treillis de 7 concepts dont 3 sont pertinents (figure 4). On obtient l'ensemble : **P** = {**acf**, **abd**, **abc**}

• Cette démarche favorise certains attributs qui ont le même comportement sur les exemples. Par exemple, l'attribut **a** par rapport aux attributs **b**, **c**, et de même **b**, **c** par rapport à **d**.

• De plus, ce que l'on recherche en apprentissage c'est de caractériser un ensemble d'exemples, autrement dit de produire des descriptions qui caractérisent cet ensemble, en le discriminant des contre-exemples. Le fait de produire des caractérisations qui apparaissent sur des contre-exemples risque malencontreusement de favoriser la reconnaissance de ceux-ci.

• En outre, lorsque le nombre de contre-exemples est relativement élevé, la complexité de l'algorithme s'accroît considérablement.

Aussi avons-nous choisi de réduire ces inconvénients en limitant la construction des concepts sur l'ensemble des exemples, d'où la méthode LEGAL-E, dont deux variantes sont présentées et discutées ci-dessous.

3.3. LEGAL-E : Une réduction de LEGAL sur l'ensemble des exemples

Lors de la production des concepts, seuls les contextes (tableau binaire) diffèrent en présence de contre-exemples. Alors que LEGAL utilise le contexte entier (**O**, **A**, **I**) - exemples et contre-exemples, LEGAL-E se limite au contexte ($\mathbf{O}^+$, **A**, **I**) contenant seulement les exemples.

Le principe de LEGAL-E consiste à construire les concepts valides sur l'ensemble des exemples et ensuite tester leur cohérence sur les contre-exemples (figures 5 et 6).

Nous présentons ici deux variantes de la méthode LEGAL-E. Nous montrons sur un exemple d'application en grandeur réelle que les résultats obtenus sont relativement identiques.

3.3.1. LEGAL-E-1 : La variante 1 de LEGAL-E

Algorithme 1 :

C'est le même algorithme que celui de LEGAL.

Remarque 2 :

• Avec l'exemple d'application précédent, LEGAL-E-1 explore 1 concept au lieu de 9 (comme le fait LEGAL) et construit un demi-treillis contenant 1 concept valide et pertinent (figure 5). On obtient l'ensemble suivant : **P** = { abcd }

• S'il semble évident que la réduction du contexte de construction des concepts valides à celui des exemples va considérablement réduire la complexité de l'apprentissage, il est nécessaire de s'assurer que la connaissance produite est meilleure. La réponse sur l'exemple d'application est positive, et de plus semble pallier les inconvénients cités dans la remarque 1. En effet, par rapport à l'ensemble des conjonctions générées par LEGAL, une analyse visuelle du contexte (figure 1) montre que la conjonction (a et b et c et d) caractérise mieux les exemples (tous sont reconnus) et les discrimine bien des contre-exemples (tous sont rejetés).

3.3.2. LEGAL-E-2 : La variante 2 de LEGAL-E

Dans cette deuxième variante de LEGAL-E, l'heuristique diffère en discrimination. La notion de maximalité est supprimée, et donc tous les concepts valides et quasi-cohérents sont pertinents. La fonction RECHERCHE insère dans le treillis tous les concepts valides et quasi-cohérents.

Heuristique :

• En Assimilation, tous les concepts valides sont **pertinents**. Tous ces concepts sont aussi quasi-cohérents car il n'y a pas de contre-exemples. **P** contient les descriptions de **U**.

• En Discrimination, seuls les concepts valides, quasi-cohérents sont **pertinents**. **P** contient les descriptions de **C**.

Algorithme 2 :

```
INIT(File, O × Ø);
repéter
        X × Y := FRONT(File);
        SUPPRIMER(File);
        pour tout X1 × Y1 successeur de X × Y faire début
            si VALIDITE(X1) = "valide" alors
                RECHERCHE(X1 × Y1 trouvé);
                si non trouvé alors INSERER(File, X1 × Y1);
        fin;
jusqu'à FILEVIDE;
```

Remarque 3 :

• Avec l'exemple d'application précédent, LEGAL-E-2 explore 9 concepts et construit un demi-treillis contenant 6 concepts valides et pertinents (figure 6). On obtient l'ensemble suivant :

$$\mathbf{P} = \{ abcd, abcdf, abcde, abcdeg, abcdeh, abcdef \}.$$

• Par rapport à la première variante, celle-ci présente le défaut d'accroître la complexité de l'apprentissage. Cependant elle génère un ensemble de descriptions qui font apparaître des attributs pertinents du point de vue de l'utilisateur (par exemple, l'attribut f est vérifié par trois objets et un contre-exemple, α et β sont fixés à 2 et 1). Ces attributs pourront servir à étayer les explications.

Une comparaison de ces deux variantes est réalisée dans la section 5 sur une application réelle.

4. LE PROCESSUS DE DÉCISION

Nous abordons la phase déductive de notre système. Celui-ci dispose d'un ensemble de connaissances apprises. Celles-ci sont les descriptions pertinentes obtenues et ne sont généralement pas réductibles à un seul énoncé (une seule régularité). La question est

"Comment utiliser cet ensemble de régularités pour prendre des décisions ?"

Les décisions sont très souvent réductibles à la classification ou reconnaissance d'un nouvel objet. Plusieurs considérations sont alors possibles. Nous savons qu'Apprentissage et Décision sont étroitement liés, car la stratégie de décision choisie ne doit en aucun cas remettre en cause le mécanisme d'apprentissage utilisé. Contrairement aux méthodes comme ID3, nous avons conçu notre fonction d'apprentissage de manière à favoriser le développement de plusieurs processus de décision. Ainsi la structure de demi-treillis obtenue permet :

• d'utiliser l'ensemble des régularités pertinentes pour développer un raisonnement empirique. Ce processus semble être le plus naturel, et mieux encore une manière logique simple de raisonner. Il est basé sur le principe de vote majoritaire déjà présent dans le système CALM [Quinqueton & Sallantin, 1986].

• d'utiliser les régularités du treillis pour développer un raisonnement analogique. Le raisonnement est basé sur la notion de proximité. La proximité peut être liée à l'objet le plus "proche" ou à l'ensemble des objets les plus "proches" [Mephu Nguifo, 1992].

• de pondérer les attributs des régularités pertinentes en tenant compte de la structure du treillis (ordre entre les régularités).

• etc...

Nous nous intéressons ici au raisonnement empirique qui est identique pour les deux méthodes LEGAL et LEGAL-E.

4.1. Principe

Ce type de raisonnement est le plus naturel à utiliser, au vu des critères de construction et de sélection des régularités pertinentes. Il est basé sur les notions de justification, réfutation et silence.

(1) Un objet est considéré comme un *exemple* s'il vérifie **suffisamment** de descriptions pertinentes. L'objet est dit **justifié** car il est reconnu comme étant un exemple.

(2) Un objet est considéré comme un *contre-exemple* s'il vérifie **peu** de descriptions pertinentes. L'objet est dit **réfuté** car il est reconnu comme n'étant pas un exemple.

(3) Dans le cas contraire, le système est *silencieux*. L'objet n'est ni un exemple ni un contre-exemple. Le **silence** est défini lorsque le système ne reconnaît pas un objet.

Nous introduisons pour ce faire, un *seuil de justification* θ_a et un *seuil de réfutation* θ_b. Ces deux seuils sont proposés par le système (Cf. section 4.2), puis confirmés ou infirmés par l'usager. Dans le cas d'une infirmation, l'usager attribue de nouvelles valeurs, ou révise la connaissance apprise en modifiant certains paramètres d'apprentissage tels que le langage de description, les paramètres d'induction, l'ensemble des exemples, etc...

Les deux paramètres de justification et de réfutation sont des pourcentages satisfaisant les conditions suivantes :

(i) $0 \leq \theta_a \leq 100$ $\quad\quad$ θ_a : Pourcentage de justification;

(ii) $0 \leq \theta_b \leq \theta_a$ $\quad\quad$ θ_b : Pourcentage de réfutation.

Définitions :

Pour un objet o_i, on définit : $R_i = \{r \in \mathbf{P} \mid r \text{ est vérifiée par l'objet } o_i\}$ c'est-à-dire R_i est l'ensemble de toutes les régularités pertinentes vérifiées par l'objet o_i.

On définit également un *pourcentage de vérification* θ_i associé à un objet o_i, de la manière suivante :

$$\theta_i = \frac{n \times 100}{|\mathbf{P}|}$$

où : - $n = |R_i|$ = nombre de descriptions pertinentes vérifiées par l'objet o_i;

- $|\mathbf{P}|$ = cardinal de $\mathbf{P}$, c'est le nombre de descriptions pertinentes;

• o_i est **justifié** si $\theta_a \leq \theta_i$;

• o_i est **réfuté** si $\theta_b > \theta_i$;

• Le système reste **silencieux** si $\theta_b \leq \theta_i < \theta_a$.

4.2. Calcul des seuils de décision

Nous pensons que le système acquérant les connaissances après observation d'un ensemble d'objets, il est alors possible et voire plausible que l'on utilise cet ensemble d'objets pour proposer à l'usager, les seuils de justification θ_a et de réfutation θ_b convenables. En effet la connaissance apprise pour être représentative du concept à apprendre doit déjà couvrir les exemples d'apprentissage.

Un critère strict de sélection des "meilleurs" seuils de décision peut être calqué sur le critère de complétude et de cohérence défini précédemment. Cela revient à choisir les seuils de décision de manière que dans l'ensemble d'apprentissage, tous les exemples soient justifiés et que tous les contre-exemples soient réfutés. Ce choix n'est pas judicieux pour les raisons suivantes :

- Non conformité au processus d'apprentissage;
- Non résistance aux données incomplètes et bruitées.

Aussi, nous assouplissons ce critère tel que de "bons" seuils de décision doivent permettre de reconnaître *presque tous* les exemples et de rejeter *presque tous* les contre-exemples. Rappelons que chaque régularité pertinente est vérifiée par au moins α exemples et éventuellement par au plus β contre-exemples. L'idée est donc la suivante:

(i) Déterminer les pourcentages de vérification θ_k, $\forall\ o_k \in \mathbf{O}$, $1 \leq k \leq |O|$

(ii) Soient les pourcentages θ_i, $\forall\ o_i \in \mathbf{O}^+$, $0 \leq i \leq |\mathbf{O}^+|$, des objets de $\mathbf{O}$ qui sont exemples : Le système essaie, par dichotomie dans l'intervalle [0, 100], de déterminer un seuil de justification θ_a, tel que la propriété $\theta_i \geq \theta_a$ soit vraie pour *suffisamment* d'exemples.

(iii) De même, soient θ_j, $\forall\ o_j \in \mathbf{O}^-$, $0 \leq j \leq |\mathbf{O}^-|$, les pourcentages des contre-exemples :

Le système essaie, par dichotomie dans l'intervalle $[0, \theta_a[$, de déterminer un seuil de réfutation θ_b, tel que la propriété $\theta_j \geq \theta_b$ soit vraie pour *peu* de contre-exemples.

Remarque 4 :

- Les termes "suffisant" et "peu" sont en pratique choisis par l'usager.
- En assimilation, le seuil de réfutation est fixé par l'usager.
- Le système peut ne pas trouver de solutions. Dans ce cas, les valeurs sont fixées par l'usager, ou mieux encore l'usager peut choisir de réviser la connaissance apprise, car elle ne caractérise pas bien les objets d'apprentissage.
- Les valeurs proposées par le système doivent être confirmées par l'usager. Si elles sont infirmées, alors les nouvelles valeurs sont décidées par l'usager.

5. EXPÉRIMENTATION : PRÉDICTION DE SITES DE JONCTIONS D'ÉPISSAGE

Le domaine de la Biologie Moléculaire utilise des sytèmes de l'Intelligence Artificielle, et particulièrement d'Apprentissage pour valider ses hypothèses. En effet, la complexité des problèmes rencontrés dans ce domaine réduit l'utilisation de méthodes informatiques classiques.

Nous avons appliqué notre système au problème de la prédiction de sites de jonction d'épissage, pour lequel nous avons obtenu un bon résultat empirique [Mephu & Sallantin, 1993] en nous intéressant au jeu de données[5] utilisé par Noordewier, Towell, et Shavlik [1991] pour évaluer un algorithme d'apprentissage hybride basée sur les réseaux de neurones, KBANN.

---

[5]Ces données sont extraites de la famille des gènes primates de la banque de données biologiques GenBank 64.1, et les résultats se trouvent sur la machine UNIX ics.uci.edu accessible par la commande ftp (login "anonymous"). Le répertoire est "Pub/Machine-Learning-databases/Molecular-Biology".

5.1. Le problème

Les gènes des Primates sont composés de parties codantes - appelées exons - et de parties non codantes ou introns. La frontière entre les introns et les exons forme un site de jonction d'épissage. Toutes les jonctions d'épissage connues sont divisées en 2 sites : les sites *donneurs* et les sites *accepteurs*. Un site accepteur est la frontière entre un intron et un exon (encore appelé **jonction 3'**), alors qu'un site donneur est la frontière entre un exon et un intron (**jonction 5'**). Un segment d'intron ou d'exon est une suite de nucléotides ou bases de longueur quelconque, une base étant symbolisée par l'une des quatre lettres suivantes: A, G, C, T.

• *Ensemble d'exemples* :
L'ensemble des données contient 767 sites de jonctions 3' (soit 25%), 768 sites de jonctions 5' (soit 25%), et 1655 segments de faux sites (ni jonction 3', ni jonction 5'). Tous les segments de ces 3 sous-ensembles sont de longueur 60 bases, et sont extraits de manière symétrique autour du site, à partir de la position -30 jusqu'à la position +30. Certains caractères autres que les bases apparaissent et indiquent une incertitude à cette position du segment. Ce sont les caractères : D, N, S, et R qui correspondent respectivement à (A ou G ou T), (A ou G ou C ou T), (C ou G), (A ou G).

• *Comparaison des résultats* :
Utilisant une technique de validation croisée sur 1000 segments choisis de manière aléatoire de cet ensemble, Noordewier & al. ont obtenus des taux d'erreur présentés dans la figure 16 pour plusieurs algorithmes d'apprentissage de type symbolique (ID3, COBWEB) et neuronal (KBANN, BACKPROP, PEBLS, PERCEPTRON).

Remarque 5 :

• Il apparaît clairement dans ces ensembles que tous les vrais sites de jonction ont le consensus 'AG' pour les 3' aux positions 29-30 et 'GT' pour les 5' aux positions 31-32. Sur les 1655 segments de faux sites, il y en a 8 (soit 0,48%) qui ont les deux consensus, 119 (soit 7,19%) qui ont le consensus 'AG', 90 (soit 5,44%) qui ont le consensus 'GT', et 1438 (soit 86,89%) qui n'ont aucun consensus. Le fait d'utiliser cet ensemble comme ensemble de contre-exemples, comme le font Towell, Noordewier et Shavlik, peut altérer les résultats (taux d'erreur) sur ces faux sites à cause du fait qu'il y a beaucoup de segments qui seront nécessairement rejetés par l'absence de consensus.

• Pour éviter ce biais, nous avons procédé comme précédemment en n'utilisant que les exemples et les contre-exemples qui contenaient le consensus. Ainsi nous avons obtenu 4 ensembles de (384+384) exemples de sites 3', (60+59) contre-exemples de sites 3', (384+383) exemples de sites 5', et (45+45) contre-exemples de sites 5'. Chaque ensemble a été divisé en deux suivant le principe de validation croisée, la première moitié pour l'apprentissage et l'autre pour le test.

• De plus nous avons conservé l'ordre alphabétique des séquences pour éviter d'avoir une similarité trop forte entre l'ensemble d'apprentissage et l'ensemble test.

5.2. Une comparaison des deux variantes de LEGAL-E

Nous avons fait une étude comparative des deux variantes de LEGAL-E au travers de leur capacité à prédire les sites de jonctions d'épissage.

Chaque nucléotide est décrit par une suite constante de propriétés binaires. Nous avons choisi un code de 6 attributs, nous permettant de décrire la présence d'un singleton et d'un doublet à une position. Le nombre d'attributs positionnés utilisés pour construire le sup-demi treillis correspondant est de 360 dans le cas des sites de jonction 3' et 5'.

Les deux paramètres d'apprentissage ont été choisis de manière aléatoire par essais/erreurs. Le seuil de validité est défini par la présence d'au moins 200 exemples dans un concept valide (soit environ 50%). De même, les concepts quasi-cohérents contiennent au plus 4 (resp. 3) contre-exemples pour les jonctions 3' (resp. 5') - soit environ 6,5%.

	Jonctions 3'		Jonctions 5'	
	RV	RP	RV	RP
Variante 1	2986	92	1928	104
Variante 2	3297	403	2282	458

Figure 7 : Nombre de régularités produites par LEGAL-E.
RV: Régularités valides. RP: Régularités pertinentes.

LEGAL-E génère un ensemble de régularités valides et quasi-cohérentes dont les totaux sont présentés dans la figure 7. Parmi toutes ces régularités, certaines sont sûrement plus pertinentes que d'autres. Le nombre de régularités très élevé peut sembler non pertinent pour l'usager. Aussi l'étude de la pertinence d'une régularité peut être effectuée en utilisant un principe de sanctions/récompenses. Les régularités les plus pertinentes sont celles qui seront les plus récompensées, et les autres les moins pertinentes. Mais, l'effet pervers de ce mécanisme est que le système sanctionne ou récompense les mêmes régularités autant de fois qu'on lui présente un même objet. Il est alors nécessaire de recourir à l'usager pour confirmer les sanctions ou les récompenses.

Chacune des variantes de LEGAL-E détermine un seuil de justification et de réfutation qui lui semble judicieux pour valider l'ensemble des régularités apprises. Nous considérons que les seuils de justification et de réfutation sont identiques. Nous avons fait varier le seuil de justification entre les valeurs (10% à 30%) afin d'étudier le comportement des deux variantes. Les figures 8 à 15 récapitulent les taux d'erreur (arrondi à l'entier le plus proche) en prédiction pour chaque variante dans le cas des jonctions 3' et 5', et sur les ensembles d'apprentissage et de test.

Les figures 8 à 15 montrent qu'avec un seuil de justification bas, la variante 2 se comporte mieux que la première. Par contre, lorsque le seuil est élevé, la première variante fournit des résultats meilleurs. Cependant, les meilleurs résultats sont obtenus à des seuils de justification différents pour les deux variantes. Ces résultats sont présentés dans la figure 16, et comparés aux résultats d'autres méthodes d'apprentissage. Pour cela, les seuils de justication sont fixés à 19% (variante 2) et 22% (variante 1) dans le cas des jonctions 3'. Ils sont fixés à 15% (variante 2) et 20% (variante 1) dans le cas des jonctions 5'.

La variante 1 prédit mieux les jonctions 3', et la variante 2 prédit mieux les jonctions 5'. Nous remarquons que les résultats des deux variantes sont relativement identiques. Aussi, en raison de sa complexité réduite, la variante 1 peut être préférée à la variante 2.

5.3. Une comparaison à d'autres méthodes d'apprentissage

Utilisant une technique de validation croisée sur 1000 segments choisis de manière aléatoire de cet ensemble, Towell, Noordewier & Shavlik ont obtenu des taux d'erreur (figure 16) pour plusieurs algorithmes d'apprentissage de type symbolique (ID3, COBWEB) et neuronal (KBANN, BACKPROP, PEBLS, PERCEPTRON).

θ_a (%)	Variante 1		Variante 2	
	VJ	FJ	VJ	FJ
10	1	18	4	10
15	4	10	6	8
20	6	8	9	8
25	10	7	13	7
30	14	5	17	5

Figure 8. Taux d'erreur (%) de LEGAL-E sur l'ensemble d'apprentissage des Jonctions 3'.
θ_a:Seuil Justification VJ:Vraie Jonction FJ:Fausse Jonction.

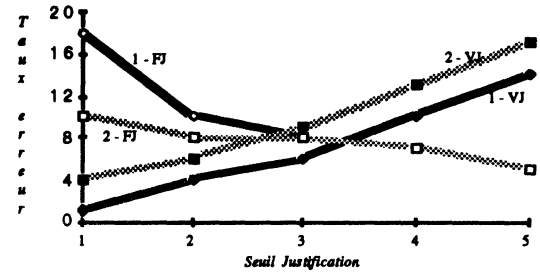


Figure 9. Graphique de la figure 8

θ_a (%)	Variante 1		Variante 2	
	VJ	FJ	VJ	FJ
10	4	25	5	17
15	6	14	9	12
20	8	14	11	8
25	10	8	16	5
30	17	5	20	5

Figure 10. Taux d'erreur de LEGAL-E sur l'ensemble test des Jonctions 3'.
θ_a:Seuil Justification VJ:Vraie Jonction FJ:Fausse Jonction.

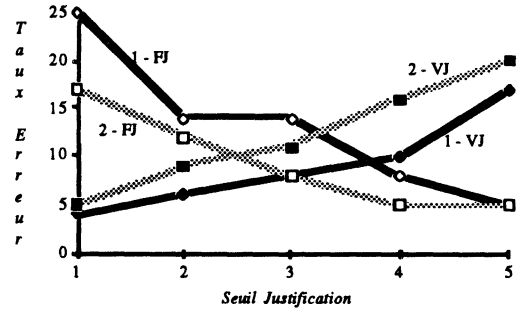


Figure 11. Graphique de la figure 10

θ_a (%)	Variante 1		Variante 2	
	VJ	FJ	VJ	FJ
10	0	20	1	7
15	1	7	3	4
20	3	4	8	2
25	9	4	14	0
30	13	0	22	0

Figure 12. Taux d'erreur de LEGAL-E sur l'ensemble d'apprentissage des Jonctions 5'.
θ_a:Seuil Justification VJ:Vraie Jonction FJ:Fausse Jonction.

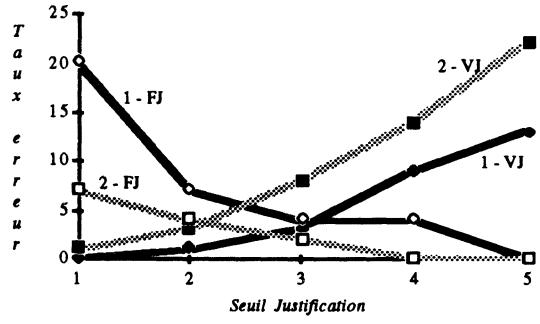


Figure 13. Graphique de la figure 12

θ_a (%)	Variante 1		Variante 2	
	VJ	FJ	VJ	FJ
10	1	29	2	16
15	4	13	5	4
20	5	9	10	2
25	10	4	14	2
30	13	2	27	0

Figure 14. Taux d'erreur de LEGAL-E sur l'ensemble test des Jonctions 5'.
θ_a:Seuil Justification VJ:Vraie Jonction FJ:Fausse Jonction.

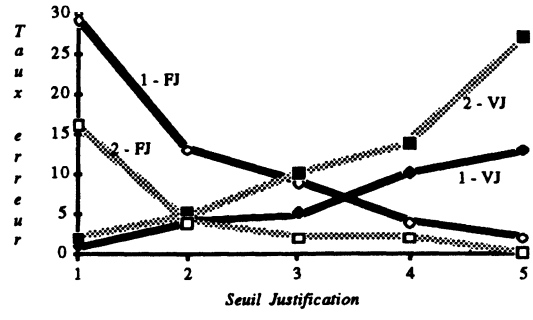


Figure 15. Graphique de la figure 14

$$\text{Taux d'erreur (\%)} = \frac{\text{Nombre d'instances mal prédites} \times 100}{\text{Nombre total d'instances}}$$

Les résultats de la figure 16 confirment donc la robustesse et l'efficacité de notre méthode. En effet, elle obtient des résultats aussi bons que les méthodes basées sur les réseaux de neurones, outre l'avantage de l'argumentation qu'elle possède sur celles-ci. Comparée aux méthodes symboliques comme ID3, ou COBWEB, elle fournit des résultats meilleurs. En outre, la remarque 5 faite précédemment ne peut que renforcer nos résultats.

Ensemble	VJ 3'	VJ 5'	FJ
KBANN	08.47	07.56	04.62
BACKPROP	10.75	05.74	05.29
PEBLS	07.55	08.18	06.86
PERCEPTRON	17.41	16.32	03.99
ID3	13.99	10.58	08.84
COBWEB	09.46	15.04	11.80
Nearest Neighbour	09.09	11.65	31.11

	VJ 3'	VJ 5'	FJ 5'	FJ 3'
LEGAL-E				
Variante 1	08.23	04.96	08.89	09.66
Variante 2	09.63	04.96	04.44	10.12

Figure 16 : Taux d'erreur en prédiction

Les résultats des autres méthodes ont été obtenus à l'Université du Wisconsin avec quelquefois des implémentations locales d'algorithmes publiés (voir Towell & al., [1992]). VJ: Vraie Jonction FJ: Fausse Jonction

Discussion :

Les résultats de LEGAL-E sont probablement imputables à l'exhaustivité du treillis, mais aussi au fait que le treillis de Galois peut être assimilable à l'architecture simple et dynamique d'un réseau de neurones comme le montre la figure 17. Les concepts du Sup-demi treillis constituent les noeuds d'un réseau de neurones multi-couches, sur lesquels il est possible de définir une fonction d'activation. Les concepts-feuilles du demi-treillis constituent les noeuds de la couche d'entrée. Le concept supérieur du demi-treillis constitue le seul nœud de la couche de sortie. Tous les autres concepts du demi-treillis font partie de la couche cachée qui elle-même peut être divisée en plusieurs couches. Mais il n'y a pas de structuration ni d'ordre a priori sur les différentes couches cachées. La simplicité provient du fait que les opérations manipulées se limitent à de simples conjonctions ou disjonctions de vecteurs binaires décrivant les exemples du concept à apprendre, la valeur des poids de connexions étant égale à 1. Le dynamisme vient de l'architecture qui est fonction du contenu du tableau binaire, soit donc de la description des exemples. Elle n'est pas figée comme dans les méthodes neuronales classiques généralement utilisées. On remarquera d'ailleurs que dans la méthode KBANN [Towell & al., 1992], on utilise une connaissance du domaine et on effectue un prétraitement sur les données initiales avant de définir la "meilleure" architecture adapté au problème. Ce qui semble justifier les bons résultats de cette méthode.

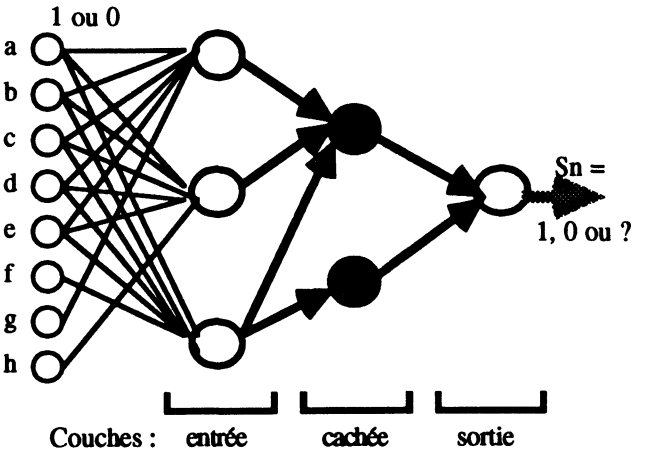


Figure 17 : Exemple d'architecture correspondant à la figure 6;.

Seules les connexions nécessaires sont représentées. La valeur des poids de connexion est initialement égale à 1. Les neurones de la couche d'entrée correspondent aux concepts feuilles du Sup-demi treillis. La valeur de sortie du neurone quelconque j est équitablement répartie comme valeur d'entrée sur les k neurones qui sont extrémité d'une connexion d'origine j (la valeur de sortie de j est divisée par k, et cette nouvelle valeur est transmise aux k neurones comme valeur d'entrée)

Au niveau de la couche d'entrée, les entrées de chaque neurone sont des valeurs binaires fonction de la description d'un objet, et sa sortie (égale à 0 ou 1) est le produit (conjonction) de ces valeurs. Sur les couches cachées, les entrées de chaque neurone sont des valeurs réelles positives et sa sortie est la somme pondérée de ces valeurs. La couche de sortie comprend un seul neurone qui correspond au concept supérieur du demi-treillis. Sur cette couche de sortie, les entrées du seul neurone sont des valeurs réelles positives et en fonction de la somme pondérée des entrées, la sortie est l'une des trois valeurs 1 (exemple), 0 (contre-exemple) ou ? (silence. Le '?' peut être remplacé par -1). Les deux seuils de décision sont des paramètres du neurone de sortie. Si le silence est supprimé (seuil de justification égal au seuil de réfutation), alors la fonction d'activation sur la dernière couche est la fonction de Heaviside.

En outre, plusieurs études comparatives avec les méthodes symboliques montrent que les méthodes neuronales [Kodratoff, 1993] :

- apprennent mieux en présence de données bruitées. LEGAL-E résiste bien au bruit à cause de l'utilisation de critères de validité et de quasi-cohérence au lieu des critères de complétude et de cohérence.
- apprennent mieux quand la connaissance du domaine est pauvre. Mis à part les attributs nécessaires pour la description des objets, LEGAL-E n'utilise aucune autre connaissance du domaine.
- apprennent très lentement. Ceci est également le cas pour LEGAL-E.
- ont besoin d'un travail d'affinage énorme pour apprendre proprement. Le choix des seuils de validité et de quasi-cohérence peut nécessiter un temps de travail considérable.
- ne délivrent pas de règles compréhensibles. LEGAL-E est une méthode symbolique capable de produire des règles compréhensibles par l'usager.

Tout ceci explique donc pourquoi une méthode symbolique telle que LEGAL-E fournit des résultats identiques sinon meilleurs que certaines méthodes neuronales.

6. CONCLUSION

Le bon comportement de LEGAL-E par rapport à LEGAL sur l'exemple de la figure 1, ne peut pas a priori se généraliser dans toutes les applications. Il serait intéressant de confirmer ce résultat par des expérimentations sur des problèmes réels. Cependant on peut à l'évidence

prévoir que dans le cas où l'usager a une certaine assurance sur la notion d'exemple et de contre-exemple, LEGAL-E devrait être utilisé. Si par contre, il existe des difficultés à choisir les exemples et les contre-exemples, autrement dit si l'on se trouve dans un cas où un objet peut avoir le statut d'exemple ou de contre-exemple, alors il peut être nécessaire de recourir à la méthode LEGAL.

L'intérêt principal de notre méthode découle de la structure du treillis de Galois. Le treillis de Galois offre l'espace de concepts le plus large et le plus concis, et de plus semble présenter des similitudes avec les réseaux de neurones. Les régularités qui en découlent sont par conséquent ordonnées par la relation de spécialisation/généralisation. Cette structuration offre la possibilité de réduction ou d'extension des régularités pertinentes lors d'une éventuelle révision des connaissances. Nous avons donc conçu et implémenté un nouveau système d'apprentissage de concepts à partir d'exemples et de contre-exemples. Nous l'avons en outre doté de deux algorithmes qui présentent chacun son avantage et son inconvénient.

On peut remarquer que la manière dont on formule et représente des données influe nécessairement sur le raisonnement que l'on peut adopter, et réciproquement. L'utilisation du treillis de Galois a bien évidemment une incidence sur la manière de décrire les objets, car il restreint le langage de description à celui de la logique des propositions.

L'incomplétude des connaissances apprises, et par conséquent l'erreur de la déduction qui en découle, nécessite qu'un contrôle du raisonnement soit effectué par le système, ou par l'utilisateur. Le système pourra alors s'appuyer sur ses méthodes de contrôle pour argumenter ses décisions. Les concepts de LEGAL-E sont des couples de sous-ensembles d'objets ou d'attributs maximaux au sens de l'inclusion. Lorsqu'un sous-ensemble d'exemples de l'ensemble d'apprentissage est constitué d'*exemples presque identiques*, alors LEGAL-E générera très peu de régularités (voire une) pour caractériser ce sous-ensemble. Par conséquent, même s'ils sont en nombre suffisant (largement supérieur au seuil de validité), peu de régularités correspondront aux objets de ce sous-ensemble. Et donc avec une déduction empirique où toutes les régularités ont un poids identique, ces exemples ont de fortes chances de ne jamais être reconnus par LEGAL-E. De même, lorsque les critères de validité et de quasi-cohérence sont élevés, les exemples isolés, très peu similaires aux autres exemples ne seront sûrement pas caractérisés par les régularités apprises. Ces exemples peuvent néanmoins vérifier quelques régularités, mais ils en vérifient très peu pour être reconnus comme exemple. Nous avons implémenté un mécanisme de contrôle pour répondre à ce problème [Mephu Nguifo, 1992]. Mais il est aussi possible de construire une méthode de déduction plus fine, dans laquelle de nouveaux critères de sélection et d'utilisation des concepts pertinents seront développés, comme un critère de représentativité d'un concept.

Godin & al. [1991] présentent une implémentation d'un algorithme incrémental de construction du treillis de Galois. Ce qui peut rendre possible l'utilisation d'une approche globale ou incrémentale de construction d'hypothèses par la méthode d'apprentissage.

Les deux paramètres d'apprentissage (α et β) sont fixés aléatoirement pour la mise en œuvre des seuils de validité et de quasi-cohérence. De manière aléatoire, l'utilisateur par essai/erreurs va choisir α et β jusqu'à ce qu'il soit satisfait des résultats obtenus en apprentissage [Lagrange & al., 1993], mais aussi en décision. Il serait judicieux en pratique de faciliter cette tâche de l'utilisateur.

Disponibilité : Le programme LEGAL-E est disponible sur demande, et gratuit pour les universitaires.

Remerciements
L'auteur a bénéficié du soutien financier de l'Association Française de lutte contre la Myopathie. Ce travail fait partie du programme de recherche du GDR Informatique et Génômes. L'auteur tient à remercier Bruno Leclerc, Michel Liquière, Jean Sallantin et les relecteurs anonymes pour leurs remarques pertinentes sur ce travail. Je remercie également toute l'équipe de l'Atelier de BioInformatique de l'Institut Curie, en particulier Alain Viari et Joël Potier pour leur aide technique.

BIBLIOGRAPHIE

BARBUT M., MONJARDET B., *Ordre et Classification, Algèbre et Combinatoire*, T.1, Chap.4: "Fermetures, Correspondances de Galois, Treillis d'une corresp.", Paris, Hachette 1970.
BORDAT J.P., "Calcul pratique du treillis de Galois d'une correspondance", *Math. Sci. Hum.*, 24ème année, n° 96, 1986, pp 31-47.
DUQUENNE V., GUIGUES J.-L. "Familles minimales d'implications informatives resultant d'un tableau de données binaires", *Math. Sci. Hum.*, n°95, 1986, pp.5-18.
GANASCIA J.G., "CHARADE : une sémantique cognitive pour les heuristiques d'apprentissage", *Proc. of the 8th international conf. of Experts Systems and their Applications*, Avignon, 1988, pp.567-586.
GODIN R., "Complexité de Structures de Treillis", *Ann. Sci. Math.*, Québec, vol.13, n°1, 1989, pp.19-38.
GODIN R., MISSAOUI R., ALAOUI H., "Learning Algorithms using a Galois Lattice Structure", *Proc. of the 1991 IEEE Int. Conf. on Tools for AI*, San José, CA, November 1991, pp.22-29.
GUÉNOCHE A., "Construction du treillis de Galois d'une relation binaire", *Math. Inf. Sci. Hum.*, n°109, 1991, pp.5-47.
KODRATOFF Y., "Recent Advances in Machine Learning", *The Intl. Journal of Pattern Recognition and Artificial Intelligence*, vol.7, 1993, pp.469-511.
LAGRANGE M.S., RENAUD M., MEPHU NGUIFO E., SALLANTIN J., "Apprentissage automatique et typologie. PLATA: une expérience d'acquisition de connaissances dans le domaine de la céramique archéologique", *Rapport de recherche LIRMM*, n°93-103, Décembre 1993.
LIQUIERE M., MEPHU NGUIFO E., "LEGAL : LEarning with GAlois Lattice", *5th JFA Proceedings*, 1990, pp.93-113.
MEPHU NGUIFO E., "Improvement and Control of Similarity-Based Decision for Knowledge Acquisition", *Proceedings of the first African Conference on Research in Computer Science*, Yaoundé (Cameroun), October 14-20 1992, pp.173-184, Ed. M. Tchuenté, INRIA.
MEPHU NGUIFO E., "Concevoir une abstraction à partir de ressemblances", *Thèse de Doctorat*, Université de Montpellier II (USTL), 11 Mai 1993.
MEPHU NGUIFO E., SALLANTIN J., "Prediction of primate splice junction gene sequences with a cooperative knowledge acquisition system", *Proc. of the 1st International Conference on Intelligent Systems for Molecular Biology*, Washington DC, July 7-9 1993, Eds. L. Hunter, D. Searls, and J. Shavlik, AAAI/MIT Press, Menlo Park CA.
MICHALSKI R.S., KODRATOFF Y., "Research in Machine Learning.: Recent progress, classification of methods, and future directions", *Machine Learning*: an AI approach, Kodratoff & Michalski eds, M.Kaufman, 1990, pp.1-30.

NOORDEWIER M.O., TOWELL G.G., SHAVLIK J.W., "Training Knowledge-Based Neural Networks to Recognize Genes in DNA sequences", *Advances in Neural Informat° Processing Systems*, vol.3, 1991, M. Kaufmann.
QUINLAN J.R., "Induction of Decisions Trees", *Machine Learning*, Mitchell & al. Eds, vol.1, 1986, pp.81-106.
QUINQUETON J., SALLANTIN J., "CALM: Constestation for Argumentative Learning Machine", *Machine Learning*, a Guide to current Research, T.M.Mitchell & al. Eds, 1986, pp.247-253.
TOWELL G.G., SHAVLIK J.W., "Interpretation of Artificial Neural Networks : Mapping Knowledge-based Neural Networks into Rules", *Advances in Neural Informat° Processing Systems*, vol.4, 1992, M Kaufman.
WILLE R., "Restructuring Lattice Theory : an Approach Based on Hierarchies of Concepts",, *in Ordered Sets* (ed. I. Rival), D. Reidel, Dordrecht, 1982, pp.445-470.
WILLE R., "Concept Lattices & Conceptual Knowledge Systems", *Comp. Math. App.*, vol.23, n°6-9, 1992, pp.493-515.