# THESE

présentée par

**Paul MUNTEANU**

pour obtenir le titre de

**DOCTEUR DE**

**L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

(arrêté ministériel du 30 Mars 1992)

Spécialité : *Sciences Cognitives*

# Extraction de connaissances dans les bases de données Parole : Apport de l'apprentissage symbolique

Date de la soutenance : 6 Décembre 1996

**Membres du jury :**

| | |
|---|---|
| Président | M. Augustin LUX |
| Rapporteurs | M. Yves KODRATOFF |
| | M. Djamel Abdelkader ZIGHED |
| Examinateurs | Mme. Geneviève CAELEN-HAUMONT |
| | M. Jean-Paul HATON |
| Directeur de thèse | M. Jean-François SERIGNAT |

Thèse préparée au sein de l'Institut de la Communication Parlée de Grenoble

# *Remerciements*

*Avant tout, je tiens à remercier mon directeur de thèse, Monsieur Jean-François SERIGNAT, Maître de Conférences à l'ENSERG. Je veux l'assurer de ma profonde reconnaissance pour les conseils qu'il m'a prodigués au long de mes recherches et de la rédaction de cette thèse et pour son soutien constant pendant tous ces années où j'ai eu le plaisir de travailler sous sa direction.*

*J'adresse mes remerciements à Monsieur Pierre ESCUDIER, Directeur de Recherche au CNRS, Directeur de l'Institut de la Communication Parlée et Responsable de la Formation Doctorale en Sciences Cognitives pour m'avoir accueilli dans son laboratoire. Grâce à son appui j'ai pu bénéficier du soutien matériel qui m'a permis de réaliser cette thèse dans d'excellentes conditions.*

*Je veux exprimer ma profonde reconnaissance à Monsieur Yves KODRATOFF, Directeur de Recherche au CNRS et à Monsieur Djamel Abdelkader ZIGHED, Professeur à l'Université Lyon 2, pour avoir accepté d'être rapporteurs et pour avoir contribué, par leurs remarques critiques et suggestions, à l'amélioration de ce mémoire.*

*Je tiens à remercier sincèrement Madame Geneviève CAELEN-HAUMONT, Chargée de Recherche au CNRS, pour sa disponibilité et pour l'apport substantiel de son expertise à mon travail, ainsi que pour sa participation au jury de cette thèse.*

*Mes plus vifs remerciements vont également à Monsieur Augustin LUX, Professeur à l'ENSIMAG, président de ce jury, pour l'intérêt qu'il a porté à mes travaux.*

*Je remercie aussi vivement Monsieur Jean-Paul HATON, Professeur à l'Université Nancy 1, pour avoir accepté de participer au jury.*

*Une grande partie des travaux présentés dans cette thèse ont été réalisés en collaboration étroite avec mon collègue Bertrand CAILLAUD. Je veux l'assurer ici de ma reconnaissance et de mon amitié.*

*Je tiens également à remercier mes collègues et compatriotes Adrian NEAGU, Costin HAGIU, Bogdan CRISTOFOR et Matei RIPEANU pour leurs contributions à ce travail et pour leur collaboration amicale.*

*Enfin merci à tous ceux, trop nombreux pour être cités ici, qui ont rendu mon séjour au sein de l'ICP fructueux et agréable.*

# Chapitre 1 :
# INTRODUCTION

## 1.1 Contexte de recherche : le projet SIDOC

Les travaux de recherche présentés dans cette thèse s'inscrivent dans le cadre du projet SIDOC-Parole (**S**ystème d'**I**ntégration de **Do**nnées et de **C**onnaissances sur la Parole). Ce projet "vise à fournir un environnement de gestion d'informations sur la parole pour appuyer la recherche scientifique" [Fernandez, 1989]. Il a pris naissance à l'I.C.P. dans le cadre des recherches à long terme pour la modélisation des données et des connaissances sur la parole et pour la réalisation des systèmes de gestion appropriés.

Des systèmes experts à règles de production sont développés à l'I.C.P. pour la reconnaissance automatique de la parole, le décodage acoustico-phonétique et, plus généralement, pour tout type d'étiquetage du signal acoustique : phonétique, prosodique, articulatoire, linguistique, etc., correspondant à différents niveaux de représentation de la connaissance tels que, par exemple, production, syntaxe ou sémantique. Une des principales difficultés que rencontrent les chercheurs dans le domaine de la reconnaissance et de la synthèse de parole est l'immense variabilité, inter- et même intra-locuteur, du signal de parole.

Le projet SIDOC vise à tirer avantage des bases de données parole pour proposer des critères d'appréciation objective, afin d'améliorer les performances des systèmes de reconnaissance et de synthèse ou de comparer entre elles les réalisations similaires. Il s'agit donc de développer des méthodologies et des outils pour le test et la mise au point de connaissances sur la parole et de les intégrer dans un environnement multimédia performant, centré autour d'un système de gestion de bases de données et de connaissances parole.

Mon D.E.A. a porté d'une part sur l'intégration dans le SIDOC des connaissances représentées sous forme de règles de production et d'autre part sur le développement des outils pour le test et la mise au point de ces connaissances par les experts. Les tests effectués sur un grand nombre de cas représentatifs et la possibilité de demander des justifications détaillées dans les situations d'échec devaient, selon notre hypothèse, faciliter la mise au point des bases de règles affectées par certaines erreurs.

Pour expérimenter les outils proposés, nous nous sommes orientés vers un problème concret et représentatif pour ce type de travail : le test et la mise au point d'un agent d'étiquetage prosodique dans le cadre du développement d'un système multi-agents de compréhension de la parole [Munteanu & Serignat, 1994a]. Le fonctionnement de l'agent était fondé sur des règles qui formalisaient les connaissances de type expert d'un chercheur en prosodie de notre laboratoire [Caelen-Haumont, 1991], [Caelen-Haumont, 1993].

Cette application pratique a fait ressortir certaines faiblesses de la méthodologie proposée. Même si un certain nombre d'imperfections des règles ont pu être mises en évidence et corrigées, nous nous sommes aperçus que la grande dimensionnalité de l'espace de recherche (les règles utilisaient 12 paramètres numériques) et des exemples considérés (>1000) empêchait l'expert d'explorer toutes les causes possibles de situations d'échec et de trouver des solutions en accord avec les exemples.

Il nous est donc apparu nécessaire de concevoir des outils capables d'automatiser dans une plus grande mesure ce travail d'exploration des bases de données pour la découverte et la mise au point des connaissances, en conservant toutefois la compréhensibilité des résultats produits et l'apport des connaissances et du raisonnement des experts humains.

Les travaux présentés dans cette thèse s'inscrivent dans ce nouvel axe de recherche émergé dans le cadre du projet SIDOC et font appel à des techniques d'apprentissage automatique symbolique pour la découverte et la mise au point de connaissances en parole.

## 1.2 Organisation du mémoire de thèse

Le premier chapitre de cette thèse se propose de réaliser un état de l'art des travaux dans le domaine de l'apprentissage automatique symbolique qui nous semblent intéressants pour la recherche en communication parlée.

Après une analyse succincte des particularités des problèmes d'apprentissage en parole, nous présentons, dans un deuxième chapitre, quelques propositions personnelles qui vont dans le sens d'une meilleure prise en compte de ces particularités (attributs numériques, bruit et représentations incomplètes, connaissances expertes et scientifiques) par les méthodes d'apprentissage symbolique.

L'évaluation expérimentale de l'apport de ces méthodes pour des problèmes pratiques de recherche en parole tels que l'acquisition de connaissances prosodiques pour l'amélioration de l'accès lexical et la discrimination des occlusives sourdes fera l'objet du troisième chapitre de ce mémoire.

Le quatrième chapitre présente avec plus de détails le projet SIDOC, la démarche que nous avons suivie pour le développement du système dans un environnement de gestion de bases de données orienté objets et l'intégration des fonctionnalités d'extraction automatique de connaissances dans ce contexte.

Nous concluons par quelques remarques sur le travail effectué et par la présentation des perspectives de ces recherches.

# Chapitre 2 :
# APPRENTISSAGE AUTOMATIQUE SYMBOLIQUE SUPERVISE

L'appellation "apprentissage automatique" est revendiquée actuellement par des approches issues de disciplines variées des sciences pour l'ingénieur : l'intelligence artificielle (I.A.) et l'algorithmique, la théorie des probabilités et la statistique, les réseaux de neurones et la logique floue. Par ailleurs, dans le cadre fédérateur des sciences cognitives, on attend dans ce domaine un apport bénéfique des recherches en sciences de l'homme et de la vie comme la psychologie, la didactique, l'éthologie ou la neurophysiologie.

Initialement très différentes entre elles et même parfois ouvertement opposées (I.A. symbolique vs. connexionisme, logique floue vs. probabilités), ces disciplines semblent s'accorder récemment sur un certain nombre de concepts et méthodes communes ou complémentaires (les "approches hybrides"). Il subsiste quand même un nombre important de particularités et il n'existe pas encore de théorie unificatrice complète. Compte tenu de nos objectifs de recherche, nous avons donc préféré présenter dans ce chapitre un aperçu, forcément incomplet face à l'extrême richesse du domaine, **des approches en apprentissage symbolique supervisé issues de l'intelligence artificielle**, tout en essayant de situer ces recherches par rapport aux autres disciplines qui étudient l'apprentissage automatique.

Ces recherches sont usuellement regroupées sous le nom d'apprentissage symbolique (*Machine Learning* en anglais). Leur naissance semble se situer dans les années soixante-dix et la consécration académique de la discipline est réalisée avec le "*First Machine Learning Workshop*" de 1980 et l'ouvrage édité à partir de cette manifestation : "*Machine Learning : An Artificial Intelligence Approach*" [Michalski *et al.,* 1983]. Parmi les personnalités qui ont contribué à la naissance et à l'affirmation de l'apprentissage symbolique en tant que discipline scientifique autonome, il faut citer R.S. Michalski, T.M. Mitchell et J.R. Quinlan.

Dans le reste de ce chapitre nous allons présenter très succinctement les modèles théoriques d'apprentissage les plus utilisés, la structure et les composantes d'un algorithme d'apprentissage automatique et quelques familles d'algorithmes, classés en fonction de la représentation de leur résultats.

## 2.1 La théorie de l'apprentissage calculatoire

L'étude théorique de l'apprentissage calculatoire (*computational machine learning theory*) constitue une branche relativement autonome dans l'ensemble des disciplines intervenant dans l'apprentissage automatique, qui vise à développer des modèles mathématiques d'apprentissage et à produire des résultats théoriques en s'appuyant sur ces modèles. Les résultats de cette discipline dépassent le cadre strict de l'apprentissage symbolique mais il nous a semblé intéressant de les citer pour leur caractère théorique et fondamental, complémentaire des approches plutôt empiriques qui seront présentées dans les sections suivantes de ce chapitre.

Dans cette section nous allons donner un très bref aperçu de deux modèles représentatifs utilisés dans ce domaine de recherche. Il faut préciser dès le début que les variantes de base de ces modèles, telles qu'elles seront présentées dans cette section, imposent des contraintes très fortes sur les problèmes d'apprentissage étudiés (concepts de classification binaire, exemples supposés sans bruit, etc.) qui sont rarement satisfaites dans la pratique. Ces contraintes sont affaiblies ou éliminées dans des travaux plus récents, aux dépens de la complexité des modèles mathématiques. Nous renvoyons le lecteur intéressé, pour plus d'information sur les extensions et sur les résultats obtenus dans le cadre de ces modèles, à des ouvrages spécialisés, comme [Anthony & Biggs, 1992].

Plusieurs définitions sont nécessaires pour fixer le vocabulaire :

**Définition 1** *Un exemple x est un n-uplet* $x = \left(x^1, x^2, \ldots, x^n\right) \in X$ *où X est l'espace des exemples.*

**Définition 2** *Un concept c est une fonction* $c \in C, c: X \rightarrow \{0,1\}$ *où C est la classe des concepts.*

**Définition 3** *Un exemple étiqueté par le concept c est une paire* $(x,c(x))$, $x\in X$, $c\in C$. *x est un exemple positif du concept c si* $c(x)=1$. *Sinon, x est un exemple négatif du concept c.*

La tâche d'un algorithme d'apprentissage consiste dans la découverte (ou l'approximation) du concept inconnu *c*, en s'appuyant sur des exemples étiquetés par ce concept. Les différents modèles théoriques précisent des contraintes sur le système d'apprentissage, sur les classes de concepts à apprendre, sur la façon dont le système d'apprentissage accède aux exemples et sur les performances recherchées.

### 2.1.1 Apprentissage en ligne avec nombre d'erreurs borné

Dans ce modèle [Littlestone, 1988], l'algorithme d'apprentissage accède à un seul exemple $x \in T \subseteq X$ à la fois, effectue une prédiction de l'étiquette $c(x)$ et reçoit ensuite l'étiquette correcte de l'exemple considéré. Les exemples peuvent être arbitrairement choisis dans $X$, éventuellement même par un adversaire malicieux, et sont supposés sans bruit. Il s'agit donc de contraintes extrêmement fortes, qui feront l'objet des principales critiques sur ce modèle. La qualité de l'apprentissage est évaluée à partir du nombre total d'erreurs de prédiction effectuées par l'algorithme sur l'ensemble d'exemples considérés.

**Définition 4** *Soit $E(A,c,T)$ le nombre d'erreurs effectuées par l'algorithme $A$ en apprenant le concept $c$ sur l'ensemble d'apprentissage $T$. Un algorithme d'apprentissage en ligne $A*$ est optimal pour une classe de concepts $C$, dans ce modèle, si le nombre total d'erreurs effectuées dans la situation la plus défavorable est minimal par rapport à tous les autres algorithmes possibles. Soit $c*$ et $T*$ le concept et l'ensemble d'exemples d'apprentissage qui constituent la situation la plus défavorable pour l'algorithme $A*$. Alors, par défintion :*

$$E(A*,c*,T*) = \min_{A} \max_{c \in C} \max_{T \subseteq X} E(A,c,T) = Opt(C)$$

Les résultats recherchés dans le cadre de ce modèle concernent l'obtention des limites supérieures de plus en plus basses pour $Opt(C)$, dans le cas général ou pour certaines classes particulières de concepts $C$, et la conception d'algorithmes efficaces pour approcher ces limites. Quelques limites générales [Rivest, 1994] pour $Opt(C)$ sont évidentes :

- $Opt(C) \leq 2^n$, où $n$ est la dimension des exemples (le nombre d'attributs binaires qui les décrivent). En effet, le nombre maximal d'erreurs effectuées par un algorithme ne peut pas dépasser le nombre maximal d'exemples.

- $Opt(C) \leq |C| - 1$, où $|C|$ dénote le nombre de concepts. Un algorithme peut accorder un ordre arbitraire aux concepts, prédire toujours en accord avec le premier concept restant, tant qu'aucune erreur n'est rencontrée, et enlever le concept courant de la liste après sa première erreur rencontrée. Il en résulte que, dans la situation la plus défavorable, où le concept recherché est le dernier de la liste, l'algorithme effectue une erreur pour chacun des autres concepts (en nombre de $|C|$ - 1).

- $Opt(C) \leq \log_2(|C|)$. Un algorithme peut prédire en accord avec la majorité des concepts restants et enlever de la liste des concepts possibles, pour chaque erreur rencontrée, tous les concepts qui ont fourni cette prédiction erronée. La limite mentionnée peut être obtenue à partir de l'observation que, après chaque prédiction erronée, plus de la moitié des concepts sont enlevés de la liste.

Il est intéressant de noter qu'on peut aussi obtenir des limites inférieures pour ce critère. Une des méthodes pour trouver une telle limite passe par l'estimation de la "complexité" d'une classe de concepts :

**Définition 5** *La dimension Vapnik-Chervonenkis d'une classe de concepts C, ou $VC_{DIM}(C)$, est le plus grand entier d tel qu'il existe d exemples distincts $x_1$, $x_2$, ..., $x_d$ qui peuvent être étiquetés de $2^d$ façons différentes par des concepts de C.*

Par exemple, soit $L$ la classe des concepts séparables par une ligne droite dans le plan. La $VC_{DIM}(L)$=3, parce qu'on peut toujours trouver une ligne séparatrice qui étiquette 1, 2 ou 3 points de toutes les façons possibles mais il existe des configurations de 4 points étiquetés qui ne peuvent pas être séparés en classes par une seule ligne (les coins d'un carré, étiquetés avec une classe différente sur chaque diagonale).

Il est alors évident que $Opt(C) \geq VC_{DIM}(C)$. En effet, par la définition de la $VC_{DIM}(C)$, un adversaire malicieux peut fournir à tout algorithme d'apprentissage $d$ exemples et contredire toujours ses prédictions en étant assuré qu'il existe un concept $c \in C$ en désaccord avec ces prédictions.

Supposons, par exemple, qu'il s'agit de l'apprentissage des concepts séparables par une ligne droite dans un plan (la classe de concepts $L$ définie plus haut). Comme la $VC_{DIM}(L)$=3, nous pouvons fournir, un par un, trois exemples différents à l'algorithme d'apprentissage et refuser systématiquement chacune de ses prédictions sur l'étiquette de l'exemple. Comme, par la définition de la $VC_{DIM}$, il existe des ensembles de trois exemples qui peuvent être étiquetés de toutes les façons possibles par les concepts de $L$, en choisissant un ensemble d'exemples approprié (les coins d'un triangle), nous sommes assurés que la configuration résultante (qui associe à chacun des trois exemples l'étiquette complémentaire à celle prédite par l'algorithme) est compatible avec un concept séparable par une ligne droite dans le plan.

### 2.1.2 Apprentissage probablement approximativement correct (PAC)

L'adéquation du modèle précédent aux problèmes réels d'apprentissage peut être critiquée de plusieurs façons. Premièrement, dans la pratique, la phase d'apprentissage précède d'habitude l'utilisation effective pour la prédiction. Il serait donc intéressant de pouvoir estimer les performances de l'algorithme après avoir traité un certain nombre d'exemples.

Deuxièmement, accepter que les exemples d'apprentissage soient choisis d'une façon arbitraire, éventuellement même par un adversaire malicieux, constitue une contrainte très forte et qui peut ne pas s'avérer indispensable dans la pratique.

Ces motivations sont à la base d'un autre modèle : l'apprentissage probablement approximativement correct (*probably approximately correct learning*), ou PAC [Valiant, 1984]. Ce modèle est probablement le plus connu et le plus utilisé dans les études théoriques de l'apprentissage automatique.

Dans l'apprentissage PAC, l'algorithme accède aux exemples étiquetés par l'intermédiaire d'un oracle $EX(c,D)$. L'oracle fournit les exemples d'une façon aléatoire et indépendante, en accord avec une distribution de probabilité stationnaire $D$. Les exemples ne sont pas corrompus par du bruit.

L'algorithme d'apprentissage utilise l'oracle pour produire une hypothèse $h$ qui approxime le concept à apprendre, $c$. La qualité de $h$ est appréciée par rapport à la même distribution de probabilité $D$ utilisée pour l'apprentissage.

**Définition 6** *Le taux d'erreur d'une hypothèse h par rapport à une distribution D et à un concept à apprendre c est* $Erreur(h) = \sum_{x \in h\Delta c} D(x)$ *où* $h\Delta c$ *représente l'ensemble d'exemples pour lesquels h et c sont en désaccord.*

**Définition 7** *Une classe de concepts C peut être apprise PAC s'il existe un algorithme d'apprentissage A tel que pour tout concept* $c \in C$*, toute distribution D sur X, tout* $\varepsilon \in (0,1)$ *et tout* $\delta \in (0,1)$*, l'algorithme A avec* $\varepsilon$ *et* $\delta$ *en entrée et accédant à EX(c,D) produit avec une probabilité au moins égale à* $1-\delta$ *une hypothèse h avec une* $Erreur(h) \leq \varepsilon$*. L'algorithme A doit* *présenter une complexité polynomiale par rapport à 1/ε, 1/δ, la taille d'un exemple (le nombre d'attributs par exemple) et la taille (la complexité syntactique) du concept à apprendre.*

Plus simplement, cette définition demande que l'algorithme d'apprentissage traite les exemples efficacement et produise, avec une grande probabilité, une bonne approximation du concept à apprendre. Pour illustrer cette définition, nous allons prouver que la classe des conjonctions peut être apprise PAC, en nous appuyant sur la démonstration présentée dans [Rivest, 1994]. Les démonstrations de ce type de résultat pour des classes de concepts plus complexes suivent d'habitude une démarche similaire.

Soit *X* l'espace des exemples décrits par des attributs binaires de dimension *n* et *C* la classe de toutes les conjonctions possibles entre certains de ces attributs et de leurs négations (par exemple $x^1 \wedge \overline{x^3} \wedge x^5$). Nous allons appeler *littéral* chaque terme (correspondant à un attribut ou à sa négation) d'une telle conjonction. Soit *A* un algorithme qui part d'une hypothèse initiale *h* contenant la conjonction de tous les attributs et de leurs négations et qui, pour chaque exemple positif d'apprentissage, enlève de *h* tous les littéraux qui sont faux dans l'exemple respectif.

Soit *p*(*z*) la probabilité que l'oracle fournisse un exemple positif dans lequel le littéral *z* est faux (qui est donc égale à la probabilité que *z* soit enlevé de *h* par un exemple d'apprentissage arbitraire). Il est donc évident que *p*(*z*)=0 si *z* figure dans le concept à apprendre, parce que l'oracle fournit les exemples en accord avec ce concept. Si *z* figure dans l'hypothèse finale (s'il n'a pas été enlevé), *p*(*z*) correspond également à la probabilité que cette hypothèse va faire une erreur à cause de ce littéral. Il en résulte, donc, que $Erreur(h) \le \sum_{z \in h} p(z)$. Alors, si $p(z) \le \frac{\varepsilon}{2n}$, $Erreur(h) \le \varepsilon$ (parce que *h* peut contenir au maximum 2*n* littéraux). Il faut donc trouver le nombre minimal d'exemples d'apprentissage *m* qui nous assure que, avec une probabilité supérieure ou égale à 1-*δ*, tout littéral *z* qui reste dans *h* après l'apprentissage aura une $p(z) \le \frac{\varepsilon}{2n}$. Dans une formulation équivalente, il faut donc que la probabilité qu'il reste en *h* au moins un littéral ayant $p(z) > \frac{\varepsilon}{2n}$ soit inférieure ou égale à *δ*.

Soit $z$ un littéral avec $p(z) > \frac{\varepsilon}{2n}$. La probabilité que $z$ reste dans $h$ après le traitement du premier exemple d'apprentissage est, par la définition de $p(z)$, $1 - p(z) < 1 - \frac{\varepsilon}{2n}$. Après le traitement de $m$ exemples d'apprentissage, cette probabilité devient : $(1 - p(z))^m < \left(1 - \frac{\varepsilon}{2n}\right)^m$. La probabilité qu'au moins un tel littéral, avec $p(z) > \frac{\varepsilon}{2n}$, reste dans $h$ est donc limitée par $2n\left(1 - \frac{\varepsilon}{2n}\right)^m \leq 2ne^{-\frac{m\varepsilon}{2n}}$, où nous avons utilisé le fait qu'il existe au plus $2n$ littéraux dans $h$ et l'inégalité bien connue $1 - x \leq e^{-x}$ pour $0 \leq x \leq 1$.

Il nous reste donc à choisir un $m$ tel que $2ne^{-\frac{m\varepsilon}{2n}} \leq \delta \Rightarrow m \geq \frac{2n}{\varepsilon}\left(\ln(2n) + \ln\left(\frac{1}{\delta}\right)\right)$. Nous avons donc démontré, d'une manière constructive, qu'il existe un algorithme *A* qui produit, avec une probabilité $\geq$ 1-$\delta$, une hypothèse $h$ avec une *Erreur*($h$)$\leq \varepsilon$, après avoir traité un nombre $m$ d'exemples d'apprentissage, polynomial en $1/\varepsilon$, $1/\delta$ , la taille d'un exemple ($n$), et la taille du concept à apprendre (limitée par $2n$). La classe des conjonctions peut donc être apprise PAC.

Un grand nombre de résultats théoriques sur des classes de concepts qui peuvent ou ne peuvent pas être apprises PAC ont été obtenus [Valiant, 1984], [Pitt & Valiant, 1988], [Blumer et al., 1989]. Il existe quand même des classes importantes du point de vue pratique pour lesquelles il n'existe pas encore, à notre connaissance, des résultats définitifs : la classe de toutes les fonctions DNF (forme normale disjonctive), la classe de toutes les fonctions CNF (forme normale conjonctive) ou la classe de tous les arbres de décision.

## 2.2 Algorithmes d'apprentissage automatique

En dépit de l'intérêt certain de cette entreprise de formalisation, la théorie de l'apprentissage calculatoire est encore loin de constituer un fondement mathématique unitaire et autosuffisant pour toutes les recherches en apprentissage automatique. Comme dans la plupart des branches de l'Intelligence Artificielle, les avancées dans ce domaine sont fondées dans une mesure importante sur une approche plutôt empirique et expérimentale.

Cette section est consacrée à l'analyse de l'architecture et des composants des algorithmes d'apprentissage. Nous allons nous occuper plus particulièrement des algorithmes d'apprentissage supervisé mais une bonne partie de ces observations s'appliquent aussi à d'autres catégories d'apprentissage.

### 2.2.1 Structure d'un algorithme d'apprentissage

A première vue, les algorithmes d'apprentissage automatique présentent une diversité impressionnante. Le but de cette section est d'analyser leur structure intime pour mettre en évidence leurs caractéristiques communes et leurs différences et, donc, de proposer une taxonomie des algorithmes d'apprentissage automatique.

Les algorithmes d'apprentissage effectuent la recherche d'une "bonne" représentation des exemples fournis. L'appréciation de la qualité de la représentation recherchée est fondée d'habitude sur son pouvoir prédictif mais il subsiste des différences importantes en fonction du modèle théorique d'apprentissage adopté, comme nous l'avons vu dans la section 2.1.

Pour certaines classes restreintes de problèmes il existe des solutions analytiques rigoureuses, d'une complexité raisonnable et qui garantissent l'obtention d'une solution optimale. C'est le cas, par exemple, de l'approximation d'une fonction linéaire par la méthode des moindres carrés, si on estime la qualité des résultats à partir de l'erreur quadratique moyenne sur l'ensemble d'apprentissage. Dans d'autres situations les dimensions de l'espace des représentations possibles sont suffisamment limitées pour qu'une recherche exhaustive de la meilleure solution puisse être envisagée.

Pour la plupart des problèmes d'apprentissage, cependant, des solutions qui garantiraient l'obtention d'une représentation optimale ne sont pas connues ou présentent une complexité de calcul inacceptable. C'est le champ d'action privilégié de l'Intelligence Artificielle, qui propose des stratégies de recherche fondées sur l'utilisation des contraintes et des heuristiques pour résoudre ce genre de problèmes.

| |
|---|
| **Initialisation** → une **représentation** interne initiale<br><br>**Recherche** d'une représentation améliorée :<br><br>Tant que non *condition d'arrêt*<br><br>Choix du (des) **opérateur(s)** à appliquer sur la représentation courante en vue de l'amélioration de la **fonction d'évaluation**<br><br>Obtention d'une (ou plusieurs) représentation(s) par l'application du (des) opérateur(s) choisi(s)<br><br>Fin tant que<br><br>*Post-traitement de la représentation obtenue*<br><br>Transformation de la représentation interne finale dans un format externe (optionnel) |

**Figure 2-1** Structure d'un algorithme d'apprentissage

La figure 2-1 présente la structure d'un algorithme d'apprentissage supervisé générique et met en évidence les axes de comparaison que nous allons détailler dans les sections suivantes. L'apprentissage automatique est généralement un processus itératif, qui part d'une **représentation interne initiale** et la modifie, en appliquant des **opérateurs** spécifiques et en suivant une certaine **stratégie de recherche**, pour améliorer une certaine **fonction d'évaluation** de la qualité de la représentation.

### 2.2.2 Représentation des résultats et opérateurs de transformation

Un axe fondamental de classification de ces algorithmes est constitué par le formalisme utilisé dans la représentation, interne et externe, des résultats de l'apprentissage. Les opérateurs utilisés pour l'amélioration de la représentation interne du phénomène étudié sont étroitement liés à ces formalismes.

Un aspect important dans le choix du formalisme de représentation est constitué par la puissance expressive de celui-ci. Il est évident qu'un formalisme plus puissant peut accroître les chances de trouver une bonne approximation du concept à apprendre dans l'espace des représentations généré mais il peut, en même temps, rendre la recherche beaucoup plus coûteuse.

En fonction de l'intelligibilité, pour les humains, de ces résultats nous allons considérer trois grandes classes d'algorithmes d'apprentissage :

#### 2.2.2.1 Les approches "boîte noire"

Nous regroupons sous cette appellation tous les algorithmes d'apprentissage dont les résultats ne sont pas destinés à la compréhension et à l'utilisation directe par les humains. Les représentants les plus notables de cette famille sont issus de l'approche connexionniste qui utilise généralement la structure et la matrice des poids d'un réseau de neurones pour représenter les résultats de l'apprentissage. Les opérateurs les plus utilisés dans ces approches affectent les valeurs des poids (comme dans le célèbre algorithme de rétro-propagation du gradient [Rumelhart et al., 1986]) ou la structure même du réseau ([Reilly et al., 1982], par exemple).

D'autres approches, comme les chaînes de Markov, peuvent également être citées dans cette catégorie. Les chaînes de Markov cachées utilisent les états et les probabilités de transition entre les états et d'émission de symboles pour représenter leurs résultats. L'apprentissage est fondé sur des opérateurs d'estimation itérative de ces paramètres.

La présentation de ces approches dépasse les objectifs de cette thèse (l'extraction de connaissances directement intelligibles pour les humains dans les bases de données parole). Il faut cependant mentionner que les chaînes de Markov cachées et les réseaux de neurones sont utilisés depuis longtemps, avec des résultats remarquables, dans la recherche en parole. Un état de l'art récent et des références vers ces travaux sont disponibles dans [Cole, 1995].

#### 2.2.2.2 Les approches symboliques

Cette appellation regroupe généralement les algorithmes d'apprentissage issus de l'Intelligence Artificielle classique. Ils s'appuient donc sur les formalismes de représentation des connaissances utilisés dans ce domaine et mettent l'accent sur une compréhensibilité et une utilisabilité directe par les humains des connaissances produites. Cette approche se propose aussi de constituer une alternative ou un complément à l'acquisition de connaissances pour les

Systèmes Experts par rapport aux techniques d'élicitation de connaissances auprès d'un expert humain. Elle fera l'objet d'une présentation détaillée dans les sections suivantes de ce chapitre.

On peut classer les algorithmes d'apprentissage symbolique en fonction du paradigme et de la puissance expressive du formalisme utilisé pour la représentation des connaissances produites. Les approches classiques sont fondées sur des :

- **arbres de décision** : CART [Breiman et al., 1984], ID3 [Quinlan, 1986], C4.5 [Quinlan, 1993], par exemple. L'opérateur de transformation consiste généralement dans le remplacement d'une feuille terminale par un sous-arbre.

  Une extension de ce formalisme est constitué par les **graphes de décision** non-arborescents (SIPINA [Zighed, 1992]) qui utilisent en plus des opérateurs de regroupement des branches.

- **règles**

  - **propositionnelles** ou **attributs-valeurs** : AQ15 [Michalski et al., 1986], CN2 [Clark & Niblett, 1989], C4.5+C4.5*rules* [Quinlan, 1993], par exemple.

  - **prédicatives** : FOIL [Quinlan, 1990], GOLEM [Muggleton & Feng, 1992], par exemple.

  Les opérateurs de transformation des règles, utilisés au cours de la recherche, consistent dans la généralisation ou la spécialisation des règles existantes (par la suppression ou l'ajout de conditions), la création ou la suppression de règles, la construction de nouveaux traits (*constructive induction*).

#### 2.2.2.3 Les approches hybrides

Plus récemment, on observe un intérêt croissant dans les recherches qui essaient d'utiliser conjointement plusieurs paradigmes de modélisation. L'intégration des approches symboliques et connexionistes a fait, par exemple, l'objet de plusieurs colloques spécialisés [Sun *et al.*, 1992], [Sun & Alexandre, 1995], à l'occasion des grandes conférences d'Intelligence Artificielle, et de plusieurs numéros spéciaux de certaines revues internationales [Hendler, 1989], [Hinton, 1990].

Ces approches se focalisent en grande partie sur la conception d'architectures (modulaires ou monolithiques) qui combinent différents formalismes de représentation dans un même système ou sur les possibilités de translation entre ces formalismes. Un des problèmes importants qui se posent lors de cette hybridation concerne la conception des méthodes d'apprentissage à utiliser dans ces nouveaux systèmes.

A titre d'exemple, on peut citer des approches qui :

- intègrent étroitement, dans un modèle unique, des idées issues de l'apprentissage symbolique et connexioniste (voir [Utgoff, 1988] ou [Sanger, 1991] pour des combinaisons entre les arbres de décision et les perceptrons);

- utilisent des réseaux de neurones en tant que représentation interne des résultats de l'apprentissage et les règles de production (éventuellement pourvues des coefficients de certitude) pour l'importation ou l'exportation des connaissances ([Towell & Shavlik, 1991], [Lacher *et al.*, 1992], [Mahooney & Mooney, 1993]).

D'autres recherches sur les représentations hybrides concernent également l'intégration des réseaux de neurones avec la logique floue, des règles symboliques avec des représentations génétiques et beaucoup d'autres combinaisons entre ces différents formalismes.

### 2.2.3 Stratégies de recherche

Les stratégies de recherche utilisées en apprentissage automatique s'apparentent à celles développées pour les parcours de graphe. Elles peuvent être irrévocables ou par tentatives et retour arrière (*backtracking*), appliquer un seul opérateur à la fois (*hill-climber*) ou poursuivre en parallèle plusieurs pistes (*beam-search*), etc.

D'autres stratégies plus sophistiquées utilisent des algorithmes génétiques ou des techniques de recuit simulé, des contraintes du domaine fournies par les experts ou une interaction directe avec l'utilisateur au cours de la recherche.

### 2.2.4 Fonction d'évaluation

La fonction d'évaluation est utilisée au cours de la recherche pour apprécier la qualité des représentations candidates. Son importance est donc capitale pour les performances de l'apprentissage. Malheureusement, aucune fonction d'évaluation n'a encore prouvé sa totale suprématie par rapport aux autres et les recherches se poursuivent dans plusieurs directions.

Les fonctions d'évaluation réalisent un compromis entre différents critères, souvent contradictoires, comme la maximisation de la pureté des exemples couverts et du nombre de ces exemples ou la minimisation de la complexité de la représentation. Elles sont fondées d'habitude sur des résultats issus de la statistique ou de la théorie de l'information. Des représentants des familles de fonctions d'évaluation les plus utilisées dans l'apprentissage symbolique sont détaillés dans la section 2.3. Dans la section 3.3 nous allons présenter également un cadre original d'étude de ces fonctions du point de vue de l'induction des arbres de décision.

En fonction du formalisme utilisé pour la représentation des résultats de l'apprentissage et de la stratégie de recherche de la représentation optimale, il subsiste des différences importantes entre les objectifs des fonctions d'évaluation correspondantes. Il existe également des centres d'intérêt commun dans ces recherches, parmi lesquels l'estimation locale des probabilités des classes à partir d'un sous-ensemble d'exemples d'apprentissage. Dans les paragraphes qui suivent nous allons donner un court aperçu de cette problématique.

Pour simplifier les notations et faciliter les représentations graphiques, considérons pour l'instant un problème de classification binaire. Soit $p$ et $n$ les nombres d'exemples positifs et négatifs du concept à apprendre. Il s'agit d'estimer la fréquence d'apparition d'exemples positifs à partir de ces effectifs. La densité de probabilité de cette fréquence est donnée par la formule suivante, attribuée à Bayes et présentée, par exemple, dans [Jaynes, 1994] :

$$p(f) = \frac{(p+n+1)!}{p! \times n!} f^{p} \times (1-f)^{n}$$

Cette densité de probabilité est représentée graphiquement dans la figure 2-2. Plus le nombre d'exemples disponibles est important, plus la probabilité sera concentrée dans un intervalle de fréquences très étroit et tend, en limite, vers un pic de Dirac. Réciproquement, pour des effectifs d'exemples réduits, la courbe a tendance à s'aplatir, jusqu'à devenir constante dans la situation extrême où aucun exemple n'est disponible.

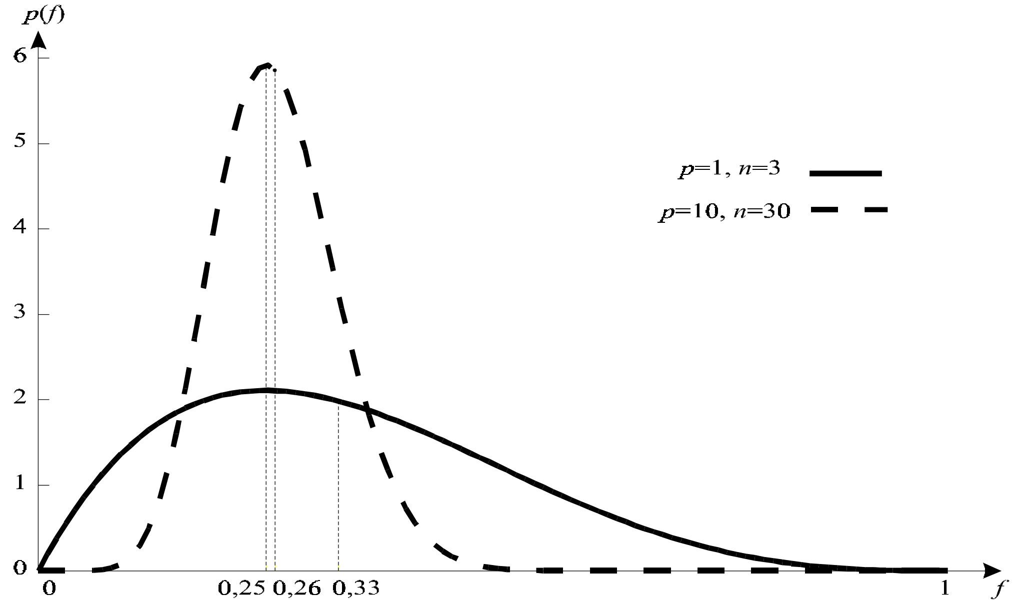



**Figure 2-2** Densités de probabilités et estimations ponctuelles des fréquences d'exemples positifs pour des effectifs différents

A partir de cette formule nous pouvons avoir deux démarches différentes pour calculer une estimation de la fréquence $f$. Premièrement, nous pouvons choisir la valeur qui réalise le maximum de cette densité de probabilité :

$$\frac{dp(f)}{df} = 0 \Rightarrow \hat{f} = \frac{p}{p+n}$$

Cette estimation, dont l'interprétation est très intuitive, nous indique la valeur ponctuelle la plus probable de la fréquence $f$. Elle peut être généralisée naturellement à l'estimation des fréquences dans le cas de plusieurs classes : $\hat{f}_i = \frac{N_i}{N}$, où $N$ dénote le nombre total d'exemples et $N_i$ représente le nombre d'exemples disponibles appartenant à la classe $i$. Un grand nombre d'algorithmes d'apprentissage intègrent cette estimation dans leurs fonctions d'évaluation : CART [Breiman *et al.*, 1984], ID3 [Quinlan, 1986], C4.5 [Quinlan, 1993] (voir section 2.3.1). On peut reprocher à toutes ces fonctions d'évaluation qu'elles ne tiennent pas compte de l'aplatissement de la courbe de $p(f)$ au fur et à mesure que le nombre d'exemples disponibles décroît et qu'elles vont réaliser la même estimation ($\hat{f} = 0{,}25$) à partir des distributions aussi différentes, intuitivement, que $p$=1, $n$=3 par rapport à $p$=10, $n$=30. Dans les algorithmes qui utilisent des fonctions d'évaluations fondées sur ce type d'estimation, l'importance des effectifs disponibles est généralement reconsidérée au cours de l'élagage (voir section 2.5.1).

Une autre démarche possible pour la résolution de ce problème consiste dans la minimisation de la différence attendue entre l'estimation utilisée $\hat{f}$ et la fréquence réelle $f$ à estimer. Si on adopte un critère d'erreur quadratique :

$$\min_{\hat{f}}\left(E\left(\left(f-\hat{f}\right)^2\right)\right)=\min_{\hat{f}}\left(\int_0^1\left(f-\hat{f}\right)^2\times p(f)df\right)\Rightarrow\frac{d\left(\int_0^1\left(f-\hat{f}\right)^2\times p(f)df\right)}{d\hat{f}}=0\Rightarrow\hat{f}=\frac{p+1}{p+n+2}$$

On peut observer facilement que, à la différence de la précédente, cette estimation dépend du nombre d'exemples disponibles. En effet, pour reprendre l'exemple précédent, $\hat{f}(1,3)=0,33$ et $\hat{f}(10,30)=0,26$. Elle peut être facilement généralisée pour un nombre arbitraire de classes *k*. En utilisant les mêmes notations que précédemment, l'estimation de la fréquence d'une classe *i* devient : $\hat{f}_i=\frac{N_i+1}{N+k}$. Cette formule est plus connue sous le nom de la loi de succession de Laplace (voir [Jaynes, 1994], par exemple, pour une discussion plus approfondie de cet estimateur, du point de vue de la théorie des probabilités).

L'utilisation de cette estimation de probabilité dans le domaine de la classification automatique est assez ancienne (voir [Gaillat, 1970] pour une application à la reconnaissance des caractères manuscrits, par exemple). Plus récemment, elle a été appliquée avec succès pour l'évaluation des règles dans CN2 [Clark & Boswell, 1991] (voir section 2.3.2). Une extension de cette formule, qui prend en compte les probabilités *a priori* des classes, (la *m*-estimation de probabilité) est proposée dans [Cestnik & Bratko, 1991] et sert à estimer l'erreur pour l'élagage des arbres de décision. Elle est également intégrée dans la fonction d'évaluation fondée sur l'entropie de l'algorithme d'induction de graphes de décision SIPINA [Zighed & Rakotomalala, 1996], sous une forme généralisée : $\hat{f}_i=\frac{N_i+\lambda}{N+k\times\lambda}$, où $\lambda$ est un paramètre qui contrôle la croissance du graphe et peut être utilisé pour adapter le comportement de l'algorithme aux besoins de l'utilisateur.

### 2.2.5 Conditions initiales

En fonction de la représentation initiale, qui constitue le point de départ de la recherche, on peut classer les algorithmes d'apprentissage dans trois catégories :

- **représentation indépendante du domaine** (irrévocable ou par essais successifs)
  - **choisie d'une façon aléatoire** : la matrice des poids dans certains algorithmes connexionnistes, la population initiale dans certaines approches génétiques, par exemple
  - **très générale** : une grande partie des algorithmes symboliques, comme ID3 [Quinlan, 1986], C4.5 [Quinlan, 1993], CN2 [Clark & Niblett, 1989], FOIL [Quinlan, 1990] (l'approche descendante ou *top-down*)
  - **très spécialisée** : DBLEARN [Cai et al., 1991], GOLEM [Muggleton & Feng, 1992], etc. (l'approche ascendante ou *bottom-up*)
- **résultats d'un apprentissage précédent** : ID5R [Utgoff, 1989], AQ15 [Michalski et al., 1986] (en option), etc.(algorithmes d'apprentissage incrémental)
- **connaissances d'un expert** : EITHER [Ourston & Mooney, 1990], KBANN [Towell & Shavlik, 1991], etc. (algorithmes de révision de théorie).

## 2.3 Induction automatique de connaissances

Cette section présente quelques catégories représentatives d'algorithmes d'induction de connaissances, classés en fonction du formalisme utilisé pour la représentation des connaissances apprises. Nous présentons deux catégories d'algorithmes fondées sur une logique des propositions (inductions d'arbres de décision et de règles attribut-valeur) et une troisième catégorie fondée sur la logique des prédicats (programmation logique inductive).

La présentation de chaque catégorie s'appuie sur des exemples d'algorithmes représentatifs (notamment ceux qui seront le plus souvent référencés dans les chapitres suivants de cette thèse), en suivant les axes d'analyse introduits dans la section précédente. Ces descriptions d'algorithmes sont forcément incomplètes et très schématiques. Le lecteur est renvoyé à des textes de référence pour plus de détails sur chacun de ces algorithmes.

Pour donner un aperçu de la variété des formalismes utilisés par ces algorithmes pour la représentation de leurs résultats nous présentons, dans l'Annexe B, quelques exemples de classificateurs produits sur la base d'exemples d'apprentissage *breast-w* de l'UC-Irvine.

### 2.3.1 Induction d'arbres de décision

Les arbres de décision constituent un des premiers formalismes de représentation de connaissances qui ont été utilisés avec succès dans l'apprentissage automatique. Il nous semble très important de mentionner que les représentations par arbres et graphes de décision et l'utilisation de fonctions d'évaluation fondées sur l'entropie (le gain informationnel) ont été proposées depuis longtemps en France, par des recherches sur les pseudo-questionnaires et les processus d'interrogation non arborescents, issues du domaine de la reconnaissance de formes. On peut citer dans ce sens les recherches effectuées à Grenoble [Terrenoire, 1970], fondées sur les travaux qui seront publiés plus tard dans [Picard, 1972], et qui ont été continuées ensuite par plusieurs thèses soutenues à Lyon, dont [Tounissoux, 1980] et [Zighed, 1985].

Nous avons choisi C4.5 [Quinlan, 1993] pour illustrer cette approche, parce que cet algorithme est très connu dans le domaine de l'apprentissage symbolique et sera utilisé dans les comparaisons théoriques et expérimentales avec les algorithmes que nous proposons dans le chapitre 3.

#### 2.3.1.1 Représentation de connaissances

Les arbres de décision constituent un formalisme de représentation de connaissances fondé sur une logique propositionnelle ou attributs-valeurs. Chaque noeud d'un arbre de décision correspond à un test effectué sur la valeur d'un attribut qui figure dans la description des exemples. Les branches partant d'un noeud correspondent à chacune des valeurs prises par l'expression attachée au noeud. Les feuilles terminales correspondent aux classes à apprendre (en nombre fini).

Par défaut, C4.5 crée une branche pour chacune des valeurs possibles d'un attribut symbolique figurant dans un noeud (figure 2-3). Le regroupement de plusieurs valeurs sur la même branche est proposé en option mais demande une recherche plus coûteuse. Dans IDF (section 3.3), nous allons utiliser une autre catégorie de tests pour les attributs symboliques : des tests binaires, qui comparent la valeur de l'attribut à une valeur particulière, les deux branches résultantes correspondant à des valeurs égales ou différentes.

Pour les attributs numériques, C4.5 produit deux branches correspondant aux tests d'inégalité $A_i \leq N$ et $A_i > N$ où $N$ est un seuil déterminé par l'algorithme (figure 2-3). Ce processus, qui divise le domaine de variation continu d'un attribut numérique en deux (ou, généralement, en plusieurs) intervalles est connu sous le nom de discrétisation. Il permet d'attribuer une étiquette unique à toutes les valeurs appartenant à chacun de ces intervalles et de les traiter ensuite de la même manière que les valeurs symboliques. Plusieurs méthodes de discrétisation des attributs numériques dans des intervalles multiples ont été proposées (ChiMerge [Kerber, 1992], MDLPC [Fayyad & Irani, 1993], FUSINTER [Zighed *et al.*, 1996], par exemple). Les différentes méthodes de discrétisation diffèrent entre elles par les fonctions d'évaluation des partitions résultantes, par la décision de prendre en compte ou pas la classe des exemples au cours de la discrétisation et par la place qu'occupe ce traitement dans l'algorithme d'apprentissage (discrétisation unique, qui précède l'apprentissage ou discrétisation dynamique, répétée à chaque itération). C4.5 réalise une discrétisation binaire et dynamique des attributs numériques, en utilisant une fonction d'évaluation fondée sur l'entropie, qui sera détaillée dans la section 2.3.1.3.

Il faut également mentionner que, si C4.5 crée uniquement des noeuds qui testent les valeurs d'un seul attribut, d'autres algorithmes d'induction d'arbres de décision (*linear machine* [Utgoff & Brodley, 1990], OC1 [Heath et al., 1993], etc.) peuvent utiliser aussi des expressions impliquant plusieurs traits simultanément (des combinaisons linéaires des attributs numériques, par exemple).

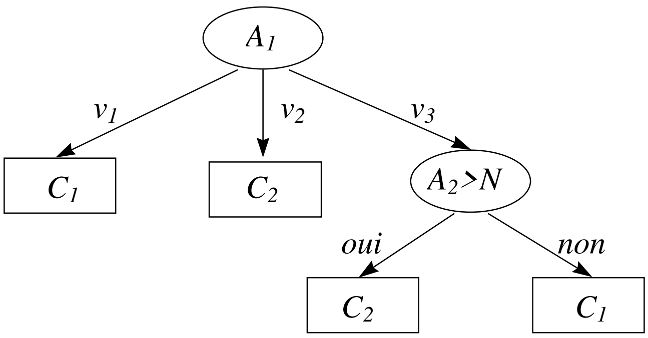



**Figure 2-3** Tests utilisés par C4.5

### 2.3.1.2 Stratégie de recherche

La plupart des algorithmes d'induction d'arbres de décision effectuent une recherche heuristique de type descendant (*top-down*) et irrévocable. La construction d'un arbre de décision commence par le choix du test qui figure à la racine de l'arbre et la création des branches correspondant au test et continue, d'une façon récursive, par la création des sous-arbres pour chacune des branches (figure 2-4). Ce type de stratégie de recherche, proposé initialement dans [Hun et al., 1996], est connu sous le nom "diviser pour régner" (*divide and conquer*).

```
C4.5(ensemble_apprentissage)
crée le noeud racine
construis_arbre(ensemble_apprentissage, racine)
erreur_prédite=élague_arbre(racine)

construis_arbre(T, noeud)
si tous les exemples de T ∈ à une même classe C
        étiquette noeud avec C
sinon si T est vide
        étiquette noeud avec la classe du père(noeud)
sinon
        choix du meilleur test X pour noeud
        pour chaque branche i de X
                soit T_i l'ensemble d'exemples correspondant à cette branche
                crée noeud_i
                construis_arbre(T_i, noeud_i)
```

**Figure 2-4** Principe de l'algorithme C4.5

Les arbres de décision construits en suivant cette procédure sont souvent sur-spécialisés et présentent, donc, un assez faible pouvoir prédictif. Pour cette raison, C4.5 comporte une deuxième étape, d'élagage, au cours de laquelle des sous-arbres sont remplacés par des feuilles ou par une de leurs branches. La recherche d'une représentation élaguée commence par les feuilles terminales de l'arbre de décision et remonte vers sa racine, en effectuant tous les remplacements qui améliorent l'estimation du taux d'erreur qui sera réalisé sur de nouveaux exemples (voir section suivante).

#### 2.3.1.3 Fonctions d'évaluation

Pour évaluer la qualité d'un test, C4.5 utilise le "rapport de gain" (*gain ratio*), qui représente le rapport entre le gain d'information utile pour la classification, apporté par l'ajout d'un test, et l'information potentielle générée par la partition des exemples.

Soient $C_j$, $j$=1..$k$, les classes d'un problème d'apprentissage et $T$ l'ensemble des exemples qui restent à classer dans une branche de l'arbre de décision. L'information nécessaire pour décider qu'un exemple appartient à la classe $C_j$ est :

$$-log_2\left(\frac{fréq\left(C_j,T\right)}{|T|}\right)\text{bits,}$$

où $fréq\left(C_j,T\right)$ et $|T|$ dénotent le nombre d'exemples de la classe $C_j$ et le nombre total d'exemples dans $T$.

L'information moyenne nécessaire pour connaître la classe d'un exemple arbitraire de cette branche est donc :

$$info(T)=-\sum_{j=1}^{k}\frac{fréq\left(C_j,T\right)}{|T|}\times log_2\left(\frac{fréq\left(C_j,T\right)}{|T|}\right)\text{bits.}$$

Soient $T_i$, $i=1..n$ les ensembles d'exemples correspondant à chaque branche $i$ obtenue par l'ajout d'un nouveau noeud sur la branche initiale. L'information nécessaire pour classer un exemple, après avoir rajouté ce noeud, correspond à la moyenne pondérée de l'information nécessaire pour chaque nouvelle branche $i$ :

$$info_X(T)=\sum_{i=1}^{n}\frac{|T_i|}{|T|}\times info(T_i)\text{ bits.}$$

Le gain d'information pertinente pour la classification obtenue par l'ajout du noeud $X$ est donc :

$$gain(X) = info(T) - info_X(T)$$

Cette mesure du gain informationnel a été utilisée pendant longtemps en tant que fonction d'évaluation par ID3 [Quinlan, 1986], l'ancêtre de C4.5. Ce dernier apporte un raffinement supplémentaire, fondé sur l'observation que le gain informationnel favorise d'une façon injustifiée les tests avec un grand nombre de branches. Pour éviter ce biais, C4.5 évalue l'information potentielle générée par la partition des exemples :

$$info\ partition(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times log_2\left(\frac{|T_i|}{|T|}\right) \text{bits.}$$

Le rapport entre le gain informationnel et l'information potentielle générée par la partition des exemples est utilisé pour départager les tests qui réalisent au moins la valeur moyenne du gain informationnel parmi l'ensemble des tests considérés.

Pour les attributs symboliques, le calcul du rapport de gain est trivial. Dans le cas des attributs numériques, C4.5 trie les exemples par rapport aux valeurs de l'attribut considéré et calcule le gain informationnel pour toutes les coupures entre deux exemples successifs, pour trouver la coupure optimale. Ce calcul peut s'effectuer en un seul passage mais demande l'examination de $v\text{-}1$ coupures, où $v$ est le nombre de valeurs distinctes de l'attribut considéré (dans le cas le plus défavorable, $v$ est égal au nombre total d'exemples). Une optimisation possible de cette procédure passe par l'observation que la meilleure coupure du point de vue du rapport de gain surviendra toujours à la frontière entre deux exemples de classes différentes [Fayyad & Irani, 1992]. Une discussion plus détaillée sur l'évaluation des tests sur les attributs numériques et des modifications récentes, proposées dans [Quinlan, 1996], sont présentées dans la section 3.2.3.

Une deuxième fonction d'évaluation, qui essaie de prédire le taux d'erreur qui sera réalisé par un sous-arbre sur des nouveaux exemples, est utilisée par C4.5 dans la recherche d'une représentation élaguée de l'arbre de décision. Le taux d'erreur est estimé à partir de la probabilité d'erreur *a posteriori* calculée pour les exemples d'apprentissage couverts par une feuille de l'arbre. Cette approximation génère une distribution binomiale, dont la limite supérieure de l'intervalle de confiance peut être évaluée pour un niveau de confiance donné.

En tenant compte du fait que l'utilisation des exemples d'apprentissage pour l'évaluation du taux d'erreur va sûrement générer des prédictions trop optimistes, c'est cette limite supérieure qui sert d'estimation du taux d'erreur pour l'élagage. Dans l'implémentation de C4.5 proposée par son auteur, le niveau de confiance constitue un paramètre optionnel d'exécution. La valeur de 25% est utilisée par défaut pour l'élagage et s'avère performante dans la plupart des applications pratiques.

### 2.3.2 Induction de règles

Tout arbre de décision peut être transformé dans une base de règles équivalente, simplement en créant une règle pour chaque feuille et en collectant dans le corps de la règle les conditions de tous les noeuds rencontrés sur le chemin qui relie la racine et la feuille considérée (des stratégies plus sophistiquées, qui réalisent en même temps une simplification des règles, sont proposées dans C4.5*rules* [Quinlan, 1993] ou dans SIPINA [Rabaséda & Zighed, 1996]).

La réciproque de cette affirmation n'est pas valable : les arbres de décision réalisent une partition complète et disjointe de l'espace des exemples tandis que les représentations à base de règles acceptent généralement des recouvrements qui seront tranchés au moment de la classification par des stratégies de résolution de conflits appropriées. Comme plusieurs règles peuvent contribuer conjointement à la classification d'un exemple, on peut espérer, en utilisant cette représentation, une atténuation des effets des règles aberrantes, produites à cause du manque ou de la corruption des données [Clark & Boswell, 1991].

Les représentations à base de règles sont aussi, potentiellement, plus facilement compréhensibles par un expert humain. Tandis que la compréhension d'un test figurant dans un noeud d'un arbre de décision nécessite la compréhension et le rappel de tous les autres tests qui l'ont précédé sur la même branche, les règles de classification sont indépendantes, les unes par rapport aux autres, et, donc, auto-explicatives.

Un des algorithmes très connus de cette catégorie est CN2 [Clark & Niblett, 1989]. Cet algorithme combine des idées de AQ15 (surtout sa stratégie de recherche fondée sur l'algorithme de l'étoile) et de ID3. Nous allons maintenant présenter cet algorithme pour donner un aperçu de l'induction de règles.

#### 2.3.2.1 Représentation des connaissances

Il existe deux options pour la représentation des résultats dans CN2. La version originale produit des règles ordonnées, appelées aussi listes de décision. Ensuite, pour classer un nouvel exemple, les conditions de chacune des règles sont testées, en suivant leur ordre, jusqu'à ce qu'une première règle puisse être tirée. A ce moment, l'algorithme de classification arrête la recherche et retourne la classe prédite par cette règle. Ce modèle de représentation de connaissances ne demande donc aucune stratégie de résolution des conflits mais présente encore la plupart des désavantages associés aux arbres de décision.

Une version plus récente de l'algorithme [Clark & Boswell, 1991] ajoute, en option, la possibilité de produire des règles indépendantes, sans aucun ordonnancement imposé. Pour résoudre les éventuels conflits au moment de la classification, CN2 réalise la sommation des distributions des classes parmi les exemples d'apprentissage couverts par les règles tirées et prédit la classe la plus représentée dans la distribution résultante.

Pour les deux variantes de représentation, le corps des règles est constitué par des conjonctions de tests qui relient un attribut à une valeur ou à une disjonction (ensemble) de valeurs.

#### 2.3.2.2 Stratégie de recherche

```
CN2règles_ordonnées(exemples)
règles := vide
répète
        conds := trouve_meilleures_conditions(exemples)
        si conds ≠ nulle
                classe := la classe majoritaire des exemples couverts par conds
                rajoute 'Si conds Alors classe' à règles
                enlève de exemples les exemples couverts par conds
jusqu'à ce que conds = nulle
rajoute à règles la règle par défaut qui prédit la classe majoritaire dans exemples
retourne règles
```

**Figure 2-5** Induction de règles ordonnées dans CN2

Pour apprendre des règles ordonnées, CN2 utilise une procédure itérative qui produit le corps d'une règle à chaque itération, lui associe en conclusion la classe la plus représentée parmi les exemples couverts, la rajoute à la liste de décision et élimine de l'ensemble d'apprentissage tous les exemples couverts par la règle respective (figure 2-5). Quand aucune règle n'est plus trouvée, CN2 termine la liste avec une règle par défaut, sans condition et qui prédit la classe la plus représentée dans l'ensemble d'apprentissage.

```
trouve_meilleures_conditions(exemples [,classe])
étoile := {'vrai'}, nouvelle_étoile := {}, meilleures_conds := nulle
tant que étoile ≠ {}
        pour chaque conds dans étoile
                pour chaque test possible
                        conds' := 'conds ET test'
```

si *conds'* meilleure que *meilleures_conds* et *conds'* statistiquement signifiantes

*meilleures_conds* := *conds'*

rajoute *conds'* à *nouvelle_étoile*

si taille de *nouvelle_étoile* > *MAXETOILE*

enlève la moins bonne condition de *nouvelle_étoile*

*étoile* := *nouvelle_étoile*

retourne *meilleures_conds*

**Figure 2-6** Recherche des meilleures conditions

**CN2règles_indépendantes**(*exemples*)

*règles* := vide

pour chaque *classe*

*règles_une_classe* := CN2pour_une_classe(*exemples*, *classe*)

rajoute *règles_une_classe* à *règles*

retourne *règles*

**CN2pour_une_classe**(*exemples*, *classe*)

*règles* := vide

répète

*conds* := trouve_meilleures_conditions(*exemples*, *classe*)

si *conds* ≠ nulle

rajoute 'Si *conds* Alors *classe*' à *règles*

enlève de *exemples* les exemples dans *classe* couverts par *conds*

jusqu'à ce que *conds* = nulle

retourne *règles*

**Figure 2-7** Induction de règles indépendantes dans CN2

A chaque itération, CN2 cherche la meilleure règle en utilisant une stratégie descendante (*top-down*) et en examinant plusieurs hypothèses en parallèle (*beam search*). La taille de l'ensemble d'hypothèses maintenues au cours de la recherche est limitée par un paramètre modifiable (*MAXETOILE*). A chaque étape de la recherche, cet ensemble est remplacé par l'ensemble des meilleures spécialisations obtenu par son intersection avec l'ensemble de toutes les conditions possibles (figure 2-6).

Pour apprendre des règles indépendantes, CN2 utilise une procédure similaire qui cherche les meilleures règles pour chacune des classes à son tour et enlève de l'ensemble d'apprentissage seulement les exemples couverts de la classe respective (figure 2-7).

#### 2.3.2.3 Fonction d'évaluation

CN2 utilise deux fonctions d'évaluation différentes pour apprécier la *qualité* de la distribution des exemples couverts et la *confiance* statistique de la différence entre cette distribution observée et la distribution attendue en cas de sélection aléatoire des exemples.

Pour estimer la qualité d'une hypothèse, CN2 propose deux options à l'utilisateur. Dans sa version originale, la fonction de qualité utilisait une mesure d'entropie, fondée sur la théorie de l'information, similaire à celle utilisée par ID3. L'autre option, plus récente [Clark & Boswell, 1991], s'appuie sur la loi de succession de Laplace (voir section 2.2.4 pour la présentation de cet estimateur) :

$\frac{n_c + 1}{n_{tot} + k}$, où $k$ est le nombre de classes du domaine, $n_c$ est le nombre d'exemples de la classe prédite couverts par la règle et $n_{tot}$ est le nombre total d'exemples couverts par la règle.

L'autre fonction d'évaluation sert à élaguer l'espace de recherche. La formule utilisée est fondée sur le taux de vraisemblance statistique :

$2\sum_i f_i \, log\left(\frac{f_i}{e_i}\right)$, où $f_i$ est le nombre d'exemples de la classe $i$ actuellement couverts par la règle et $e_i$ est le nombre attendu d'exemples de la même classe en supposant que les exemples sont choisis d'une façon aléatoire (cette valeur est estimée par la fréquence de la classe dans l'ensemble global d'apprentissage, pondérée par le nombre total d'exemples couverts par la règle).

Cette statistique correspond à une sorte de distance entre les distributions. Plus cette valeur est basse, plus il est probable que la distribution observée soit due à la chance. En choisissant un seuil inférieur élevé, on peut obliger l'algorithme à prendre en compte au cours de la recherche seulement les hypothèses statistiquement significatives.

### 2.3.3 Induction d'hyperrectangles

L'induction d'hyperrectangles constitue un point de rencontre, relativement récent, entre plusieurs directions de recherche différentes en apprentissage automatique. Ce paradigme regroupe actuellement des approches issues de l'apprentissage symbolique [Salzberg, 1991], [Christiensen, 1992], [Wettschereck & Dietterich, 1995], de la logique floue et des réseaux de neurones [Carpenter, 1992], [Simpson, 1992] et de la théorie de l'apprentissage calculatoire [Helmbold et al., 1989].

Ces algorithmes peuvent être rangés dans une catégorie de méthodes d'apprentissage automatique qui retardent la généralisation complète des exemples jusqu'au moment de la classification - l'apprentissage paresseux (*lazzy learning*), dont les représentants les plus connus sont les algorithmes fondés sur les plus proches voisins (IBn [Aha et al., 1991]). Ils apportent à ces méthodes de nouvelles idées issues de l'induction de règles et de l'apprentissage non-supervisé.

Les fondements des algorithmes d'induction d'hyperrectangles ont été posés dans la communauté d'apprentissage symbolique par la "théorie des exemples généralisés imbriqués" (*Nested Generalized Exemplars Theory*-NGE) introduite dans [Salzberg, 1991]. Ces travaux ont été directement continués par [Wettschereck & Dietterich, 1995], qui propose un nouvel algorithme, plus performant, BNGE. Nous allons utiliser ces deux algorithmes pour exemplifier l'induction d'hyperrectangles. Dans la section 3.2, nous allons également présenter un algorithme original d'induction d'hyperrectangles (PmBc) et quelques considérations personnelles sur les particularités de ces approches.

#### 2.3.3.1 Représentation des connaissances

Dans la phase d'apprentissage, les algorithmes d'induction d'hyperrectangles construisent des généralisations abstraites des exemples présentés. Leur nom provient du fait que ces représentations peuvent être figurées, dans les espaces numériques, sous forme d'hyperrectangles alignés par rapport aux axes.

La partition de l'espace des exemples réalisée par les hyperrectangles construits n'est généralement pas complète (c'est le cas de NGE et BNGE) et souvent non-disjointe (comme dans NGE). Des stratégies de classification appropriées doivent être envisagées pour résoudre les situations de conflit ou de manque de décision. Les deux algorithmes que nous avons choisis comme exemples déterminent la classe des exemples non-couverts à partir de l'hyperrectangle le plus proche d'eux (la distance entre un exemple et un hyperrectangle qui le recouvre est 0 par définition). De ce point de vue, ces représentations peuvent être assimilées à des règles de classification floues dont la fonction d'appartenance décroît avec la distance.

Dans NGE, les situations de conflit sont tranchées en faveur de l'hyperrectangle d'aire minimale.

#### 2.3.3.2 Stratégie de recherche

Le principe de fonctionnement de ces algorithmes, tel qu'il est défini dans [Wettschereck & Dietterich, 1995], vise à trouver le nombre minimal d'hyperrectangles qui classent correctement les exemples d'apprentissage. Plusieurs stratégies de recherche, de nature heuristique, ont été envisagées pour implémenter cet objectif. Elles sont fondées d'habitude sur des critères de similarité et se rapprochent souvent des techniques d'apprentissage non-supervisé. Des différences importantes subsistent quant à l'acceptation des hyperrectangles imbriqués et /ou se recouvrant.

```
pour les s (les noyaux) premiers exemples e
        crée un hyperrectangle ponctuel pour e
pour chaque nouvel exemple e
        cherche le plus proche hyperrectangle h1
        si classe(e)=classe(h1)
                généralise(h1,e)
        sinon cherche le deuxième plus proche hyperrectangle h2
                si classe(e)=classe(h2)
                        généralise(h2,e)
                sinon
                        crée un hyperrectangle ponctuel pour e
```

**Figure 2-8** Principe de l'algorithme NGE

L'algorithme NGE original (dont le pseudo-code simplifié est présenté dans la figure 2-8) réalise un apprentissage incrémental dans lequel chaque nouvel exemple contribue à la généralisation d'un des deux hyperrectangles les plus proches déjà existants ou produit la création d'un nouvel hyperrectangle. Cette stratégie permet la construction d'hyperrectangles imbriqués ou se recouvrant partiellement.

*H*=l'ensemble d'hyperrectangles ponctuels correspondants aux exemples
exécute la boucle :
  cherche les deux hyperrectangles *h1*, *h2* les plus proches l'un de l'autre qui satisfont les
    conditions suivantes:
    *classe*(*h1*)=*classe*(*h2*)
    *h=généralise*(*h1*,*h2*) ne chevauche aucun hyperrectangle d'une autre classe
  si *h1*, *h2* trouvés
    remplace *h1* et *h2* par h dans *H*
  sinon
    finis la boucle

**Figure 2-9** Principe de l'algorithme BNGE

BNGE abandonne le caractère incrémental de NGE pour un traitement par lots plus efficace et sans paramètres à régler. La supériorité de BNGE par rapport à NGE est validée d'une façon expérimentale sur huit bases de données de l'UCI [Murphy & Aha, 1993] plus trois tâches de classification artificielles.

BNGE démarre avec un ensemble d'hyperrectangles ponctuels correspondants aux exemples et, à chaque pas, regroupe les deux hyperrectangles les plus proches appartenant à la même classe. Des tests sont effectués pour éviter que les hyperrectangles ainsi produits se recouvrent. La figure 2-9 présente les lignes générales de BNGE.

#### 2.3.3.3 Fonction d'évaluation

A la différence des autres algorithmes d'apprentissage présentés, les fonctions d'évaluation de NGE et BNGE sont intimement imbriquées dans leurs stratégies de recherche. Pour couvrir un nouvel exemple, NGE généralise toujours un des deux hyperrectangles les plus proches de l'exemple à couvrir. BNGE utilise aussi un critère de distance minimale (mais vérifie en plus le manque de recouvrement entre les représentations des différentes classes) pour choisir les deux hyperrectangles qui seront fusionnés à chaque itération.

### 2.3.4 Programmation logique inductive

Les algorithmes d'induction d'arbres de décision ou de règles présentés dans les sections précédentes utilisent pour l'apprentissage uniquement une table d'exemples décrits sous la forme d'un vecteur d'attributs-valeurs. Les représentations obtenues sont des fonctions, exprimées dans un formalisme logique, entre les valeurs des attributs et la classe d'un exemple.

Dans certains problèmes d'apprentissage, les exemples sont mieux décrits sous la forme d'objets complexes, difficilement convertibles dans des enregistrements plats et de longueur fixe. Les recherches en apprentissage automatique ont été récemment orientées vers des formalismes de représentation de connaissances plus puissants comme les modèles centrés objets [Manago & Kodratoff, 1991] et les logiques d'ordre supérieur.

La programmation logique inductive (*inductive logic programming* ou *ILP*) est un domaine de recherche situé à l'intersection de l'apprentissage symbolique et de la programmation logique. Tous les objets manipulés par ces travaux (exemples, représentations construites à partir des exemples, connaissances du domaine) sont décrits sous forme de programmes logiques de premier ordre. Ce cadre fédérateur regroupe actuellement des approches issues de domaines variés comme l'apprentissage symbolique traditionnel (FOIL [Quinlan, 1990]), la synthèse de programmes logiques et la théorie de l'apprentissage calculatoire, (MIS [Shapiro, 1983]) ou l'acquisition de connaissances, (MOBAL [Kietz & Wrobel, 1992]). La maturité de cette discipline est confirmée par le développement de concepts et méthodes spécifiques, comme la moins générale généralisation (*least general generalisation*) utilisée par GOLEM [Muggleton & Feng, 1992], qui semblent converger vers une théorie unificatrice de la programmation logique inductive [Muggleton & De Raedt, 1994].

Pour donner un aperçu de ce domaine, nous avons choisi FOIL [Quinlan, 1990], parce qu'il est directement issu de l'apprentissage symbolique classique (il reprend un grand nombre d'idées de ID3 et C4.5) et donc, il est plus simple à présenter dans ce contexte, et parce que sa stratégie de recherche est à la base de notre système d'induction interactive de règles, COPAC (présenté dans la section 3.4).

#### 2.3.4.1 Représentation de connaissances

Le but de la programmation logique inductive est de produire une définition intentionnelle pour une relation cible décrite par des exemples, en s'appuyant éventuellement sur d'autres relations (*background knowldege*) fournies sous une forme intentionnelle ou extensionnelle. La plupart de ces algorithmes permettent certaines formes de récursivité et de quantification et utilisent le langage PROLOG pour la présentation de leurs résultats.

FOIL produit des définitions dans un dialecte PROLOG sans *cut* (!), *fail*, buts disjonctifs et fonctions autres que les constantes. Plus précisément, les littéraux qui peuvent apparaître dans le corps d'une clause produite par FOIL sont :

- $Q(V_1, V_2, ..., V_k)$ et $not(Q(V_1, V_2, ..., V_k))$ où $Q$ est une relation.
- $V_i = V_j$ et $V_i \neq V_j$, où $V_i$ et $V_j$ sont des variables du même type.
- $V_i = c$ et $V_i \neq c$ où $V_i$ est une variable et $c$ est une constante, indiquée par l'utilisateur comme étant acceptable dans une définition, du même type que la variable $V_i$.
- $V_i \leq V_j$, $V_i > V_j$, $V_i \leq t$ et $V_i > t$ où $V_i$ et $V_j$ sont des variables numériques et $t$ est un seuil calculé par FOIL.

Par exemple, une représentation produite par FOIL pour définir l'addition arithmétique (*plus*), d'une façon récursive, à partir de la relation *dec* (diminution d'une unité), est :

*plus*(*A*,*B*,*C*) :- *A*=0, *B*=*C*.

*plus*(*A*,*B*,*C*) :- *dec*(*A*,*D*), *dec*(*C*,*E*), *plus*(*B*,*D*,*E*).

#### 2.3.4.2 Stratégie de recherche

FOIL est fondé sur une stratégie de recherche appelée "séparer pour régner" (*separate and conquer*), similaire à celle utilisée par CN2 pour la création des règles indépendantes. Cette stratégie consiste dans la création itérative de clauses qui couvrent une partie des exemples positifs et aucun exemple négatif. Après chaque itération, les exemples couverts sont enlevés de l'ensemble d'apprentissage et le processus continue jusqu'à ce que tous les exemples positifs soient couverts par au moins une clause. En fait, pour éliminer les effets du bruit, FOIL utilise un critère d'arrêt qui empêche la spécialisation ou la création de règles quand le nombre d'exemples qui restent à traiter est apprécié comme insignifiant.

Pour la construction des clauses, à chaque itération, FOIL adopte une approche descendante qui commence par la clause la plus générale (*relation_cible :-*) et rajoute successivement des littéraux au corps de la clause, jusqu'à ce qu'elle ne couvre plus d'exemples négatifs (figure 2-10). Il faut signaler ici l'existence des approches alternatives, de type ascendant, fondées sur des généralisations successives, dont le représentant le plus connu est GOLEM [Muggleton & Feng, 1992].

```
théorie := programme vide
restant := tous les exemples ∈ relation_cible
tant que restant n'est pas vide et le principe MDL est respecté
        clause := 'relation_cible:-'
        tant que clause couvre des exemples ∉ relation_cible et le principe MDL est respecté
                trouve le meilleur littéral L
                rajoute L au corps de clause
        enlève de restant les exemples ∈ relation_cible couverts par clause
        rajoute clause à théorie
```

**Figure 2-10** Principe de l'algorithme FOIL

Soit une clause partielle qui contient *v* variables. On appelle les *substitutions* de cette clause toutes les listes de *v* constantes qui satisfont le corps de la clause. Ces substitutions seront étiquetées comme positives ou négatives selon l'appartenance à la relation cible des sous-listes de constantes qui correspondent aux variables de la tête de la clause (les exemples d'apprentissage). Il faut noter donc qu'un même exemple d'apprentissage peut donner naissance à plusieurs substitutions différentes d'une clause partielle.

Les littéraux rajoutés par FOIL à la clause courante servent à enlever des exemples négatifs couverts ou à introduire des nouvelles variables qui seront utilisées dans les littéraux suivants. Parmi les littéraux qui introduisent des nouvelles variables, FOIL préfère ceux pour lesquels la nouvelle clause partielle a exactement une substitution pour chaque substitution positive de la clause initiale et au plus une substitution pour chaque substitution négative de celle-ci (les littéraux *déterminés*).

La stratégie adoptée par FOIL choisit, à chaque itération :

- le littéral qui maximise le gain informationnel si ce gain informationnel est proche du maximum possible; <u>sinon</u>
  - tous les littéraux déterminés trouvés; <u>sinon</u>
    - un littéral avec un gain informationnel positif; <u>sinon</u>

– un littéral qui introduit une nouvelle variable.

D'autres aspects de la stratégie de recherche utilisée par FOIL concernent la récursivité, l'élagage de l'espace de recherche, le retour arrière, la simplification des définitions et les effets de frontière causés par un vocabulaire limité.

#### 2.3.4.3 Fonction d'évaluation

La qualité d'un littéral qui pourrait être rajouté à une clause dans le but d'enlever des exemples négatifs encore couverts est évaluée par une mesure du gain informationnel apporté par ce littéral, similaire à celle utilisée par ID3 (section 2.3.1).

Soit $p$ et $n$ les nombres de substitutions positives et négatives d'une clause partielle. L'information moyenne apportée par la découverte qu'une des substitutions est étiquetée comme positive est de :

$$I(p,n) = -log_2\left(\frac{p}{p+n}\right)\text{bits}$$

Soit p' et n' le nombre de substitutions positives et négatives de la clause complétée par le littéral examiné. Par l'ajout de ce littéral, certaines des substitutions initiales seront enlevées et chacune de celles qui restent produiront une ou plusieurs substitutions pour la nouvelle clause. Soit $k$ le nombre de substitutions positives qui ne sont pas enlevées par l'ajout du littéral considéré. Le gain informationnel apporté par ce nouveau littéral est donc de :

$$k \times \left(I(p,n) - I(p',n')\right)\text{bits.}$$

FOIL utilise aussi, d'une manière similaire à CN2, un critère d'arrêt fondé sur le principe MDL (*Minimum Description Length*) [Rissanen, 1978], qui arrête l'apprentissage d'une clause au moment où le coût de l'encodage de la clause (en quantité d'information) dépasse le coût total de l'encodage des exemples couverts par la clause.

## 2.4 Révision automatique de connaissances

Dans les algorithmes de révision de connaissances, l'apprentissage est initialisé par une théorie fournie par un expert. Pour exemplifier ce domaine nous allons présenter très succinctement l'algorithme de raffinement de règles EITHER [Ourston & Mooney, 1990].

### 2.4.1 Représentation de connaissances

EITHER révise des théories représentées sous forme de règles constituées par des conjonctions de conditions qui imposent des restrictions sur les valeurs des attributs (représentation de type attribut-valeur). Les attributs peuvent être symboliques (binaires ou multivalués) ou numériques.

A la différence de la plupart des algorithmes d'induction pure, EITHER peut travailler aussi sur des règles qui contiennent des concepts intermédiaires en plus des attributs des exemples (observables) et des concepts à apprendre (classes). Par exemple, dans la "théorie des tasses" utilisée par ses auteurs pour expliquer le fonctionnement de EITHER, un tel concept intermédiaire est constitué par la propriété "peut être saisi" (*graspable*), appréciée à partir des propriétés observables (dimensions, présence d'une poignée) et qui contribue à la définition des tasses (la classe à apprendre). Cette facilité peut augmenter l'intelligibilité des règles et leur capacité à fournir des explications.

EITHER peut réviser des théories sur-spécialisées, sur-généralisées ou affectées par les deux catégories d'erreurs. Il garantit l'obtention d'une théorie finale en accord avec l'ensemble d'exemples d'apprentissage.

### 2.4.2 Fonction d'évaluation

EITHER essaie de modifier d'une façon "minimale" la théorie initiale fournie par l'expert dans le but de classer correctement tous les exemples de l'ensemble d'apprentissage. La qualité d'une révision est donc appréciée par un rapport bénéfices-coût (*benefit-to-cost ratio*) qui réalise un compromis entre le nombre d'exemples dont la classification serait corrigée par cette révision, la complexité syntactique de la révision et le nombre de nouveaux exemples incorrects qu'elle produit.

### 2.4.3 Stratégie de recherche

EITHER utilise une combinaison de plusieurs méthodes d'inférence pour la révision de théorie : la déduction, l'abduction et l'induction (figure 2-11).

La déduction (par chaînage arrière) est utilisée pour isoler les exemples mal classés par la théorie initiale. On appelle positif erroné tout exemple pour lequel on ne peut pas déduire l'appartenance à sa classe. Un exemple pour lequel on peut déduire l'appartenance à une autre classe que la sienne est appelé négatif erroné. EITHER essaie de supprimer des conditions pour couvrir les positifs erronés et de supprimer des règles pour éliminer les négatifs erronés.

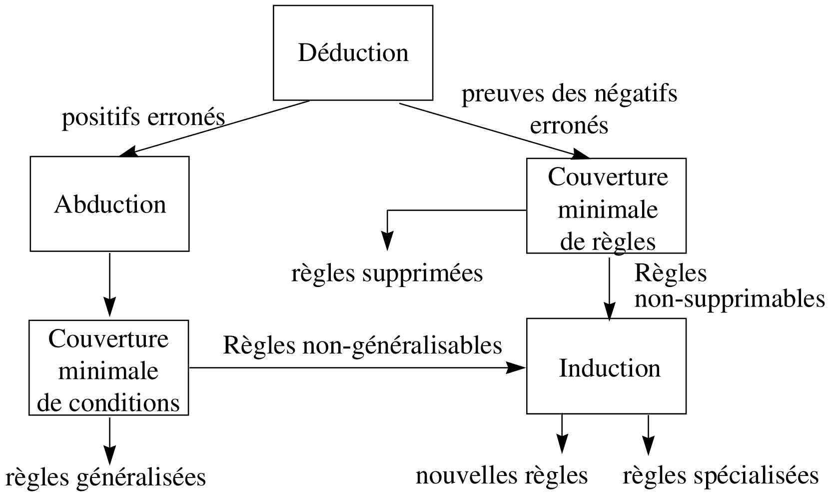



**Figure 2-11** Principe de l'algorithme EITHER

Pour chaque positif erroné, EITHER utilise l'abduction pour générer tous les ensembles minimaux des conditions les plus spécifiques qui, enlevées de certaines règles, corrigeraient l'erreur. Ensuite, il utilise une stratégie heuristique de recherche, fondée sur le rapport bénéfices-coût, pour choisir parmi ces révisions un ensemble réduit qui garantit la correction de tous les positifs erronés.

Pour chaque négatif erroné, EITHER analyse les preuves construites pendant la déduction pour déterminer toutes les suppressions de règles qui pourraient corriger l'erreur. Pour construire un ensemble réduit de révisions, il utilise ensuite une procédure de recherche itérative qui choisit, à chaque pas, la règle qui contribue au plus grand nombre de preuves erronées, jusqu'à ce que tous les négatifs erronés soient corrigés.

Si certaines suppressions de conditions ou de règles conduisent à l'apparition de nouvelles erreurs, EITHER fait appel à des techniques d'induction (ID3) pour apprendre de nouvelles règles ou de nouvelles conditions qui corrigeront complètement la théorie par rapport aux exemples.

## 2.5 Extraction automatique de connaissances dans les bases de données

L'extraction automatique de connaissances dans les bases de données (*Knowledge Discovery in Databases* [Piatesky-Shapiro & Frawley ,1991]ou *Data Mining* [Holsheimer & Siebes, 1994]) est un domaine de recherche relativement récent qui a comme but la découverte des régularités non-triviales et compréhensibles pour les humains dans les bases de données. Par rapport à l'apprentissage automatique classique, souvent expérimenté uniquement sur des petits ensembles d'exemples artificiels ou soigneusement préparés en laboratoire, ces recherches mettent l'accent sur la résolution des problèmes rencontrés dans l'application de ces techniques aux larges bases de données accumulées dans les activités pratiques d'ordre industriel, financier ou de recherche.

Pour réaliser ces objectifs, l'extraction automatique de connaissances dans les bases de données s'appuie sur les techniques issues de l'apprentissage automatique mais fait aussi appel aux méthodes statistiques et de gestion de bases de données. Les sections suivantes donnent un bref aperçu, loin d'être exhaustif, des problèmes spécifiques rencontrés dans ce domaine de recherche.

### 2.5.1 Corruption des données

L'élimination des effets négatifs des exemples corrompus sur la qualité de l'apprentissage constitue depuis longtemps un sujet privilégié de recherche dans le domaine de l'apprentissage automatique. Ce problème se pose même avec plus d'acuité pour l'extraction de connaissances dans les bases de données accumulées dans la pratique.

#### 2.5.1.1 Attributs manquants

Une des formes de corruption, souvent rencontrées dans les bases de données, concerne le manque de disponibilité des valeurs pour certains attributs. Ceci pose deux types de problèmes aux algorithmes d'extraction de connaissances : comment construire des représentations correctes en s'appuyant sur des exemples incomplets et comment utiliser les connaissances produites pour traiter des exemples avec des attributs manquants?

A part la solution triviale mais parfois inapplicable, qui consiste à simplement ignorer les exemples incomplets, deux types d'approches sont souvent utilisées dans cette situation. L'algorithme d'apprentissage peut estimer des valeurs pour ces attributs manquants, en les remplaçant par les valeurs les plus probables sur l'ensemble d'apprentissage [Clark & Niblett, 1989] ou en essayant de les prédire à partir des valeurs des autres attributs et de la classe des exemples en question [Quinlan, 1986].

Dans d'autres cas, il peut s'avérer plus intéressant de traiter les attributs manquants comme des valeurs particulières (*INCONNU*) et d'essayer d'induire des connaissances à partir de cette absence de valeurs pour certains exemples. L'absence systématique de valeurs pour l'attribut *enceinte* ou *nom_de_jeune_fille* peut, par exemple, être utilisée au cours de la mise au point de stratégies commerciales à partir des bases de données, pour mettre en évidence des différences entre les préférences des hommes et des femmes, même si l'attribut *sexe* n'a pas été enregistré.

#### 2.5.1.2 Bruit

Si les exemples affectés par le manque d'attributs peuvent facilement être repérés dans une base de données et, donc, peuvent être traités d'une façon particulière, il est beaucoup plus difficile de détecter et d'isoler les exemples qui contiennent des informations incorrectes. Malheureusement, cette forme de corruption est souvent rencontrée dans la pratique et l'extraction des connaissances dans les bases de données nécessite, donc, des moyens de traitement du bruit efficaces pour s'avérer utile dans ce type d'environnement.

Le bruit peut affecter les valeurs des attributs ou la classe des exemples, aussi bien dans la phase d'apprentissage que dans l'utilisation effective des connaissances produites sur des nouveaux exemples. Pendant l'apprentissage, le bruit a tendance à forcer les algorithmes d'apprentissage à construire des représentations très spécialisées, qui essaient de modéliser les exemples aberrants au même titre que les informations utiles. Si aucune mesure n'est prise pour gérer le bruit, ces représentations seront extrêmement dépendantes de l'ensemble d'apprentissage et auront, donc, un faible pouvoir de prédiction.

Les techniques utilisées par les algorithmes d'apprentissage pour éviter la sur-spécialisation des représentations construites sont connues sous le nom d'élagage (pour les arbres de décision elles consistent dans la suppression des branches très spécialisées). On distingue deux approches différentes, selon la place qu'occupe ce traitement au cours du processus d'apprentissage : le pré-élagage et le post-élagage.

Le pré-élagage consiste dans l'utilisation d'un critère d'arrêt qui empêche l'algorithme d'apprentissage de continuer la spécialisation de la théorie produite quand le nombre d'exemples à modéliser devient statistiquement insignifiant. Dans la section 2.3 nous avons mentionné deux des tests les plus utilisés, fondés sur :

- le test de confiance statistique, dans CN2.

– le principe MDL (*Minimum Description Length*), dans FOIL.

Le post-élagage constitue une famille de techniques alternatives d'élimination des effets du bruit, qui consistent dans des traitements effectués sur les résultats finaux d'un apprentissage. Dans une première variante, l'ensemble d'apprentissage est divisé en deux sous-ensembles : un ensemble pour l'apprentissage (*Grow*) et un autre pour l'élagage (*Prune*). L'apprentissage est réalisé sur le premier sous-ensemble, sans tenir compte du bruit, et la théorie obtenue est simplifiée (en enlevant des règles et des conditions, par exemple) tant que la qualité de la classification estimée sur le deuxième sous-ensemble ne décroît pas.

On peut citer, comme exemples d'algorithmes de cette catégorie, REP [Brunk & Pazzani, 1991] et GROW [Cohen,1993], fondés sur des heuristiques différentes de simplification de la théorie initiale. Des approches mixtes, qui combinent ces techniques avec le pré-élagage, comme I-REP [Fürnkranz & Widmer, 1994], ont été développées pour obtenir une meilleure efficacité de calcul.

Pour éviter la diminution du nombre d'exemples disponibles pour l'apprentissage, le post-élagage peut-être réalisé en se basant sur une prédiction probabiliste du taux d'erreur de classification. Ce type de post-élagage est utilisé dans C4.5 pour l'élagage des arbres de décision (section 2.3.1). Dans l'induction de règles, plusieurs méthodes d'élagage peuvent être rangées dans cette catégorie, comme :

- l'utilisation d'une *j*-mesure [Goodman & Smith, 1988], [Quinlan, 1988], qui réalise une estimation de la diminution moyenne de l'information nécessaire pour spécifier la classe de la règle après l'application de ses conditions, par rapport à la situation initiale;

- l'utilisation d'un seuil de spécialisation, qui impose une probabilité *a posteriori* minimale pour la classe proposée par la règle, et d'un nombre minimal d'exemples couverts de cette classe [Zighed, 1992];

- l'utilisation d'une mesure de l'intensité de l'implication, fondée sur un test statistique qui indique que le nombre de contre-exemples couverts est "improbablement petit" [Rakotomalala *et al.*, 1996].

### 2.5.2 Particularités liées aux systèmes de gestion de bases de données

Les larges quantités de données explorées pour l'extraction de connaissances sont d'habitude prises en charge par des systèmes de gestion appropriés (SGBD). Une des directions de recherche dans ce domaine vise à réaliser une intégration étroite entre les techniques d'apprentissage automatique et ces systèmes de gestion de bases de données.

#### 2.5.2.1 Larges quantités d'information

Les bases de données contiennent souvent beaucoup d'objets décrits par de nombreux attributs. En même temps que l'espace des représentations possibles exploré par les algorithmes d'apprentissage croît rapidement avec le nombre d'attributs, les requêtes nécessaires pour estimer la qualité des hypothèses deviennent extrêmement coûteuses, en temps de réponse, pour un nombre important d'exemples.

Un certain nombre de recherches [Kaufman et al., 1991], [Holsheimer & Kersten, 1994], [Freitas & Lavington, 1996], [Han *et al.*, 1996] proposent l'exploitation directe ou l'adaptation des techniques disponibles dans les SGBD pour rendre l'accès aux données plus efficace. Ces optimisations passent par l'utilisation de structures de stockage appropriées, par l'encapsulation des fonctionnalités spécifiques à l'apprentissage comme des opérations standards du SGBD, par l'utilisation des traitements parallèles, etc.

#### 2.5.2.2 Evolutivité de l'information

L'information gérée par les SGBD est continuellement actualisée par des ajouts, des suppressions ou des modifications d'objets. Dans la mesure où ces modifications correspondent à des changements dans les caractéristiques des phénomènes sous-jacents, il est important qu'elles soient répercutées sur les connaissances extraites de ces données.

La fréquence de l'actualisation des connaissances, la pondération de l'importance des informations récentes par rapport aux anciennes données et le type d'apprentissage à utiliser au cours des actualisations sont autant de problèmes auxquels l'extraction automatique de connaissances dans les bases de données doit apporter des réponses.

# Chapitre 3 : CONTRIBUTIONS EN APPRENTISSAGE AUTOMATIQUE

Ce chapitre présente, dans une première partie, un aperçu des recherches en parole (sans se vouloir un panorama exhaustif) et les problèmes engendrés pour l'apprentissage symbolique. Dans les sections suivantes, nous exposons quelques contributions en apprentissage, développées spécialement en tenant compte de cette spécificité.

## 3.1 Spécificité des problèmes d'apprentissage symbolique pour la recherche en Parole

Les travaux qui seront présentés dans les sections suivantes de ce chapitre ont été motivés par un certain nombre d'observations sur les spécificités de certaines problèmes d'apprentissage en parole. Face à l'impressionnante complexité du domaine il faut, avant tout, préciser le champ d'intérêt de ces recherches.

Pour résumer, la parole est étudiée par les scientifiques à différents niveaux complémentaires ou superposés :

- le niveau **acoustique** : l'analyse du signal acoustique de parole;
- le niveau **articulatoire** : l'analyse de la production de la parole par les articulateurs du conduit vocal et des signaux visuels associés à la dynamique de ces articulateurs;
- le niveau **phonétique** : l'analyse des caractéristiques acoustiques et articulatoires des phonèmes, de leurs enchaînements et de leurs déformations en fonction du contexte;
- le niveau **phonologique** : l'analyse des différentes façons possibles de prononcer un mot en contexte, des élisions, des insertions ou des substitutions de mots et des liaisons entre les mots;
- le niveau **prosodique** : l'analyse des variations des paramètres acoustiques tels que la fréquence fondamentale, la durée, l'énergie et les pauses;

- les niveaux **lexical** et **syntaxique** : l'analyse des mots et des enchaînements possibles entre eux en fonction d'une grammaire de la langue;

- le niveau **sémantique** : l'analyse du sens des mots, de leurs relations et de la cohérence du sens dans les énoncés;

- le niveau **pragmatique** : l'analyse de la communication parlée en fonction de la situation et des objectifs.

Les recherches effectuées à l'Institut de la Communication Parlée se concentrent surtout sur les aspects proches des signaux physiques (acoustique, articulatoire, phonétique, prosodie) de la parole. Ces recherches ont des objectifs distincts mais souvent complémentaires :

- création des artefacts pour la **reconnaissance** et la **synthèse** de la parole.

- compréhension de la **production** et de la **perception** de la parole naturelle.

L'apprentissage automatique constitue depuis longtemps un outil privilégié de travail pour les chercheurs en communication parlée. A cause de la grande variabilité du signal de parole, une grande partie des recherches sur la parole s'appuie sur des analyses statistiques de ses différents aspects. Parmi ces méthodes, les chaînes de Markov cachées [De Mori & Brugnara, 1995] se sont avérées les plus performantes pour rendre compte de la variabilité temporelle du signal de parole. Les approches connexionistes sont également utilisées à grande échelle dans ces recherches, souvent en association avec les chaînes de Markov [Bourlard & Morgan, 1995].

Ces approches ont démontré indubitablement leurs performances dans des problèmes de recherche difficiles et variés. En proposant l'utilisation des techniques d'apprentissage symbolique pour la recherche en parole, notre objectif n'est pas de remettre en question l'intérêt des approches classiques, mais plutôt d'élargir la palette des méthodes d'apprentissage disponibles dans ce domaine.

Cette tâche n'est pas facile. Les résultats relativement décevants des projets proposant des architectures de reconnaissance issues de l'Intelligence Artificielle ont éloigné la communauté de recherche en parole des approches dites "symboliques" ou fondées sur les connaissances, au moins dans les études proches des signaux physiques (ce genre d'approche reste toujours majoritaire pour les analyses de la parole aux niveaux syntaxique, sémantique ou pragmatique). Nous pensons, cependant, que les avancées récentes en apprentissage symbolique supervisé permettent aujourd'hui à ces méthodes de rivaliser du point de vue des performances de classification avec les techniques statistiques ou connexionistes, tout en gardant une compréhensibilité accrue de leurs résultats qui pourrait faciliter les échanges avec les connaissances scientifiques ou de nature experte des chercheurs en parole.

Les paragraphes suivants de cette section se proposent d'analyser quelques particularités des problèmes d'apprentissage en parole auxquels les techniques d'apprentissage symbolique devraient apporter des réponses.

### 3.1.1 Attributs numériques

Les signaux physiques (acoustique, visuel, laryngographique, etc.) de parole, de nature intrinsèquement analogique, font l'objet de pré-traitements variés (amplification, filtrage et échantillonnage) en vue de leur analyse par des méthodes numériques sur ordinateur. Le but de ces méthodes est de compresser l'information transportée par les signaux pour faciliter la découverte des régularités correspondant aux messages codés dans la parole.

La mise en correspondance des mesures numériques qui caractérisent les signaux de parole avec des symboles associés aux événements porteurs d'un certain type d'information (les *étiquettes*) constitue une des tâches les plus rencontrées dans ces recherches. Nous allons appeler *étiquetage du signal* ce processus de translation entre les représentations numériques et symboliques des signaux de parole. Cette définition recouvre des activités variées en décodage acoustico-phonétique, prosodique, articulatoire, etc. (Le concept d'étiquetage peut également être généralisé pour les translations entre des représentations symboliques ou mixtes de la parole, à des niveaux d'abstraction différents : l'étiquetage des actes du dialogue, par exemple).

Pour réaliser l'étiquetage automatique du signal, deux classes de méthodes sont généralement envisagées : la réalisation des systèmes experts fondés sur les connaissances déjà acquises par les chercheurs en parole ou l'apprentissage automatique des étiquettes, à partir de corpus de signaux préalablement étiquetés.

Pour s'avérer performantes dans ce contexte, les méthodes d'apprentissage symbolique doivent accorder une attention particulière au traitement des tâches de classification caractérisées par un grand nombre d'attributs numériques, qui ont été souvent ignorées ou négligées dans ce domaine. Des propositions qui vont dans ce sens seront présentées dans les sections 3.2 et 3.3.

### 3.1.2 Bruit et représentations incomplètes

Nous avons vu précédemment qu'il est nécessaire de soumettre la parole à un grand nombre de transformations successives pour obtenir des représentations compactes qui seront utilisées comme exemples dans les tâches de classification. Au long de ces chaînes de traitements, différents facteurs (bruit acoustique imprévu présent lors de l'enregistrement, effets de filtrage du microphone, imprécisions des programmes de traitement du signal ou erreurs d'étiquetage) contribuent à éloigner la représentation finale de l'idéal théorique. Dans la mesure où ces facteurs sont difficilement modélisables pour une prise en compte explicite, il est donc indispensable de faire appel à des techniques appropriées pour éliminer ou pour diminuer l'influence de ce **bruit** (présent aussi bien dans les attributs que dans les classes des exemples) sur les résultats de l'apprentissage (section 2.5.1).

Il existe aussi une autre raison (apparentée selon nous à la précédente) pour laquelle la modélisation des exemples ne peut pas être réalisée d'une façon parfaite. Sans nous lancer dans le débat philosophique sur l'existence des phénomènes intrinsèquement aléatoires, on peut facilement observer qu'il est difficile de prendre en compte toutes les caractéristiques pouvant participer à la discrimination des exemples dans une application concrète. S'ils ne disposent pas de la totalité des informations nécessaires pour une discrimination parfaite entre les classes, les algorithmes d'apprentissage ne pourront pas produire une représentation en accord avec tous les exemples disponibles et qui garde en même temps un pouvoir de prédiction élevé. C'est le cas, par exemple, d'un problème de classification en trois dimensions, où on ne présenterait à l'algorithme d'apprentissage qu'une description bidimensionnelle, en utilisant les deux axes les plus informatifs. Les éventuelles informations discriminantes qui manquent, d'une manière systématique, dans la description des exemples, jouent un rôle similaire au bruit pour l'apprentissage et rendent nécessaire l'utilisation des techniques d'élagage pour augmenter le pouvoir prédictif de la représentation obtenue.

Les recherches présentées dans la section 3.2 ont été en partie motivées par ces observations.

### 3.1.3 Connaissances des experts et connaissances scientifiques

Les avancées les plus importantes dans la recherche en communication parlée passent, comme pour les autres disciplines expérimentales, par l'établissement de certains résultats, validés à grande échelle, et dont l'accumulation constitue les fondements d'un raisonnement déductif, le seul capable de préserver la vérité des assertions obtenues par inférence : les **connaissances scientifiques**.

Malheureusement, l'utilisation de la déduction peut s'avérer extrêmement complexe dans certaines situations et son approximation peut devenir moins précise qu'une induction. Sans prétendre justifier d'une façon rigoureuse l'appel à l'induction, nous remarquons son utilisation à grande échelle dans les recherches en parole (qu'il s'agisse des techniques d'analyse exploratoire, de l'apprentissage connexioniste ou des chaînes de Markov). Notre objectif est de soutenir l'intérêt des méthodes d'induction symbolique dans ce contexte.

Nous considérons que l'utilisation des techniques d'induction symbolique, qui ne modifient généralement pas la représentation des exemples familière aux chercheurs et qui fournissent des résultats directement interprétables dans le cadre d'une théorie donnée, peut s'avérer très utile dans la prise en compte des connaissances scientifiques et même dans la formulation d'hypothèses qui peuvent évoluer vers ce statut.

Selon nous, les **connaissances expertes** (moins sûres et formelles que celles scientifiques mais cependant utilisées avec succès par les experts humains) peuvent également apporter une aide considérable aux processus d'apprentissage symbolique et peuvent bénéficier à leur tour des propositions d'amélioration issues de l'analyse automatique de volumes importants de données.

La section 3.4 présente nos propositions pour la réalisation d'un environnement d'acquisition et de mise au point de connaissances, fondée sur une étroite coopération entre l'expert/chercheur et les techniques d'apprentissage automatique.

### 3.1.4 Conclusion

Les observations présentées dans les sections précédentes, sur les problèmes de recherche en parole, mettent en évidence certaines particularités qui devront être prises en compte par les méthodes d'apprentissage symbolique pour augmenter leur utilisabilité dans ce domaine. Il faut cependant remarquer que ces contraintes dépassent le cadre strict des recherches en parole et donc, les avancées dans ces directions pourront s'avérer bénéfiques pour une catégorie beaucoup plus large d'applications.

## 3.2 PmBc : apprentissage dans les domaines numériques fondé sur la similarité des exemples

PmBc se situe dans un axe de recherche relativement récent dans l'apprentissage automatique, généralement connu sous le nom d'induction d'hyperrectangles (section 2.3.3). Pour la recherche en communication parlée, ces algorithmes présentent l'avantage de produire des représentations compactes et directement compréhensibles pour les humains (par rapport aux techniques connexionistes ou markoviennes), tout en restant proches du raisonnement à base de prototypes, communément rencontré dans ce domaine. Les spécificités du domaine de la parole nous ont conduits à concentrer nos recherches essentiellement sur les espaces numériques, bien que les techniques d'induction d'hyperrectangles aient été adaptées par certains auteurs [Wettschereck & Dietterich, 1995] pour prendre également en compte les attributs symboliques.

Cette section situe notre algorithme par rapport aux approches d'induction d'hyperrectangles présentées dans la section 2.3.3 et fournit des résultats expérimentaux pour appuyer cette analyse. Pour mieux apprécier les performances de ces approches par rapport à des algorithmes plus connus dans le domaine de l'apprentissage symbolique supervisé, nous présentons aussi une comparaison expérimentale avec C4.5 et CN2. A la fin de cette présentation nous allons essayer de mettre en évidence les similitudes et les différences entre PmBc et d'autres approches (symboliques ou statistiques) en apprentissage.

### 3.2.1 Présentation de PmBc

L'algorithme que nous proposons, PmBc, diffère des approches similaires présentées dans la section 2.3.3 par deux caractéristiques importantes :

- la représentation utilisée : des hyperrectangles qui se recouvrent éventuellement (mais ne sont pas imbriqués) et qui mettent en évidence deux types de zones d'incertitude accrue : les zones de conflit et les zones hors règles (compte tenu de la similarité conceptuelle entre les règles de classification et les hyperrectangles, nous allons utiliser ces deux termes indifféremment). Des stratégies de classification paramétrables sont conçues pour gérer ces situations (section 3.2.1.5).

- la stratégie de recherche, apparentée à celle utilisée dans la méthodologie "de l'étoile" (décrite dans [Michalski, 1983]) par sa focalisation itérative sur un exemple particulier (le *noyau*) et l'exploration des généralisations possibles de cet exemple, en quête d'une représentation optimale (ici, dans le sens du nombre d'exemples positifs couverts). Dans PmBc, la recherche est réalisée d'une manière ascendante (en agglutinant des exemples autour du noyau, sur des critères de similarité) et irrévocable.

Le pseudo-code de l'algorithme d'apprentissage et de la stratégie de classification de PmBc sont présentés dans les figures 3-1 et 3-2. Certains choix de conception (signalés en gras dans les figures) nécessitent des explications supplémentaires et seront détaillées dans les sections suivantes.

**Normalisation** des attributs des exemples d'apprentissage dans l'intervalle [0,1]

*BaseRègles*={}

Tant qu'il reste des exemples non-couverts par les règles

  **Choix d'un exemple *N* (le *noyau*)** parmi les exemples non-couverts. Soit *C* la classe de *F*.

  *Règle*=*N*

  Pour chaque exemple *E* de *C* (**dans l'ordre de leur distance** par rapport à *N*)

        Tentative de généralisation de la règle courante : *Règle'*=**Généralisation(*Règle*,*E*)**

        Si *Règle'* ne couvre aucun exemple d'une autre classe que *C*

                *Règle*=*Règle'*

  *BaseRègles*=*BaseRègles*∪{*Règle*}

Dénormalisation des constantes dans les conditions des règles

**Figure 3-1** Algorithme d'apprentissage

Vérification de l'applicabilité de toutes les règles

Si les règles applicables sont en accord sur la classe de l'exemple à classer

Retourne la classe prédite par les règles

Si les règles applicables sont en désaccord

Applique la **stratégie de résolution des conflits**

Si pas de règle applicable

Applique la **stratégie de résolution des cas hors règles**

**Figure 3-2** Stratégie de classification

Les différences principales entre PmBc et BNGE sont liées aux particularités des représentations utilisées par les deux algorithmes. Notre proposition semble contradictoire avec l'analyse de [Wettschereck & Dietterich, 1995] et demande donc une justification de ces différences. Cette analyse, qui est à la base de BNGE, part de l'observation que la création d'hyperrectangles qui se recouvrent partiellement nuit aux performances de classification de NGE. Des résultats expérimentaux sont présentés pour appuyer cette observation. Cependant, la stratégie de résolution des conflits n'est jamais remise en question, ce qui restreint la généralité de la conclusion. En effet, dans toutes ces expériences, les conflits sont tranchés en faveur de l'hyperrectangle d'aire minimale et rien ne prouve qu'une autre stratégie ne pourrait pas aboutir à de meilleurs résultats.

Nous considérons, au contraire, que l'acceptation des recouvrements entre les hyperrectangles peut faciliter la construction de représentations plus compactes et plus robustes par rapport aux variations dans la distribution des exemples. Les recouvrements (ainsi que les régions en dehors des hyperrectangles) correspondent à des zones avec un degré accru d'incertitude qui devront être traitées au moment de la classification en fonction des buts poursuivis dans chaque tâche.

Pour appuyer cette remarque, deux situations correspondant à des petites variations dans la distribution des exemples (présence ou absence des deux exemples figurés en blanc) sont analysées dans la figure 3-3. Cette variation oblige BNGE à créer deux représentations très différentes et à utiliser chaque fois 3 hyperrectangles pour la classification. Une représentation plus simple et plus robuste (figurée en pointillé) est obtenue par PmBc en utilisant seulement deux hyperrectangles qui se recouvrent partiellement. Par rapport à cette représentation, BNGE est obligé de fragmenter davantage les exemples et, donc, de se baser sur un nombre réduit d'exemples dans la construction de chacun des hyperrectangles.

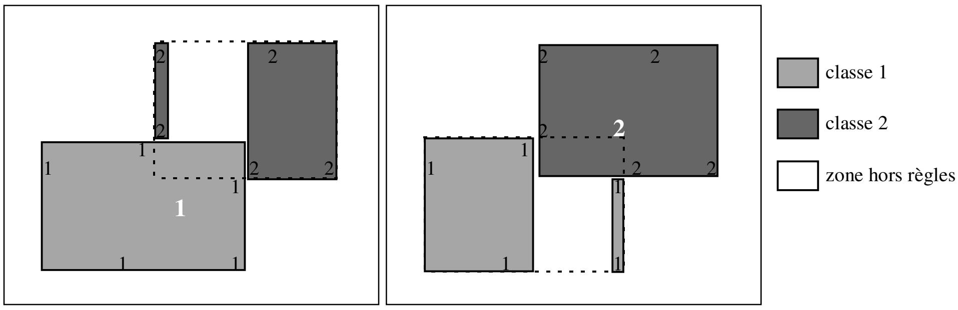



**Figure 3-3** Classificateurs produits par BNGE

### 3.2.1.1 Normalisation des exemples

L'utilisation par PmBc d'une mesure de similarité entre les exemples impose la normalisation des valeurs des attributs pour les ramener dans la même plage de variation. Nous utilisons une normalisation linéaire ( $v_{normalisée} = \frac{v - min_v}{max_v - min_v}$ ), en supposant que la différence des valeurs est une bonne mesure de distance pour chaque attribut figurant dans la description des exemples. Si des connaissances du domaine contredisent cette supposition, il reste à la charge de l'utilisateur d'appliquer les pré-traitements appropriés (le passage à une échelle logarithmique, par exemple).

Les coefficients de normalisation dépendent des valeurs extrêmes rencontrées dans l'ensemble d'apprentissage. Afin de restituer aux règles leur interprétabilité directe, les constantes apparaissant dans leurs conditions sont dénormalisées à la fin de l'apprentissage.

### 3.2.1.2 Choix du noyau

PmBc généralise les règles en les étendant autour d'un exemple particulier, le noyau, choisi à leur création parmi les exemples non-couverts par les règles déjà construites. Comme dans beaucoup d'autres algorithmes qui utilisent des noyaux pour initialiser la recherche (les nuées dynamiques [Diday *et al.*, 1983], AQ15 [Michalski *et al.*, 1986]), ce choix est fait dans PmBc d'une manière aléatoire. Même si les résultats obtenus en utilisant cette sélection aléatoire s'avèrent satisfaisants dans la plupart des cas, il faut observer que les représentations produites peuvent varier en fonction des exemples qui sont choisis comme noyaux à chaque exécution.

Certains auteurs [Kasif *et al.*, 1993] considèrent ce type de comportement comme avantageux parce que, suite à des exécutions multiples sur le même ensemble d'apprentissage, l'utilisateur pourrait choisir le classificateur le mieux adapté à ses besoins (par exemple le moins complexe) et parce que plusieurs classificateurs ainsi produits pourraient être utilisés conjointement pour des prises de décision collectives.

Cependant, dans certaines applications, cette variabilité pourrait s'avérer moins souhaitable. Nous orientons donc également nos recherches vers la conception d'autres procédures de sélection, qui devraient permettre à PmBc de produire systématiquement les mêmes résultats pour un même ensemble d'exemples d'apprentissage (quel que soit l'ordre de présentation des exemples), tout en améliorant ses performances globales.

#### 3.2.1.3 Généralisation des règles

| |
|---|
| pour chaque attribut $i$ |
| si $min_i > e_i$ alors $min'_i = e_i$ |
| sinon $min'_i = min_i$ |
| si $max_i < e_i$ $max'_i = e_i$ |
| sinon $max'_i = max_i$ |
| retourne $R' = ([min'_1, max'_1],[min'_2, max'_2],\cdots,[min'_n, max'_n])$ |

**Figure 3-4** Procédure de généralisation

Dans PmBc, les tentatives de généralisation d'une règle ont comme but de couvrir un nouvel exemple positif, tout en gardant les modifications minimales pour ne pas empêcher des généralisations ultérieures. En tenant compte de ces contraintes, la généralisation d'une règle $R = ([min_1, max_1],[min_2, max_2],\cdots,[min_n, max_n])$ pour couvrir un exemple $E = (e_1, e_2, \cdots, e_n)$ est définie dans la figure 3-4.

#### 3.2.1.4 Distance

Dans la plupart des algorithmes d'apprentissage symbolique, la recherche des représentations est guidée par des fonctions d'évaluation qui apprécient la qualité des modifications possibles et orientent l'exploration de l'espace des représentations dans la direction qui semble, localement, la plus prometteuse (voir section 2.2). Nous avons présenté, dans la section 2.3, quelques exemples de fonctions d'évaluation, fondées sur l'entropie (C4.5, FOIL) ou sur l'estimation probabiliste des fréquences des exemples (CN2). Dans l'esprit de l'induction d'hyperrectangles, nous proposons, dans PmBc, une stratégie de recherche qui repose explicitement sur l'hypothèse de la similarité (*similarity assumption*) : les exemples appartenant à la même classe (ou à la même sous-classe, dans le cas des représentations disjonctives) sont caractérisés par des descriptions similaires.

La notion de similarité, et donc de distance, est essentielle dans notre algorithme. Dans ses tentatives de généralisation d'une règle, PmBc commence toujours la recherche par les exemples les plus proches de son noyau (figure 3-5). Cette heuristique va donc préférer des modifications minimales et aussi en accord avec l'hypothèse de similarité, par rapport à des changements plus importants qui risquent de rendre impossibles des généralisations ultérieures de la règle.

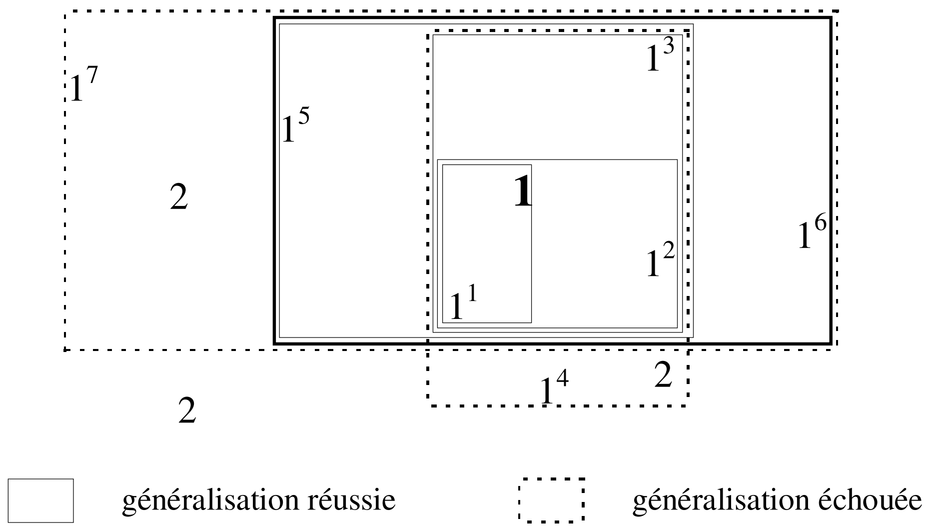



**Figure 3-5** Illustration de la stratégie de recherche de PmBc : l'exemple en gras est le noyau et les exposants indiquent l'ordre dans lequel les exemples sont examinés

Une première version de PmBc utilisait la "distance de Tchebicheff" (maximum de variation sur les axes) pour des raisons d'efficacité. Cette formule génère des lieux géométriques d'égale distance de forme hypercubique, ce qui s'avère intéressant pour le monde d'hyperrectangles exploré par PmBc. Elle présente la propriété suivante :

Soient $e_1$, $e_2$, ..., $e_n$ une chaîne d'exemples ordonnés par rapport à leur distance au noyau (c.à.d. $d(N, e_1) \leq d(N, e_2) \leq \cdots \leq d(N, e_n)$) et $R$ la règle obtenue par la généralisation du noyau pour couvrir tous ces exemples (c.à.d. $R_1$ = *Généralisation*($N$, $e_1$), $R_2$ = *Généralisation*($R_1$, $e_2$), ..., $R$ = *Généralisation*($R_{n-1}$, $e_n$)). Alors, si $d(N, c) > d(N, e_n)$, $c \notin R$.

Cette propriété affirme donc qu'aucun exemple négatif plus éloigné du noyau que le dernier exemple positif rajouté ne peut infirmer la validité de la règle. Elle peut être utilisée de deux façons pour accélérer la vérification des tentatives de généralisations par PmBc. Premièrement, la vérification des règles peut être arrêtée en toute sécurité au moment où tous les exemples négatifs qui restent à vérifier se trouvent à une distance du noyau supérieure à celle du dernier exemple positif qui à contribué à sa généralisation.

La deuxième possibilité d'utilisation de cette propriété dans PmBc est plus complexe mais apporte un gain d'efficacité plus important. Les règles en cours de construction sont accompagnées par un ensemble de *contraintes* (représentées à leur tour sous forme de règles) qui délimitent les régions de l'espace des exemples vers lesquelles les généralisations de la règle courante sont rendues impossibles par les exemples négatifs. La règle et ses contraintes établissent donc des limites similaires aux ensembles *S* et *G* de l'espace de versions [Mitchell, 1977].

Dans cette version, le traitement de chaque exemple est effectué une seule fois pour chaque règle, dans l'ordre de sa distance par rapport au noyau. Si l'exemple considéré est négatif et n'est pas couvert par les contraintes existantes, il génère une nouvelle contrainte. S'il est positif et en dehors des contraintes déjà posées, il participe à la généralisation de la règle courante, sans aucune vérification supplémentaire. Les généralisations de la règle nécessitent en plus la réactualisation de ses contraintes.

La version de PmBc qui utilise cette distance a été appliquée, avec de bons résultats, à des problèmes pratiques de recherche en parole [Caillaud et al., 1996]. Nous nous sommes cependant retournés récemment vers l'utilisation de la distance euclidienne (c'est cette nouvelle version qui sera utilisée dans l'étude expérimentale). Cette formule, qui semble plus appropriée pour l'estimation de la similarité dans les espaces numériques, apporte un léger gain de précision aux classificateurs produits, aux dépens de l'efficacité du calcul.

La vérification des tentatives de généralisation bénéficie toujours du tri des exemples par rapport à leur distance au noyau. Ceci s'explique par le fait que les exemples négatifs les plus proches sont ceux qui risquent de poser le plus de problèmes à la généralisation et donc, en les examinant avant les autres, PmBc peut arrêter plus tôt la vérification des règles qui vont s'avérer incorrectes.

Pour l'instant, PmBc n'accepte pas les tentatives de généralisation qui couvrent des exemples négatifs. Cependant, à cause de sa procédure de généralisation très conservatrice, les exemples négatifs qui correspondent à du bruit vont générer des hyperrectangles de taille réduite, dont l'influence pour la classification de nouveaux exemples sera assez faible. Ce choix nous a été dicté par des considérations d'efficacité de calcul. Il est également possible de rajouter à PmBc la possibilité de couvrir un nombre réduit d'exemples négatifs, aux dépens du temps d'exécution (on ne pourra plus, alors, arrêter la vérification des tentatives de généralisation au premier contre-exemple trouvé).

#### 3.2.1.5 Stratégies de classification

PmBc doit gérer des situations de conflit et des cas hors règles au moment de la classification (figure 3-6).

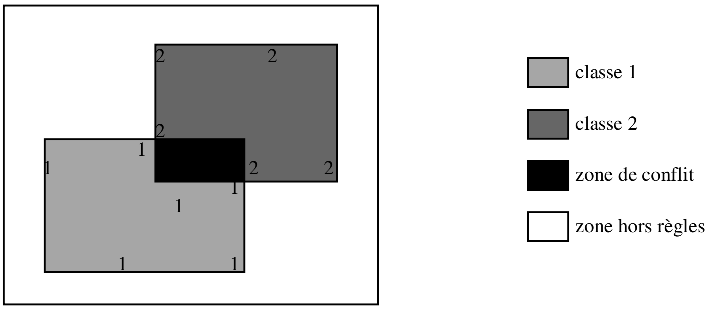



**Figure 3-6** Prise de décision dans PmBc

En fonction des buts poursuivis dans la tâche d'apprentissage, on peut souhaiter des comportements différents du classificateur dans les zones de conflit ou hors règles. Par exemple, si le risque associé aux manques de détection est beaucoup plus important pour une des classes que pour les autres (comme c'est le cas dans [Christiensen, 1992], par exemple), le classificateur devrait donner préférence à la classe respective en cas de doute.

En effet, dans l'implémentation actuelle de PmBc, les stratégies de résolution des conflits et des cas hors règles constituent des paramètres d'exécution et peuvent être choisies par l'utilisateur dans une bibliothèque de stratégies prédéfinies. Ces stratégies ont accès à une liste des priorités des classes, fournies explicitement par l'utilisateur ou assignées automatiquement par l'algorithme (proportionnellement à la distribution des exemples dans l'ensemble d'apprentissage).

Pour la plupart des tâches d'apprentissage, l'appréciation de la qualité d'un classificateur est réalisée sur la base du nombre total d'erreurs commises. La stratégie qui s'est avérée empiriquement la plus performante dans l'optimisation de ce critère réalise un vote en situation de conflit et utilise la règle la plus proche pour les cas hors règles. Les ballottages sont tranchés en faveur de la classe la plus prioritaire. Les résultats expérimentaux rapportés dans la section 3.2.2 ont été obtenus en utilisant cette stratégie.

### 3.2.2 Résultats expérimentaux

#### 3.2.2.1 Algorithmes utilisés pour la comparaison

Dans le domaine de l'apprentissage automatique, une des méthodes les plus utilisées pour évaluer l'intérêt d'un nouvel algorithme consiste en la réalisation d'études expérimentales comparatives qui confrontent les performances de l'algorithme proposé (en fonction de plusieurs critères comme : sa précision de classification, la complexité des représentations produites ou sa vitesse d'exécution) à d'autres approches du domaine. Parmi la multitude d'algorithmes d'apprentissage symbolique très connus dans ce domaine, nous avons dû réaliser un choix, forcément incomplet, d'un nombre limité de représentants, sur des critères que nous allons préciser dans les paragraphes suivants.

Pour renforcer l'objectivité de la comparaison, nous avons utilisé, pour les algorithmes choisis, les implémentations directement proposées par leurs auteurs. Il faut préciser que, à la différence de PmBc, le comportement de certains de ces algorithmes peut être modifié par des paramètres optionnels. Il existe cependant, pour chacun de ces algorithmes, une configuration de paramètres par défaut, conseillée par leurs auteurs, et qui, d'après eux (et les résultats publiés), s'avère performante dans la plupart des cas. Ce sont ces configurations (qui seront détaillées dans les paragraphes suivants) que nous avons utilisées dans nos expériences.

L'autre possibilité, que nous aurions pu adopter pour gérer les configurations des paramètres, consiste dans leur réglage individuel, sur chacune des tâches d'apprentissage considérées. Ce réglage ne fait pas partie, à proprement parler, des algorithmes d'apprentissage et il existe un grand nombre de méthodes qui peuvent être utilisées pour le réaliser. Ces méthodes vont de l'exploitation des connaissances du domaine ou des objectifs particuliers poursuivis par l'utilisateur, au choix des paramètres par essais successifs ou par l'utilisation de méthodes d'optimisation variées (souvent sujettes aux problèmes causés par les optima locaux). Ces méthodes d'optimisation de paramètres ne sont pas toujours à la portée de l'utilisateur final dont les besoins sont au centre de nos recherches. En outre, un réglage laborieux de paramètres affaiblit l'argument de la simplicité d'utilisation, souvent invoqué pour soutenir l'intérêt des algorithmes d'apprentissage symbolique par rapport aux approches connexionistes (pour toutes ces raisons PmBc est dépourvu de tout paramètre à régler).

Selon nous, l'utilisation d'une technique de réglage de paramètres aurait donc introduit des biais supplémentaires et aurait alourdi les expériences (déjà très consommatrices de temps) sans pour cela changer de beaucoup les résultats finaux. Nous sommes cependant conscients que ce choix peut être contesté et avertissons le lecteur que tous les résultats expérimentaux rapportés dans la section suivante doivent être interprétés comme étant valables pour l'utilisation des paramètres par défaut des différents algorithmes.

Au moment du choix des algorithmes pour la comparaison, il nous est apparu comme indispensable de confronter PmBc à d'autres algorithmes de la même famille : l'induction d'hyperrectangles. Nous avons choisi BNGE pour deux raisons principales. Comme précisé par ses auteurs, il est issu de la lignée de NGE, probablement le plus connu et le plus cité des algorithmes d'induction d'hyperrectangles, et la supériorité de ses performances par rapport à son ancêtre a fait déjà l'objet d'études expérimentales rigoureuses. Ensuite, il est fondé sur des principes similaires à PmBc, ce qui facilite la mise en évidence des choix de conception qui expliquent les différences de performances entre ces deux algorithmes (voir la discussion de la section 3.2.1). Nous avons utilisé dans nos expériences la version standard de l'algorithme[*], telle qu'elle est décrite dans la section 2.3.3. Par rapport à cette version, [Wettschereck &

---

[*] Nous remercions Dietrich Wettschereck et Peter Clark pour leur disponibilité et pour leurs conseils qui nous ont aidés dans la mise en place des expériences avec BNGE et CN2.

Dietterich, 1995] discutent également différentes extensions possibles, comme la pondération différente de l'importance des attributs ou l'utilisation de la méthode des plus proches voisins pour la classification.

Les algorithmes d'induction d'hyperrectangles, comme BNGE ou PmBc constituent une addition relativement récente à l'arsenal des méthodes d'apprentissage symbolique supervisé. Il nous semble donc nécessaire de situer leurs performances de classification par rapport à des techniques éprouvées. Malheureusement, peu d'études comparatives de ce type sont actuellement disponibles. [Salzberg, 1991] rapporte des résultats expérimentaux favorables à NGE par rapport à quelques algorithmes "classiques" (AQ15 [Michalski et al., 1986], CART [Breiman et al., 1984], rétro-propagation du gradient, etc.) sur trois bases de données d'apprentissage. Les performances de NGE sont remises en question par [Wettschereck & Dietterich, 1995], mais cet article se concentre surtout sur des comparaisons avec les méthodes des plus proches voisins.

Le deuxième algorithme que nous avons choisi pour cette comparaison est donc C4.5 (présenté dans la section 2.3.1), car cet algorithme a été utilisé intensivement et a fait la preuve de son efficacité dans des domaines d'application très variés. Nous le considérons comme représentatif pour une large catégorie d'algorithmes d'apprentissage qui produisent des arbres de décision, en utilisant une stratégie de type "diviser pour régner" et des fonctions d'évaluation fondées sur l'entropie des exemples. Les résultats expérimentaux rapportés dans la section suivante utilisent la version la plus récente (*Release 8* [Quinlan, 1996]) et la configuration standard de l'algorithme : construction d'un arbre de décision élagué en utilisant une valeur de 25% pour le paramètre "niveau de confiance" et l'élimination des tests qui proposent un nombre de branches ayant plus de deux exemples inférieur à deux.

Un troisième algorithme très connu qui sera utilisé dans les comparaisons est CN2 (présenté dans la section 2.3.2), qui produit des classificateurs à base de règles (son formalisme de représentation des résultats est donc plus proche de celui de PmBc). La version considérée comme la plus performante par son auteur* construit des règles indépendantes, en utilisant l'estimateur de Laplace pour évaluer leur qualité et sans faire appel à l'élagage fondé sur le test de confiance statistique (l'estimateur de Laplace pénalise directement les règles à très faibles effectifs). Le paramètre "taille de l'étoile" a été laissé à sa valeur par défaut (5 règles).

#### 3.2.2.2 Données utilisées

PmBc a été développé dans le cadre de nos recherches sur l'utilisation des techniques d'apprentissage en parole et a été utilisé plus particulièrement pour l'étude de la segmentation lexicale fondée sur l'utilisation de la prosodie [Caillaud et al., 1996]. Nous soutenons cependant que son applicabilité dépasse le cadre stricte de ces recherches et, pour réaliser une étude expérimentale objective, nous avons décidé de comparer les résultats des algorithmes sur des bases de données généralement utilisées dans le domaine de l'apprentissage automatique.

L'UC-Irvine met à la disposition des chercheurs un ensemble de bases collectées dans différents domaines [Murphy & Aha, 1993]. Nous avons sélectionné parmi les bases disponibles celles qui ne comportaient que des attributs numériques. De cette sélection sont ressorties 16 bases. Pour l'une d'entre elles, il nous a semblé difficile de faire honnêtement une comparaison car, vue sa taille (58 000 exemples), les temps d'apprentissage et de test risquaient d'être prohibitifs. Il aurait sans doute été nécessaire d'utiliser pour son apprentissage des techniques telles que le « fenêtrage » (apprentissage incrémental utilisant à chaque pas une partie réduite de l'ensemble d'apprentissage [Quinlan, 1993]) qui aurait pu introduire un biais supplémentaire dans le résultat des différents algorithmes.)

La comparaison des algorithmes sera donc effectuée sur 15 bases de données (voir annexe C) qui présentent une grande variété par les domaines dont elles sont issues, par leur nombre d'exemples (150-2000), leur nombre d'attributs (4-60) et leur nombre de classes (2-26).

Une de ces bases de données (*breast-w*) contient des exemples avec des valeurs d'attributs manquantes. Un grand nombre de stratégies ont été proposées pour gérer ce type d'exemples et la plupart d'entre elles pourraient être utilisées dans PmBc. L'utilisation des stratégies différentes par les trois algorithmes aurait introduit une variable supplémentaire dans nos expériences, en rendant plus difficile une comparaison directe des stratégies de recherche et des représentations utilisées. Nous avons donc préféré enlever de cette base de données tous les exemples incomplets.

#### 3.2.2.3 Protocole expérimental

Le protocole expérimental ainsi que les méthodes de comparaison utilisées dans une telle expérience doivent être soigneusement choisis afin de ne pas biaiser les résultats obtenus.

[Flexer, 1996] présente un ensemble minimal de conditions à satisfaire pour obtenir des résultats significatifs pour ce type de comparaison :

- utilisation d'ensembles d'exemples différents pour l'apprentissage et le test de chacun des algorithmes,
- réalisation de plusieurs apprentissages/tests,
- présentation des résultats en terme de nombre de tests, moyenne et écart-type (ou l'intervalle de confiance de la moyenne rapportée)
- utilisation d'un test de confiance statistique pour la comparaison de plusieurs algorithmes.

L'étude expérimentale des algorithmes d'apprentissage concerne d'habitude l'évaluation de plusieurs critères de qualité. Nous allons analyser ici les performances de classification, la complexité des représentations produites et la vitesse d'exécution. Il faut remarquer cependant qu'il existe également d'autres critères de qualité importants qui peuvent être utilisés pour comparer les algorithmes d'apprentissage comme, par exemple, la compréhensibilité pour les humains de leurs résultats ou leur capacité de prévoir le taux d'erreur qui sera enregistré sur un ensemble de test indépendant.

Notre protocole expérimental est le suivant :

**Etape 1** : Chaque base est mélangée aléatoirement puis découpée en deux ensembles suivant les proportions 80% / 20%.

**Etape 2** : Les trois algorithmes utilisent le premier ensemble pour l'apprentissage et sont testés sur le deuxième.

Ces deux étapes sont répétées successivement 50 fois, les moyennes et les écarts types sont alors calculés.

Pour la comparaison des algorithmes, le test de confiance statistique (test de Student pour des moyennes d'échantillons appariés [Saporta, 1990]) est calculé paire par paire. Il indique la probabilité pour que la moyenne des taux d'erreur de l'un des algorithmes soit égale à la moyenne de l'autre.

Pour cette expérience, nous avons fixé empiriquement le seuil de cinquante essais successifs. Ce seuil nous permet d'obtenir une confiance statistique supérieure à 95% ($p<0.05$) dans la plupart des comparaisons. Pour les bases qui ne satisfont pas cette condition, une augmentation même importante du nombre d'essais ne nous a pas permis d'arriver à cette valeur de confiance.

Dans ces tests, le critère de comparaison entre les algorithmes que nous utiliserons sera le nombre de bases pour lequel un algorithme aura une moyenne supérieure à une autre avec une confiance statistique suffisante (au moins 95%).

Les figures 3-7, 3-8 et 3-9 présentent pour chaque paire d'algorithmes :

- la moyenne et l'écart-type du taux d'erreur,
- >, <, = : nombre de fois, parmi les cinquante essais, où la moyenne du taux d'erreur du premier algorithme a été respectivement supérieure, inférieure et égale à la moyenne du second,
- t, p : la variable de Student et la probabilité d'égalité des moyennes, calculées par le test de Student pour échantillons appariés,
- Rapport : rapport entre la moyenne du taux d'erreur du premier algorithme et celle du deuxième,
- Meilleur : nom de l'algorithme qui a obtenu la moyenne la plus basse du taux d'erreur,
- Statis$^{\text{ent.}}$ : indique si les deux moyennes sont différentes avec un taux de confiance statistique suffisant.

#### 3.2.2.4 Performances de classification

##### 3.2.2.4.1 Comparaison PmBc-BNGE

| Base | PmBc | | BNGE | | > | < | = | t | p | Rapport | Meilleur | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Moy | ± | Moy | ± | | | | | | | | |
| *breast-w* | 4,09 | 1,57 | 4,09 | 1,49 | 21 | 27 | 2 | 0 | 0,997 | 1,00 | *Egalité* | |
| *bupa* | 34,49 | 5,04 | 36,41 | 6,37 | 20 | 30 | 0 | -2,26 | 0,028 | 0,95 | **PmBc** | Oui |
| *diabetes* | 26,00 | 3,03 | 27,86 | 3,65 | 19 | 31 | 0 | -3,15 | 0,003 | 0,93 | **PmBc** | Oui |
| *glass* | 31,49 | 5,46 | 33,67 | 6,70 | 20 | 30 | 0 | -2,36 | 0,022 | 0,94 | **PmBc** | Oui |
| *ionosphere* | 7,61 | 2,76 | 11,07 | 3,98 | 12 | 38 | 0 | -5,61 | **<0,001** | 0,69 | **PmBc** | Oui |
| *iris* | 3,27 | 4,12 | 3,20 | 3,56 | 8 | 15 | 27 | 0,23 | 0,822 | 1,02 | BNGE | |
| *letter* | | | | | | | | | | | | |
| *new-thyroid* | 4,19 | 2,44 | 5,49 | 3,07 | 15 | 35 | 0 | -3,45 | 0,001 | 0,76 | **PmBc** | Oui |
| *segment* | 3,12 | 0,80 | 4,44 | 1,02 | 6 | 44 | 0 | -8,58 | **<0,001** | 0,70 | **PmBc** | Oui |
| *sonar* | 28,81 | 7,46 | 32,38 | 7,50 | 20 | 30 | 0 | -2,67 | 0,010 | 0,89 | **PmBc** | Oui |
| *vehicle* | 28,74 | 2,55 | 32,68 | 3,76 | 9 | 41 | 0 | -6,89 | **<0,001** | 0,88 | **PmBc** | Oui |
| *vowel* | 13,57 | 2,60 | 16,65 | 2,94 | 10 | 40 | 0 | -5,7 | **<0,001** | 0,82 | **PmBc** | Oui |
| *waveform* | | | | | | | | | | | | |
| *waveform40* | | | | | | | | | | | | |
| *wine* | 4,78 | 4,38 | 4,17 | 2,71 | 18 | 28 | 4 | 1,09 | 0,280 | 1,15 | BNGE | |

**Figure 3-7** Comparaison PmBc-BNGE (taux d'erreur)

Cette comparaison n'a pas pu être totalement menée à bien. En effet, la version binaire pour architectures Sun que l'auteur a mis à la disposition du public prend beaucoup de temps sur les bases de données importantes (une semaine pour l'apprentissage et le test sur les 20000 exemples de la base *letter* sur une machine hyper Sparc moyennement chargée). Ceci semble en contradiction avec les observations sur les temps d'exécution rapportées dans [Wettschereck & Dietterich, 1995]. Cette limitation ne nous a pas permis de mener à leurs termes les cinquante essais sur les bases importantes : *letter*, *waveform*, *waveform40*.

On peut remarquer que, sur les bases étudiées, PmBc obtient 9 fois une moyenne statistiquement supérieure à BNGE alors que BNGE n'est jamais statistiquement supérieur à PmBc. Si l'on ne tient pas compte de la confiance statistique, là encore PmBc est supérieur car il obtient 9 fois le moindre taux d'erreur contre 2 fois pour BNGE et une égalité presque parfaite (sur *breast-w*).

#### 3.2.2.4.2 Comparaison PmBc-C4.5

| Base | PmBc | | C4.5 | | > | < | = | t | p | Rapport | Meilleur | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Moy | ± | Moy | ± | | | | | | | | |
| *breast-w* | 4,09 | 1,57 | 5,59 | 1,82 | 7 | 42 | 1 | -7,05 | **<0,001** | 0,73 | **PmBc** | Oui |
| *bupa* | 34,49 | 5,04 | 35,81 | 5,95 | 24 | 26 | 0 | -1,21 | 0,227 | 0,96 | PmBc | |
| *diabetes* | 26,00 | 3,03 | 25,65 | 3,93 | 26 | 24 | 0 | 0,77 | 0,448 | 1,01 | C45 | |
| *glass* | 31,49 | 5,46 | 34,15 | 6,25 | 17 | 33 | 0 | -2,93 | 0,005 | 0,92 | **PmBc** | Oui |
| *ionosphere* | 7,61 | 2,76 | 10,83 | 3,62 | 7 | 43 | 0 | -6,07 | **<0,001** | 0,70 | **PmBc** | Oui |
| *iris* | 3,27 | 4,12 | 4,46 | 4,14 | 14 | 22 | 14 | -2,67 | 0,010 | 0,73 | **PmBc** | Oui |
| *letter* | 5,93 | 0,39 | 12,52 | 0,54 | 0 | 50 | 0 | -105,46 | **<0,001** | 0,47 | **PmBc** | Oui |
| *new-thyroid* | 4,19 | 2,44 | 7,55 | 4,14 | 6 | 42 | 2 | -6,8 | **<0,001** | 0,55 | **PmBc** | Oui |
| *segment* | 3,12 | 0,80 | 3,40 | 0,83 | 18 | 32 | 0 | -1,98 | 0,053 | 0,92 | PmBc | |
| *sonar* | 28,81 | 7,46 | 25,62 | 8,79 | 34 | 16 | 0 | 2,21 | 0,032 | 1,12 | **C45** | Oui |
| *vehicle* | 28,74 | 2,55 | 27,02 | 3,22 | 34 | 16 | 0 | 2,97 | 0,005 | 1,06 | **C45** | Oui |
| *vowel* | 13,57 | 2,60 | 22,30 | 3,23 | 1 | 49 | 0 | -15,57 | **<0,001** | 0,61 | **PmBc** | Oui |
| *waveform* | 17,36 | 1,13 | 23,68 | 1,16 | 0 | 50 | 0 | -32,83 | **<0,001** | 0,73 | **PmBc** | Oui |
| *waveform40* | 22,95 | 1,47 | 24,58 | 1,34 | 8 | 41 | 1 | -6,49 | **<0,001** | 0,93 | **PmBc** | Oui |
| *wine* | 4,78 | 4,38 | 7,67 | 4,43 | 12 | 38 | 0 | -3,64 | 0,001 | 0,62 | **PmBc** | Oui |

**Figure 3-8** Comparaison PmBc-C4.5 (taux d'erreur)

Les résultats de cette expérience donnent 10 fois PmBc comme meilleur classificateur et seulement 2 fois C4.5, ceci en ne tenant compte que des moyennes significativement différentes. Si l'on ne tient pas compte de cette contrainte, on obtient 12 meilleures classifications pour PmBc et seulement 3 pour C4.5.

On peut remarquer ici que [Quinlan, 1996] se base sur un autre critère (la moyenne du rapport des erreurs pour l'ensemble des bases de données) pour soutenir la supériorité de la *Release 8* par rapport à la *Release 7* de C4.5. Même si la validité de ce critère nous semble discutable, nous l'avons évalué par souci de complétude pour la comparaison présente. Dans cette expérience on obtient avec PmBc une réduction de 19% de la moyenne des erreurs commises par C4.5, ce qui dépasse largement les 4% invoqués dans [Quinlan, 1996] pour valider la supériorité de la *Release 8*. Le rapport entre les taux d'erreur de PmBc et C4.5 varie entre 47% (sur la base *letter*) et 112% (sur la base *sonar*).

Ces résultats montrent un avantage important pour notre approche en termes de performances de classification, mais cet avantage doit être pondéré par des considérations de temps de calculs (voir section 3.2.2.6).

#### 3.2.2.4.3 Comparaison PmBc-CN2

| Base | C4.5 Moy | ± | BNGE Moy | ± | > | < | = | t | p | Rapport | Meilleur | Statis^ent |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 4,09 | 1,57 | 5,99 | 2,08 | 3 | 46 | 1 | -8,37 | **<0,001** | 0,68 | **PmBc** | Oui |
| *bupa* | 34,49 | 5,04 | 34,31 | 4,28 | 22 | 28 | 0 | 0,23 | 0,820 | 1,01 | CN2 | |
| *diabetes* | 26,00 | 3,03 | 26,26 | 3,30 | 28 | 22 | 0 | -0,52 | 0,599 | 0,99 | PmBc | |
| *glass* | 31,49 | 5,46 | 33,82 | 7,48 | 18 | 32 | 0 | -2,04 | 0,045 | 0,93 | **PmBc** | Oui |
| *ionosphere* | 7,61 | 2,76 | 9,13 | 3,54 | 16 | 34 | 0 | -2,81 | 0,007 | 0,83 | **PmBc** | Oui |
| *iris* | 3,27 | 4,12 | 5,33 | 4,47 | 6 | 25 | 19 | -5,10 | **<0,001** | 0,61 | **PmBc** | Oui |
| *letter* | 5,93 | 0,39 | 29,13 | 1,42 | 0 | 50 | 0 | -122,05 | **<0,001** | 0,20 | **PmBc** | Oui |
| *new-thyroid* | 4,19 | 2,44 | 4,25 | 2,20 | 22 | 26 | 2 | -0,18 | 0,854 | 0,98 | PmBc | |
| *segment* | 3,12 | 0,80 | 13,77 | 3,16 | 0 | 50 | 0 | -24,84 | **<0,001** | 0,23 | **PmBc** | Oui |
| *sonar* | 28,81 | 7,46 | 28,29 | 6,63 | 24 | 26 | 0 | 0,41 | 0,684 | 1,02 | CN2 | |
| *vehicle* | 28,74 | 2,55 | 38,08 | 5,55 | 2 | 48 | 0 | -12,23 | **<0,001** | 0,75 | **PmBc** | Oui |
| *vowel* | 13,57 | 2,60 | 39,87 | 3,71 | 0 | 50 | 0 | -52,54 | **<0,001** | 0,34 | **PmBc** | Oui |
| *waveform* | 17,36 | 1,13 | 30,92 | 1,59 | 0 | 50 | 0 | -49,86 | **<0,001** | 0,56 | **PmBc** | Oui |
| *waveform40* | 19,91 | 1,24 | 32,75 | 1,84 | 0 | 50 | 0 | -41,03 | **<0,001** | 0,61 | **PmBc** | Oui |
| *wine* | 4,78 | 4,38 | 7,06 | 3,36 | 10 | 40 | 0 | -3,20 | 0,002 | 0,68 | **PmBc** | Oui |

**Figure 3-9** Comparaison PmBc-CN2 (taux d'erreur)

Les résultats de PmBc face à CN2 sont comparables à ceux obtenus dans sa confrontation avec BNGE : toutes les différences statistiquement significatives (11/15) sont en faveur de PmBc et le score global (13-2) confirme sa supériorité en terme de performances de classification.

### 3.2.2.5 Complexité des représentations produites

La plupart des algorithmes d'apprentissage automatique utilisent, d'une façon plus ou moins explicite, le célèbre critère de Occam qui favorise les représentations les plus simples. Pour cette raison, les études comparatives d'algorithmes analysent souvent la complexité des représentations produites.

Dans le cas présent, nous avons dû nous limiter à une comparaison entre PmBc, BNGE et CN2. Le quatrième algorithme, C4.5, utilise un formalisme très différent, pas directement comparable avec les trois autres (la complexité des arbres de décision produits est rapportée dans l'Annexe D).

Les figures 3-10 et 3-11 présentent les résultats de cette comparaison, dans un format similaire à celui utilisé pour les comparaisons des performances de classification. La complexité des représentations est évaluée à partir du nombre de règles.

| Base | PmBc Moy | ± | BNGE Moy | ± | > | < | = | t | p | Rapport | Moins complexe | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 20,56 | 2,64 | 24,24 | 2,4 | 3 | 44 | 3 | -9,31 | **<0,001** | 0,85 | **PmBc** | **Oui** |
| *bupa* | 51,42 | 3,26 | 64,38 | 3,45 | 0 | 50 | 0 | -23,36 | **<0,001** | 0,80 | **PmBc** | **Oui** |
| *diabetes* | 81 | 3,86 | 117,64 | 7,43 | 0 | 50 | 0 | -34,59 | **<0,001** | 0,69 | **PmBc** | **Oui** |
| *glass* | 25,86 | 1,43 | 34,14 | 2,93 | 0 | 50 | 0 | -17,17 | **<0,001** | 0,76 | **PmBc** | **Oui** |
| *ionosphere* | 20,02 | 2,11 | 22,04 | 2,91 | 15 | 31 | 4 | -4,16 | **<0,001** | 0,91 | **PmBc** | **Oui** |
| *iris* | 8,64 | 1,55 | 8,18 | 1,53 | 16 | 2 | 32 | 3,57 | **<0,001** | 1,06 | **BNGE** | **Oui** |
| *letter* | 752,86 | 9,8 | | | | | | | | | | |
| *new-thyroid* | 9,56 | 1,16 | 10,26 | 1,41 | 9 | 26 | 15 | -3,24 | 0,002 | 0,93 | **PmBc** | **Oui** |
| *segment* | 36,84 | 2,71 | 57,88 | 5,97 | 0 | 50 | 0 | -24,67 | **<0,001** | 0,64 | **PmBc** | **Oui** |
| *sonar* | 8,08 | 0,94 | 26,28 | 3,12 | 0 | 50 | 0 | -37,93 | **<0,001** | 0,31 | **PmBc** | **Oui** |
| *vehicle* | 69,6 | 3,23 | 123,7 | 5,94 | 0 | 50 | 0 | -68,45 | **<0,001** | 0,56 | **PmBc** | **Oui** |
| *vowel* | 57,34 | 2,5 | 113,84 | 6,37 | 0 | 50 | 0 | -57,73 | **<0,001** | 0,50 | **PmBc** | **Oui** |
| *waveform* | 255,6 | 8,4 | | | | | | | | | | |
| *waveform40* | 221,5 | 7,14 | | | | | | | | | | |
| *wine* | 5,22 | 0,76 | 7 | 1,56 | 0 | 41 | 9 | -8,7 | **<0,001** | 0,75 | **PmBc** | **Oui** |

**Figure 3-10** Comparaison PmBc-BNGE (nombre de règles produites)

Tous les résultats de cette comparaison sont statistiquement significatifs. Les représentations produites par PmBc sont moins complexes que celles produites par BNGE pour 11 des 12 expériences qui ont pu être finalisées. On peut observer aussi que la dispersion du nombre de règles est plus importante pour BNGE que pour PmBc pour la plupart des bases de données. Ces résultats sont donc en accord avec les observations sur les particularités des représentations recherchées par BNGE, présentées dans la section 3.2.1.

| Base | PmBc Moy | ± | CN2 Moy | ± | > | < | = | t | p | Rapport | Moins complexe | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| breast-w | 20,56 | 2,64 | 16,08 | 1,40 | 45 | 3 | 2 | 10,76 | **<0,001** | 1,28 | **CN2** | Oui |
| *bupa* | 51,42 | 3,26 | 29,74 | 2,83 | 50 | 0 | 0 | 35,44 | **<0,001** | 1,73 | **CN2** | Oui |
| *diabetes* | 81,00 | 3,86 | 43,38 | 4,01 | 50 | 0 | 0 | 50,24 | **<0,001** | 1,87 | **CN2** | Oui |
| *glass* | 25,86 | 1,43 | 21,50 | 1,37 | 48 | 0 | 2 | 14,98 | **<0,001** | 1,20 | **CN2** | Oui |
| *ionosphere* | 20,02 | 2,11 | 16,94 | 1,36 | 44 | 1 | 5 | 9,77 | **<0,001** | 1,18 | **CN2** | Oui |
| *iris* | 8,64 | 1,55 | 7,12 | 1,27 | 43 | 3 | 4 | 9,37 | **<0,001** | 1,21 | **CN2** | Oui |
| *letter* | 752,86 | 9,80 | 254,22 | 5,57 | 50 | 0 | 0 | 307,38 | **<0,001** | 2,96 | **CN2** | Oui |
| *new-thyroid* | 9,56 | 1,16 | 8,52 | 0,91 | 30 | 4 | 16 | 5,55 | **<0,001** | 1,12 | **CN2** | Oui |
| *segment* | 36,84 | 2,71 | 37,32 | 2,38 | 13 | 25 | 12 | -1,02 | 0,308 | 0,99 | PmBc | |
| *sonar* | 8,08 | 0,94 | 26,42 | 2,50 | 0 | 50 | 0 | -53,19 | **<0,001** | 0,31 | **PmBc** | Oui |
| *vehicle* | 69,60 | 3,23 | 46,80 | 2,67 | 50 | 0 | 0 | 42,72 | **<0,001** | 1,49 | **CN2** | Oui |
| *vowel* | 57,34 | 2,50 | 61,96 | 2,90 | 4 | 44 | 2 | -8,37 | **<0,001** | 0,93 | **PmBc** | Oui |
| *waveform* | 255,66 | 8,40 | 192,80 | 7,27 | 50 | 0 | 0 | 43,60 | **<0,001** | 1,33 | **CN2** | Oui |
| *waveform40* | 221,56 | 7,14 | 240,30 | 7,86 | 3 | 47 | 0 | -12,37 | **<0,001** | 0,92 | **PmBc** | Oui |
| *wine* | 5,22 | 0,76 | 9,18 | 0,92 | 0 | 50 | 0 | -26,20 | **<0,001** | 0,57 | **PmBc** | Oui |

**Figure 3-11** Comparaison PmBc-CN2 (nombre de règles produites)

La complexité des représentations produites par CN2 est statistiquement inférieure à celle de PmBc dans la plupart des cas (dans 10 des 15 tâches d'apprentissage) et ceci en tenant compte uniquement du nombre de règles produites (le nombre de conditions rajoutées par CN2 aux corps des règles est généralement inférieur au nombre maximal de conditions que produit systématiquement PmBc). Comme les performances de classification de CN2 sont aussi inférieures à celles de PmBc, cette observation semble indiquer que la fonction d'évaluation de CN2 réalise un élagage trop important de ses représentations.

#### 3.2.2.6 Temps de calculs

Cette comparaison ne serait pas complète sans l'étude des temps d'exécution des trois algorithmes. Cette expérience a été réalisée sur la base de données *letter*, qui contient le plus grand nombre d'exemples parmi celles utilisées dans les expériences précédentes. La comparaison a été effectuée avec les implémentations des algorithmes proposés par leurs auteurs (langage C pour C4.5 et CN2, C++ pour PmBc et BNGE). La figure 3-12 présente l'évolution des temps d'exécution de BNGE et PmBc rapportés aux temps d'exécution de C4.5, en fonction du nombre d'exemples d'apprentissage.

Il existe des différences très importantes entre les temps d'exécution des quatre algorithmes et le vainqueur incontestable de cette épreuve est C4.5. PmBc se situe, du point de vue de la vitesse, entre C4.5 et CN2, BNGE étant beaucoup plus lent que les autres. Les courbes d'évolution des temps d'exécution de PmBc et de CN2 par rapport à C4.5 présentent une allure quasi-linéaire (avec un certain effet de saturation vers la fin de l'intervalle considéré), tandis que celle correspondant à BNGE fait apparaître une dépendance du nombre d'exemples d'ordre plutôt quadratique, dans le même intervalle.

Il faut observer qu'au cours de toutes les expériences rapportées dans la section précédente (qui devraient constituer un échantillon représentatif de tâches pratiques d'apprentissage), les temps d'exécution de PmBc sont restés dans des limites que nous considérons acceptables pour la plupart des applications classiques (35 minutes de temps CPU utilisateur sur une station hyper Sparc pour 10.000 exemples d'apprentissage de la base *letter*).

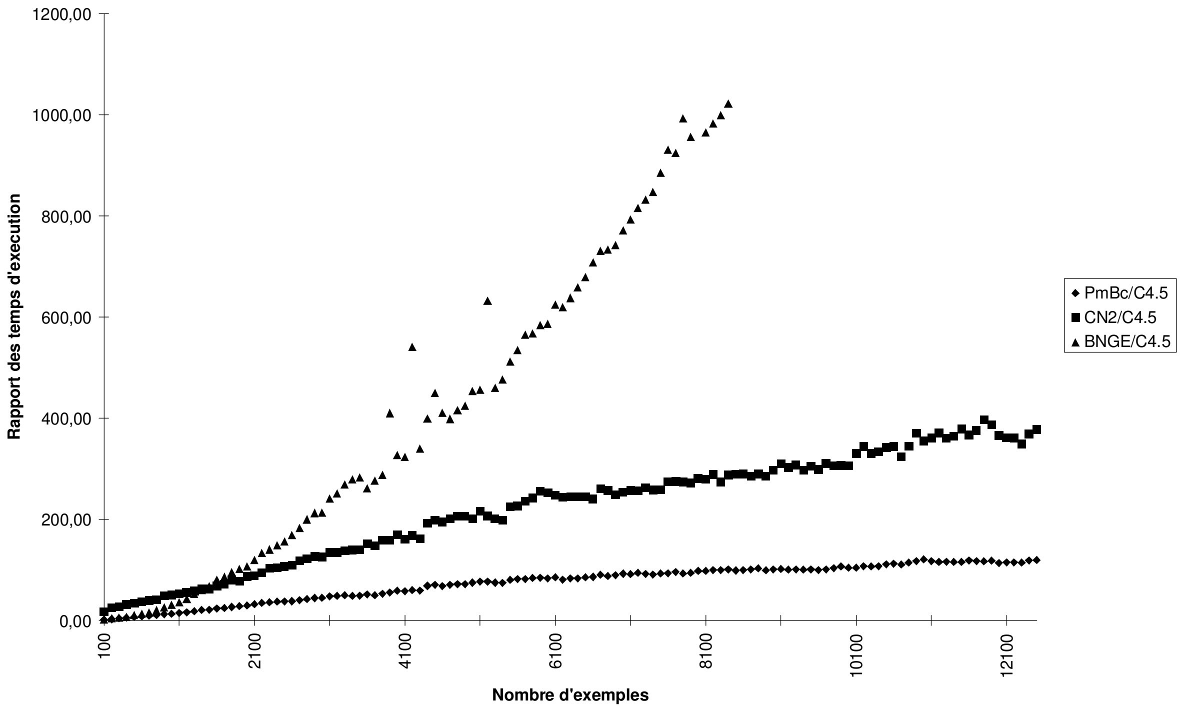



**Figure 3-12** Temps d'exécution de PmBc, BNGE et CN2 rapportés à ceux de C4.5

Cependant, dans certaines tâches d'apprentissage en ligne qui imposent des contraintes strictes sur la vitesse d'exécution, le temps de calcul nécessaire à PmBc pourrait rendre son utilisation prohibitive. Par ailleurs, l'exploitation de PmBc pour la découverte de connaissances dans les bases de données très volumineuses doit obligatoirement passer par des techniques de fenêtrage, comme celle présentée dans [Quinlan, 1993].

### 3.2.3 Similitudes et différences entre PmBc et d'autres approches en apprentissage

Vu les résultats positifs des comparaisons expérimentales entre PmBc et d'autres algorithmes issus de l'apprentissage symbolique, il est important de mettre en évidence les caractéristiques originales de cette approche qui, selon nous, expliquent ses performances, et d'analyser les nouvelles directions de recherche ouvertes par ces travaux. Dans les sections précédentes, nous avons déjà positionné PmBc face à d'autres algorithmes d'induction d'hyperrectangles. Ici, nous allons analyser les caractéristiques qui individualisent notre algorithme par rapport à d'autres algorithmes d'apprentissage plus connus.

Nous avons cité, dans la section 2.3.3, l'argument généralement invoqué par les travaux en induction d'hyperrectangles, dans l'axe desquels nous avons placé la présentation de PmBc :

l'intérêt de l'utilisation conjointe de méthodes et concepts issus de l'apprentissage symbolique (l'induction de règles) et des méthodes de discrimination, issues de la statistique, qui mémorise directement les exemples (comme la méthode des *k* plus proches voisins). Nous allons, dans cette section, analyser les particularités de PmBc par rapport à chacune de ces deux catégories d'approches. La conclusion que nous voulons défendre est que les principes sur lesquels est fondé PmBc peuvent être rapprochés, individuellement, d'autres travaux en apprentissage, mais que l'utilisation conjointe de ces principes pour l'apprentissage symbolique supervisé constitue une caractéristique originale de notre approche.

#### 3.2.3.1 Relation avec d'autres approches symboliques

Avant tout, il faut préciser que, si nous nous rallions (dans la conception de PmBc) aux principes qui sont à la base des approches connues sous le nom d'induction d'hyperrectangles, nous doutons de l'adéquation de cette appellation (proposée dans [Salzberg, 1991] et reprise ensuite dans plusieurs communications sur ce sujet). En effet, les représentations construites par la plupart des algorithmes d'apprentissage symbolique (aussi bien en induction d'arbres de décision , comme C4.5, qu'en induction de règles, comme CN2) utilisent, pour les attributs numériques, des tests d'inégalité dont les conjonctions peuvent être représentées graphiquement comme des hyperrectangles (éventuellement dégénérés).

Selon nous, l'originalité des algorithmes dits "d'induction d'hyperrectangles" réside plutôt dans l'importance accordée à la **similarité** par leurs *stratégies de classification* et, surtout, par leurs *stratégies de recherche*, que dans la forme des représentations construites. Dans les paragraphes suivants, nous allons détailler ces deux axes de comparaison.

***Stratégies de classification***

Les représentations des classes construites par la plupart des arbres de décision (dont C4.5) réalisent une partition complète et disjointe de l'espace des exemples. La classification d'un nouvel exemple revient à la vérification des conditions qui figurent dans les noeuds de l'arbre et à l'attribution de cet exemple à la classe la plus représentée dans la feuille (unique) à laquelle il appartient. Aucune autre stratégie de classification n'est donc nécessaire (aucun conflit ou manque de décision ne peut apparaître), ce qui constitue une différence importante par rapport aux représentations utilisées par PmBc.

A la différence des arbres de décision, les représentations à base de règles, comme celles utilisées dans AQ15 ou CN2, acceptent un certain recouvrement entre les représentations des classes et, également, la nécessité de prévoir des stratégies de classification spécifiques pour les exemples qui ne sont couverts par aucune règle. Dans CN2, par exemple, la classification d'un tel exemple est réalisée en utilisant une règle par défaut qui associe toutes les régions non-couvertes de l'espace des exemples à une même classe (la classe la plus représentée dans l'ensemble d'apprentissage).

Une stratégie plus sophistiquée est adoptée dans AQ15, qui consiste dans la vérification "flexible" des conditions des règles. Cette stratégie accorde aux exemples à classer des degrés d'appartenance (*mesure of fit*) aux différentes classes, de valeur fractionnaire, dans les situations où aucune règle ne peut être directement appliquée, et les affectent à la classe qui maximise ce degré d'appartenance.

Le degré d'appartenance à une classe est calculé dans AQ15 sur des considérations probabilistes, à partir des degrés d'appartenance aux règles de la classe respective qui, à leur tour, dépendent des degrés de respect de leurs conditions. Le degré de respect d'une condition vaut 1 si la condition est respectée ou une valeur fractionnaire (qui correspond à la proportion de valeurs couvertes par la condition respective par rapport au domaine entier de variation de l'attribut testé) dans le cas contraire. Il en résulte donc que le degré d'appartenance à une classe est discontinu.

La stratégie de classification des exemples non-couverts, utilisée par PmBc (et par la plupart des autres algorithmes d'induction d'hyperrectangles), met en oeuvre l'idée tout à fait naturelle de calculer un tel "degré d'appartenance", pour les espaces numériques, à partir de la distance euclidienne (et donc de la similarité géométrique). Nous obtenons ainsi un degré d'appartenance à évolution continue, plus approprié, pour les espaces numériques et aussi en accord avec l'hypothèse de similarité des exemples appartenant à la même classe.

Il faut cependant observer une grande similitude conceptuelle entre ces deux familles d'approches (induction d'hyperrectangles et AQ), en ce qui concerne les représentations et les stratégies de classification utilisées. De ce point de vue, les différences semblent s'estomper, comme le montre récemment une approche issue de la famille AQ (FCLS [Zhang & Michalski, 1995]), qui utilise des degrés d'appartenance fondés sur la distance et qui présente, d'après ses auteurs également, un grand nombre de caractéristiques communes avec l'induction d'hyperrectangles (en l'occurrence, NGE).

***Stratégies de recherche***

L'importance accordée à la similarité par sa stratégie de recherche constitue, selon nous, l'originalité principale de PmBc dans le contexte de l'apprentissage symbolique supervisé. Il s'agit, essentiellement, de l'utilisation de techniques semblables à celles de l'apprentissage non-supervisé (ou de la classification statistique) pour la construction des représentations disjonctives, dans des tâches supervisées d'apprentissage. En effet, les représentations recherchées dans ces tâches doivent simultanément réaliser la discrimination entre les exemples appartenant aux différentes classes (composante supervisée de la recherche) et leur modélisation par différentes règles qui représentent une même classe et qui correspondent généralement à des sous-phénomènes ou des sous-classes étiquetées de la même façon (composante non-supervisée).

Pour illustrer l'apport d'une prise en compte explicite de la similarité dans la stratégie de recherche d'un algorithme d'apprentissage supervisé, nous allons nous appuyer sur l'exemple du traitement des attributs numériques dans C4.5. Nous avons choisi cet exemple parce qu'il a fait l'objet, récemment, de plusieurs communications sur ce sujet et parce que nous allons revenir sur cette problématique dans la section 3.3. Des observations similaires peuvent cependant être formulées pour un grand nombre d'autres algorithmes (comme CN2, AQ15 ou FCLS, par exemple) qui, eux non plus, n'évaluent pas la similarité entre les exemples couverts par la même règle, au cours de la recherche.

C4.5 construit des arbres de décision par des spécialisations successives. L'algorithme examine à chaque pas toutes les spécialisations possibles qui impliquent un test sur un seul attribut et choisit celle qui optimise un critère issu de la théorie de l'information : le rapport du gain informationnel. Ce critère dépend exclusivement de la distribution des classes avant et après le test candidat pour la spécialisation.

Certains auteurs ont rapporté une dégradation des performances de C4.5 dans les domaines où les attributs numériques sont prépondérants. Une première explication possible de ce phénomène est donnée par l'auteur de C4.5 dans [Quinlan, 1996]. Elle passe par l'observation que le nombre d'hypothèses considérées par C4.5 dans le choix d'un test dépend, dans le cas des attributs numériques, du nombre de valeurs distinctes de l'attribut pour les exemples de l'ensemble d'apprentissage. Ceci confère une importance relative plus grande aux attributs numériques avec un grand nombre de valeurs par rapport aux attributs discrets et par rapport aux attributs numériques avec moins de valeurs.

Pour éliminer ce biais, [Quinlan, 1996] propose une pénalisation du score des tests sur des attributs numériques par une valeur qui dépend du nombre d'hypothèses considérées pour l'attribut respectif. Cette correction est prise en compte dans une nouvelle version de C4.5 (*Release 8*) et son bénéfice est validé empiriquement sur plusieurs bases de données de UC-Irvine [Murphy & Aha, 1993] dans [Quinlan, 1996]. C'est cette version de C4.5 qui a été utilisée dans nos expériences.

Une deuxième explication, probablement complémentaire à celle donnée par Quinlan, est proposée dans [Van de Merckt, 1993]. Cet article suggère que cette baisse de performance est due à une prise en compte inappropriée de la similarité des exemples dans les espaces numériques par les fonctions d'évaluation fondées sur l'entropie. La figure 3-13 reprend un exemple présenté dans [Van de Merckt, 1993] pour illustrer la différence entre la partition proposée par les techniques d'induction d'arbres de décision qui utilise une fonction d'évaluation fondée exclusivement sur l'entropie et une partition alternative, fondée sur la similarité des exemples. Pour un nombre réduit d'exemples, les procédures d'élagage de l'arbre de décision enlèveront sûrement le test marqué avec un signe d'interrogation, rendant même plus différentes les deux partitions.

Cette différence de comportement ne peut pas être imputée uniquement au nombre réduit d'exemples disponibles. En effet, une fonction d'évaluation telle que le rapport de gain de C4.5, qui repose uniquement sur des rapports d'effectifs (fréquences relatives d'exemples de chaque classe avant et après l'ajout d'un test, proportions d'exemples sur chacune des branches) sera insensible à une augmentation du nombre d'exemples si leur topologie et leurs distributions relatives restent inchangées. Par contre, si l'augmentation du nombre d'exemples disponibles est accompagnée d'une représentativité de plus en plus complète de l'espace des exemples, les deux partitions vont converger vers des représentations équivalentes (mais, dans une telle situation idéale, la plupart des fonctions d'évaluation utilisées dans l'apprentissage devraient obtenir la même représentation).

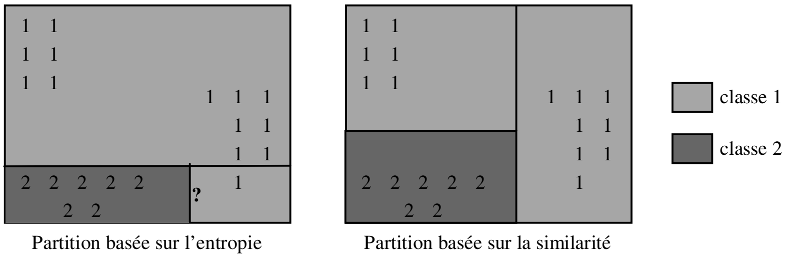



**Figure 3-13** Partitions fondées sur l'entropie et sur la similarité [Van de Merckt, 1993]

En accord avec cette deuxième explication, [Webb, 1996] propose un post-traitement des arbres de décision produits par C4.5, fondé sur l'hypothèse de la similarité, qui apporte une amélioration significative des performances de l'algorithme dans les domaines numériques.

Les résultats de la comparaison expérimentale avec PmBc (fondé explicitement sur l'estimation de la distance entre les exemples) semblent appuyer cette explication. Les représentations utilisés et la stratégie de recherche de PmBc (ascendante, par généralisations successives) diffèrent par beaucoup d'autres aspects de celles de C4.5. En outre, PmBc évalue la similarité globale entre les descriptions des exemples et C4.5 concentre sa recherche sur un seul attribut à la fois. Nous pensons, cependant, que ce type d'étude pourrait générer d'autres pistes d'investigation pour la prise en compte de la similarité dans l'induction d'arbres de décision. Dans la section 3.3 nous allons présenter nos propositions pour la construction d'une fonction d'évaluation qui s'appuie simultanément sur des critères informationnels et de similarité dans la sélection des tests.

A la fin de cette discussion on peut se demander pourquoi nous avons accordé une importance particulière aux attributs numériques. En accord avec [Van de Merckt, 1993], nous pensons que le type des tests utilisés par les arbres de décision pour partitionner l'espace des exemples sur la base des attributs symboliques suffit, en lui seul, pour regrouper ensemble les exemples similaires. En effet, les branches partant d'un noeud qui teste un attribut symbolique correspondent à des conditions de similarité $A_i = v$, tandis que celles correspondant aux attributs numériques (de la forme $A_i \leq v$ et $A_i > v$) peuvent regrouper ensemble des valeurs très différentes.

#### 3.2.3.2 Relation avec les approches statistiques

Ce positionnement de PmBc (et des autres algorithmes d'induction d'hyperrectangles) face à d'autres approches en apprentissage ne serait pas complète sans l'évocation des méthodes statistiques (de classification ou de discrimination non-paramétrique) qui utilisent depuis longtemps des techniques très proches, pour atteindre des objectifs souvent assez différents. Nous allons suivre les mêmes axes de comparaison que dans la section précédente.

***Stratégies de classification***

La discrimination constitue depuis longtemps un sujet privilégié de recherche de la communauté statistique et d'analyse de données. Les algorithmes d'induction d'hyperrectangles reprennent certaines idées issues des méthodes statistiques de discrimination (et surtout de celles qui sont appelées "non-paramétriques", parce qu'elles n'imposent pas des modèles *a priori* pour les distributions des classes).

La plupart des communications sur l'induction d'hyperrectangles citent la méthode des $k$ plus proches voisins comme étant, avec l'induction de règles, une des sources principales d'inspiration de ces travaux. En effet, cette méthode de discrimination non-paramétrique très simple, mais souvent très efficace, attribue à un nouvel exemple $x$ la classe la plus représentée parmi les $k$ exemples d'apprentissage les plus proches de $x$. Elle utilise donc, d'une façon explicite, une mesure de distance pour la prise de décision, ce qui la rapproche de la stratégie de classification de PmBc.

Il existe également d'autres méthodes de discrimination non-paramétriques qui peuvent être rapprochées des algorithmes d'induction d'hyperrectangles. L'idée de base des fenêtres de Parzen est d'approximer la densité de probabilités d'une classe $C_i$ par la sommation de fonctions élémentaires (appelées "noyaux"), d'intégrale unité, correspondant à chacun des exemples d'apprentissage de la classe $C_i$, rapportée au nombre total d'exemples de cette classe. Dans le cas le plus simple, il s'agit de noyaux hypercubiques de dimension $h$ (ces fonctions sont constantes à l'intérieur d'un hypercube de coté $h$ et nulles à l'extérieur), centrés autour de chaque exemple d'apprentissage.

Dans certaines conditions [Hand, 1981], on peut démontrer que cette approximation converge vers la densité réelle $f(x|C_i)$ quand $h$ diminue et le nombre d'exemples disponibles de la classe $C_i$ augmente. Un nouvel exemple $x$ sera attribué à la classe qui maximise l'estimation de $f(x|C_i)$, en accord avec le principe de "maximum de vraisemblance". Nous avons mentionné les noyaux de forme hypercubique pour mieux mettre en évidence la ressemblance avec la forme des représentations obtenues par l'induction d'hyperrectangles, mais d'autres classes de noyaux (qui suivent une loi gaussienne, par exemple), peuvent être également utilisées pour obtenir un meilleur lissage de l'approximation.

Par l'absence d'hypothèses contraignantes sur les distributions des exemples, l'objet des méthodes statistiques de discrimination non-paramétriques peut être rapproché de celui des méthodes symboliques d'apprentissage, en général. Par contre, il existe une différence très importante entre les démarches adoptées par ces deux catégories de méthodes pour atteindre leurs objectifs. Les méthodes symboliques d'apprentissage (parmi lesquelles PmBc) visent à construire, dans une première phase, des **représentations** compactes des classes, qui seront ensuite utilisées pour la classification. Ces représentations, qui résument l'information utile pour la classification apportée par les exemples, sont également destinées à la compréhension et à l'utilisation directe par les humaines. Leurs complexité doit dépendre, dans le cas idéal, de la complexité de la tâche de classification analysée et non pas du nombre d'exemples disponibles.

Les méthodes statistiques de discrimination non-paramétriques, telles que les fenêtres de Parzen ou les *k* plus proches voisins, réduisent la phase d'apprentissage au simple stockage en mémoire des exemples disponibles et délèguent l'intégralité du traitement à la phase de classification. Comme elles ne produisent pas des représentations explicites, elles ne peuvent pas être utilisées directement pour l'extraction de connaissances dans les bases de données parole, qui constitue l'objectif principal de nos recherches. Par contre, les stratégies de classification qu'elles utilisent, fondées sur des bases théoriques rigoureuses, sont très proches et pourraient donc fournir des pistes de recherche intéressantes à des méthodes d'apprentissage symbolique, comme les algorithmes d'induction d'hyperrectangles.

***Stratégies de recherche***

Nous avons déjà évoqué, dans la section précédente, la ressemblance entre la stratégie de recherche de PmBc et celles utilisées dans l'apprentissage non-supervisé. Les méthodes statistiques de classification, comme les nuées dynamiques [Diday *et al.*, 1983], cherchent des partitions "naturelles" des exemples (non-étiquetés) bien agrégées et bien séparées entre elles. Leurs stratégies de recherche reposent donc essentiellement sur la similarité entre les exemples, comme celle de PmBc, mais ce dernier doit également gérer la composante supervisée du processus d'apprentissage.

### 3.2.4 Conclusion

Dans cette section, nous avons présenté la conception et les performances d'un nouvel algorithme d'induction d'hyperrectangles, PmBc. Il combine des idées issues de l'induction de règles, de l'apprentissage non-supervisé et des méthodes non-paramétriques de discrimination statistique dans une méthode d'apprentissage simple à comprendre et à utiliser, dépourvue de tout paramètre à réglage laborieux. Les résultats expérimentaux obtenus montrent également un avantage certain de notre approche du point de vue de la précision des classificateurs produits. Par contre, en comparaison avec C4.5, les temps d'exécution sont nettement plus importants.

PmBc a été conçu pour assister les chercheurs en communication parlée au cours de l'extraction de connaissances dans les bases de données parole. Certains aspects de la problématique généralement abordée par les études en apprentissage automatique, mais qui sont moins souvent rencontrés dans ce domaine, ont été négligés dans la version actuelle de l'algorithme. En effet, pour élargir le champ d'applicabilité de PmBc, il est indispensable de proposer dans les versions ultérieures des solutions aux problèmes liés au traitement des attributs symboliques et des valeurs manquantes. La plupart des solutions classiques sont déjà directement applicables, mais d'autres possibilités pourraient s'avérer plus performantes dans ce nouveau contexte. Par ailleurs, l'heuristique de recherche de PmBc pourrait être adaptée pour l'apprentissage incrémental et la révision de théorie.

En outre, une piste d'investigation qui pourrait s'avérer intéressante dans l'avenir consisterait en l'utilisation de la procédure de généralisation de PmBc en tant que post-traitement des règles produites par C4.5*rules* (algorithme qui transforme les arbres de décision en règles), pour l'affinement des conditions sur les attributs numériques.

## 3.3 IDF : induction d'arbres de décision par l'approximation des fréquences

Cette section propose une réflexion d'ordre plutôt théorique sur une interprétation probabiliste de l'induction des arbres de décision. Par rapport à d'autres approches comme PmBc (section 3.2), qui peuvent s'avérer plus performantes du point de vue de la précision de classification, les arbres de décision présentent trois atouts importants :

- la vitesse d'exécution
- la représentation "logique" (caractérisation complète et disjointe des classes)
- l'approche descendante, par spécialisations successives, qui permet la prise en compte des caractéristiques de nature statistique d'un nombre important de données.

Ces recherches se proposent de modéliser la construction des arbres de décision sous la forme d'un processus itératif de décision bayesienne. Notre objectif est d'essayer de justifier d'une façon plus rigoureuse certaines heuristiques utilisées dans ce domaine et de mettre en évidence des nouvelles directions d'exploration pour la recherche de meilleures fonctions d'évaluation.

### 3.3.1 Interprétation bayesienne des choix des tests

La démarche qui est à la base de ces recherches nous a été suggérée par l'ouvrage du physicien américain E.T. Jaynes qui propose une théorie originale du raisonnement probabiliste, appelée "*Probability as Logic*" [Jaynes, 1994]. Cette théorie présente les probabilités comme une extension de la logique booléenne et formalise, dans ce cadre, les notions d'inférence, d'incertitude et d'état de connaissances. Il nous semble important de mentionner que, dans l'esprit de la théorie de Jaynes, toutes les notations de probabilités qui seront utilisées dans les sections qui suivent doivent être interprétées comme étant conditionnelles à notre état de connaissances initiales sur le phénomène étudié. Pour alléger les notations et les rendre plus proches de celles généralement utilisées, nous avons préféré omettre la spécification formelle de cet aspect chaque fois qu'il apparaît, sans négliger pour autant son importance.

Nous allons nous focaliser sur le choix du test qui figurera dans le noeud qui sera rajouté sur une des branches de l'arbre de décision. Considérons, pour l'instant, seulement les tests binaires qui imposent une condition sur les valeurs d'un seul attribut (le choix d'un test d'inégalité sur des attributs numériques dans C4.5, par exemple). Pour faciliter les notations et sans restreindre la généralité, supposons que les exemples correspondant à la branche analysée soient divisés en deux classes : $p$ exemples "positifs" et $n$ exemples "négatifs".

Pour chaque test $j$ sur l'attribut $i$ nous formulons l'hypothèse suivante :

$\boldsymbol{H_{ij}}$ : Les exemples proviennent de deux processus différents (correspondant aux deux branches du test) caractérisés par des fréquences d'apparition d'exemples positifs différentes : $f_1$ et $f_2$ (les indices $ij$ seront omis, pour simplifier les notations). La fréquence d'apparition des exemples correspondant au premier processus est *d* (les exemples générés par le deuxième processus apparaissent donc avec une fréquence *1-d*).

Les meilleures estimations des fréquences $f_1$, $f_2$ et $d$ à partir des données disponibles sont donc : $f_1 = \frac{p_1}{p_1 + n_1}$, $f_2 = \frac{p_2}{p_2 + n_2}$ et $d = \frac{p_1 + n_1}{p + n}$ où $p_1$, $n_1$ et, respectivement $p_2$, $n_2$ correspondent aux nombres d'exemples positifs et négatifs sur les deux branches.

Les hypothèses $H_{ij}$ essaient donc de mettre en évidence des points de discontinuité marquée dans la fréquence d'apparition d'exemples positifs qui pourront être utilisés par l'algorithme de construction de l'arbre de décision pour partitionner l'espace des exemples.

Nous pouvons également formuler une hypothèse supplémentaire :

$\boldsymbol{H_0}$ : Tous les exemples proviennent d'un même processus, qui fournit des exemples positifs avec une fréquence *f*.

La meilleure estimation de la fréquence *f* à partir des données est : $f = \frac{p}{p + n}$.

Cette hypothèse nous permet de prendre en compte la possibilité d'arrêter la ramification de l'arbre de décision dans les situations où une partition complète des exemples n'est pas réalisable pour une des raisons suivantes :

- présence du bruit
- absence de certains traits discriminants dans la description des exemples.

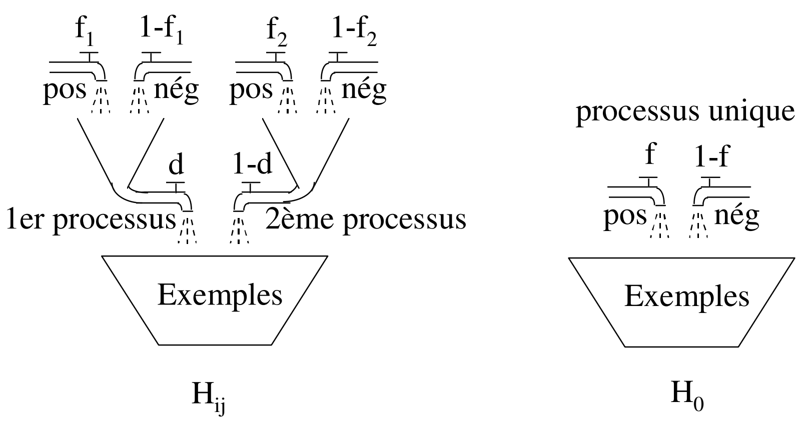



**Figure 3-14** Représentation figurative des hypothèses considérées

Le théorème de Bayes nous permet de calculer la probabilité *a posteriori* de chacune de ces hypothèses :

$P(H|D) = P(H) \times \dfrac{P(D|H)}{P(D)}$ où $D$ dénote l'évidence apportée par les exemples, $P(H)$ représente la probabilité *a priori* de l'hypothèse $H$, $P(D|H)$ symbolise la probabilité des données dans le contexte de l'hypothèse $H$ et de nos connaissances initiales et $P(D)$ constitue une constante de normalisation correspondant à la probabilité globale des données dans le contexte de nos connaissances initiales.

Pour contourner le calcul de $P(D)$, nous allons estimer les rapports de probabilités *a posteriori* : $O(H_{ij}|D) = \dfrac{P(H_{ij}|D)}{P(H_0|D)} = \dfrac{P(D|H_{ij})}{P(D|H_0)} \times \dfrac{P(H_{ij})}{P(H_0)}$. L'hypothèse la plus probable sera donc la $H_{IJ}$ qui maximise ces rapports si $O(H_{IJ}|D) > 1$, ou la $H_0$ dans le cas contraire. Nous pouvons également écrire ce critère sous une forme logarithmique :

$$E(H_{ij}|D) = log_2\left(O(H_{ij}|D)\right) = log_2\left(\frac{P(D|H_{ij})}{P(D|H_0)}\right) + log_2\left(\frac{P(H_{ij})}{P(H_0)}\right) = E(D|H_{ij}) + E(H_{ij})$$

### 3.3.2 Calcul des probabilités des données

Le premier facteur à estimer pour calculer $O(H_{ij}|D)$ est le rapport entre les probabilités des données dans le contexte des hypothèses $H_{ij}$ et $H_0$. Par "les données" nous désignons toutes les affirmations qui caractérisent les exemples d'apprentissage de la branche examinée. On peut observer que la confrontation entre les hypothèses $H_{ij}$ et $H_0$ peut se baser uniquement sur les distributions d'exemples positifs et négatifs correspondant aux deux branches du test proposé par $H_{ij}$, que nous allons noter $D'$. Toutes les autres caractéristiques des exemples (les valeurs précises des attributs, les distributions des classes par rapport à d'autres tests) sont équiprobables dans les deux cas et leur probabilité sera dénotée par $P_{autres}$. On a donc : $\dfrac{P(D|H_{ij})}{P(D|H_0)} = \dfrac{P(D'|H_{ij}) \times P_{autres}}{P(D'|H_0) \times P_{autres}} = \dfrac{P(D'|H_{ij})}{P(D'|H_0)}$.

La proposition $D'$ peut être formulée, d'une façon plus rigoureuse, comme une conjonction de trois propositions élémentaires :

"$D'_d$ : Parmi les $(p+n)$ exemples disponibles, il existe $(p_1+n_1)$ exemples sur la première branche et le reste de $(p_2+n_2)$ exemples sur la deuxième branche **ET**

$D'_{f_1}$ : Parmi les $(p_1+n_1)$ exemples de la première branche, il existe $p_1$ exemples positifs et le reste de $n_1$ exemples négatifs **ET**

$D'_{f_2}$ : Parmi les $(p_2+n_2)$ exemples de la première branche, il existe $p_2$ exemples positifs et le reste de $n_2$ exemples négatifs."

On peut observer que les propositions $D'_{f_1}$ et $D'_{f_2}$ sont indépendantes entre elles mais dépendent de la proposition $D'_d$. (Par ailleurs, la proposition $D'_d$ dépend elle aussi de la disponibilité d'un nombre total de $(p+n)$ exemples, mais nous pouvons considérer que cette condition - la même pour toutes les hypothèses évaluées - fait partie de nos connaissances initiales.) On peut donc écrire :

$$P\left(D' \middle| H_{ij}\right) = P\left(D'_d \middle| H_{ij}\right) \times P\left(D'_{f_1} \middle| D'_d\ H_{ij}\right) \times P\left(D'_{f_2} \middle| D'_d\ H_{ij}\right)$$

Compte tenu de la formulation de l'hypothèse $H_{ij}$, les trois probabilités qui apparaissent dans cette expression peuvent être exprimées par des lois binomiales qui utilisent, respectivement, les trois fréquences postulées : $d$, $f_1$ et $f_2$. L'expression finale devient donc :

$$P\left(D' \middle| H_{ij}\right) = C_{p+n}^{p_1+n_1} \times \left(\frac{p_1+n_1}{p+n}\right)^{p_1+n_1} \times \left(\frac{p_2+n_2}{p+n}\right)^{p_2+n_2} \times$$

$$\times C_{p_1+n_1}^{p_1} \times \left(\frac{p_1}{p_1+n_1}\right)^{p_1} \times \left(\frac{n_1}{p_1+n_1}\right)^{n_1} \times$$

$$\times C_{p_2+n_2}^{p_2} \times \left(\frac{p_2}{p_2+n_2}\right)^{p_2} \times \left(\frac{n_2}{p_2+n_2}\right)^{n_2}$$

D'après l'hypothèse $H_0$, un processus unique génère $p$ exemples positifs et $n$ exemples négatifs qui sont ensuite arbitrairement séparés dans deux groupes contenant $(p_1+n_1)$ et $(p_2+n_2)$ exemples par le test proposé dans l'hypothèse $H_{ij}$. La probabilité de la première partie de cette affirmation peut être calculée par une loi binomiale utilisant la fréquence $f$ d'apparition d'exemples positifs, tandis que la deuxième partie correspond à un tirage aléatoire d'exemples dans une urne contenant *p* exemples positifs et *n* exemples négatifs. La probabilité des données dans cette hypothèse devient donc :

$$P(D'|H_0) = C_{p+n}^{p} \times \left(\frac{p}{p+n}\right)^{p} \times \left(\frac{n}{p+n}\right)^{n} \times \frac{C_p^{p_1} \times C_n^{n_1}}{C_{p+n}^{p_1+n_1}}$$

$$= C_{p_1+n_1}^{p_1} \times C_{p_2+n_2}^{p_2} \times \left(\frac{p}{p+n}\right)^{p} \times \left(\frac{n}{p+n}\right)^{n}$$

Après la simplification des facteurs communs le logarithme du rapport des probabilités des données devient :

$$P(D'|H_0) = log_2\left(C_{p+n}^{p_1+n_1}\right) - p \times log_2\left(\frac{p}{p+n}\right) - n \times log_2\left(\frac{n}{p+n}\right) +$$

$$+\sum_{i=1}^{2}\left(p_i \times log_2\left(\frac{p_i}{p_i+n_i}\right) + n_i \times log_2\left(\frac{n_i}{p_i+n_i}\right) + (p_i+n_i) \times log_2\left(\frac{p_i+n_i}{p+n}\right)\right) =$$

$$= log_2\left(C_{p+n}^{p_1+n_1}\right) + (p+n) \times \left(I(p,n) - I(p_1,n_1,p_2,n_2) - I(p_1+n_1, p_2+n_2)\right)$$

où on retrouve les deux facteurs : le gain informationnel $I(p,n)$-$I(p_1, n_1, p_2, n_2)$ et l'information associée à la partition $I(p_1 + n_1, p_2 + n_2)$ qui figurent mais sont combinés différemment dans le "rapport de gain" de C4.5 [Quinlan, 1993]. Le quatrième facteur, $log\left(C_{p+n}^{p_1+n_1}\right)$ représente le logarithme d'un coefficient binomial et peut être estimé efficacement sur ordinateur en utilisant la fonction **`lgamma`** (logarithme de la fonction *gamma*), généralement disponible dans les bibliothèques mathématiques.

Enfin, cette expression peut être formulée d'une façon plus compacte en développant les logarithmes et en effectuant les réductions de termes rendues possibles par cette opération :

$$E\left(D|H_{ij}\right) = \log\left(C_{p+n}^{p_1+n_1}\right) + p_1 \times \log(p_1) + n_1 \times \log(n_1) + p_2 \times \log(p_2) +$$

$$+ n_2 \times log(n_2) - p \times log(p) - n \times log(n)$$

### 3.3.3 Allocation des probabilités *a priori*

Avant de passer à l'estimation des probabilités *a priori* il faut observer que le rapport des probabilités des données calculé dans la section précédente peut constituer en soi un critère d'appréciation de la qualité des hypothèses connu sous le nom de "maximum de vraisemblance" (*maximum likelihood*).

L'approche bayesienne [Jaynes, 1994], à laquelle nous nous rallions, demande cependant que nos connaissances initiales (ou notre manque de connaissances) sur les hypothèses testées participent aussi, sous la forme des probabilités *a priori*, au processus de décision. Il est évident que l'ampleur et la nature des connaissances initiales disponibles peuvent varier beaucoup entre les différentes applications pratiques. Leur codage sous forme de probabilités *a priori* pourrait donc constituer une possibilité d'injecter des connaissances du domaine dans les processus d'induction.

Comme notre but est de proposer une stratégie indépendante de l'application, nous allons nous limiter ici à la formalisation des *a priori* généralement assumés dans l'induction des arbres de décision et qui reflètent plutôt notre manque de connaissances sur le phénomène étudié.

Nous proposons une méthode d'allocation des probabilités *a priori* qui tient compte des deux desiderata suivants :

- Les différents tests sur un même attribut $i$ sont *a priori* équiprobables :

$$P(H_{ij}) = P(H_{ik}) \ \forall j,k.$$

- La probabilité *a priori* globale d'accepter un test sur ses valeurs est la même pour les différents attributs :

$$\sum_j P(H_{ij}) = \sum_k P(H_{lk}) \ \forall i,l.$$

Si on accorde à $H_0$ la même probabilité *a priori* que la probabilité globale des tests sur un attribut quelconque :

$$P(H_0) = \sum_j P(H_{ij}) \ \forall i$$

on obtient l'allocation des probabilités *a priori* suivante :

$$\frac{P(H_{ij})}{P(H_0)} = \frac{1}{t_i} \quad \forall j$$ où $t_i$ désigne le nombre de tests considérés pour l'attribut *i*.

On peut maintenant combiner les différents résultats obtenus précédemment dans une seule formule :

$$E(H_{ij}|D) = log\left(C_{p+n}^{p_1+n_1}\right) + (p+n) \times \left( I(p,n) - I(p_1, n_1, p_2, n_2) - I(p_1 + n_1, p_2 + n_2) - \frac{log_2(t_i)}{p+n} \right)$$

On peut observer que le terme $-\frac{log_2(t_i)}{p+n}$ correspond à la pénalisation du score des tests sur les attributs numériques ajoutée récemment à C4.5 [Quinlan, 1996]. Dans notre approche, cette pénalisation est généralisée naturellement à l'évaluation des tests sur les attributs symboliques.

### 3.3.4 Sensibilité aux effectifs

Cette section se propose de mettre en évidence quelques propriétés de la fonction d'évaluation construite dans les sections précédentes, notamment en ce qui concerne sa sensibilité aux effectifs. Nous renvoyons le lecteur à la section 2.2.4 pour un aperçu global de cette problématique.

Dans cette discussion nous allons nous appuyer sur les deux exemples proposés dans [Zighed & Rakotomalala, 1996] pour illustrer les propriétés d'une fonction d'évaluation sensible aux effectifs. Sur ces deux exemples, des fonctions d'évaluation très étudiées, parmi lesquelles celles de CART [Breiman *et al.*, 1984] et C4.5 [Quinlan, 1993], accordent le même score à des situations intuitivement très différentes du point de vue de leur impact sur l'apprentissage.

Le premier exemple est présenté dans la figure 3-15 (l'exemple original à gauche et notre généralisation à droite). Il s'agit de deux situations, dans lesquelles le test proposé par l'hypothèse $H_{ij}$ réalise une discrimination parfaite des exemples d'apprentissage. Les deux situations diffèrent uniquement par le nombre d'exemples disponibles : les effectifs correspondant à la première situation sont dix fois plus nombreux que ceux de la deuxième situation. Intuitivement, la fonction d'évaluation devrait accorder un score plus élevé à la première situation, où la décision de rajouter le test est soutenue par un nombre plus important d'exemples.

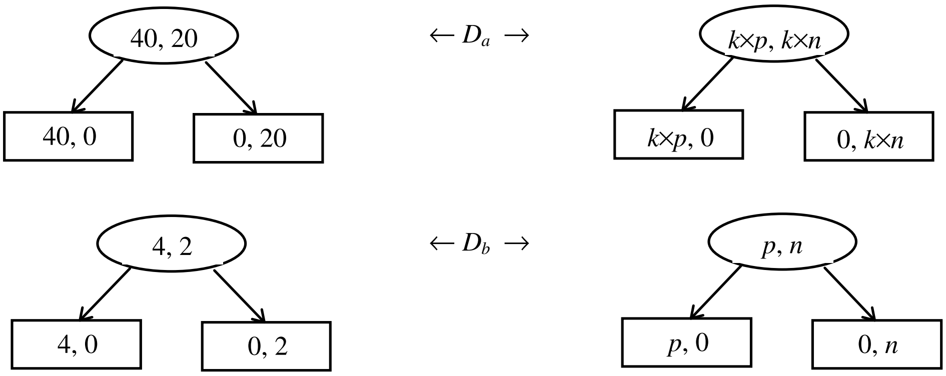



**Figure 3-15** Premier exemple et sa généralisation

Pour la généralisation de cet exemple, présentée dans la partie droite de la figure 3-15, on peut facilement démontrer que :

$$E\left(H_{ij}\middle|D_a\right) - E\left(H_{ij}\middle|D_b\right) = \log\left(\frac{C_{k\times(p+n)}^{k\times p}}{C_{p+n}^{p}}\right) > 0 \;\Rightarrow\; E\left(H_{ij}\middle|D_a\right) > E\left(H_{ij}\middle|D_b\right)\ \forall p, n > 0 \text{ et } k > 1$$

La première situation sera donc toujours préférée par notre fonction d'évaluation. En effet, pour l'exemple numérique proposé, $E(H_{ij}|D_a) - E(H_{ij}|D_b) = 51{,}8965 - 3{,}90689 = 47{,}98961 > 0$.

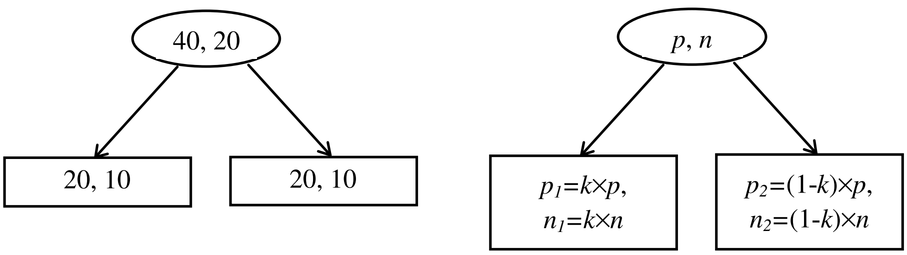



**Figure 3-16** Deuxième exemple et sa généralisation

Le deuxième exemple est présenté dans la figure 3-16. Ici, il s'agit d'étudier dans quelle mesure une fonction d'évaluation est capable de rejeter les tests qui n'apportent aucune information utile pour la classification mais réduisent les effectifs disponibles pour l'estimation des fréquences sur chacune des branches. Dans le cas général (partie droite de la figure 3-16) :

$$E\left(H_{ij}|D\right) = log\left(C_{p+n}^{p_1+n_1}\right) - log\left(\frac{(p+n)^{p+n}}{(p_1+n_1)^{p_1+n_1} \times (p_2+n_2)^{p_2+n_2}}\right) + E\left(H_{ij}\right)$$

Si $0<k<1$ et $(p+n)>0$, alors $0<(p_1+n_1)<(p+n)$ et, donc, en se basant sur un résultat mathématique bien connu :

$$C_{p+n}^{p_1+n_1} = \frac{(p+n)!}{(p_1+n_1)! \times (p_2+n_2)!} < \frac{(p+n)^{p+n}}{(p_1+n_1)^{p_1+n_1} \times (p_2+n_2)^{p_2+n_2}}.$$

Notre méthode d'allocation des probabilités *a priori* nous assure également que $E(H_{ij}) \leq 0$. Il en résulte donc que $E(H_{ij}|D) < 0$ et que notre fonction d'évaluation va rejeter systématiquement le test proposé par l'hypothèse $H_{ij}$. Pour l'exemple numérique proposé, on obtient $E(H_{ij}|D)=$ 56,7148 - 60 $+E(H_{ij})$ = -3,2852 $+E(H_{ij})$.

Les deux exemples analysés, issus des recherches sur les fonctions d'évaluation sensibles aux effectifs, suggèrent que la formule que nous proposons réalise un bon compromis entre la pureté des partitions et le nombre d'exemples disponibles et rejoint donc les objectifs visés par ces recherches. Dans l'avenir, il sera intéressant d'étudier également la possibilité d'intégrer dans notre fonction d'évaluation d'autres estimateurs de fréquence, plus robustes, proposés par ces recherches (comme celui de Laplace, présenté dans la section 2.2.4)

### 3.3.5 Présentation de IDF

Pour évaluer d'une façon expérimentale cette analyse, nous avons conçu un algorithme d'induction d'arbres de décision (IDF) qui utilise la fonction d'évaluation $E(H_{ij}|DI)$ construite dans les sections précédentes pour le choix des tests au cours de la spécialisation. Les tests considérés sont binaires :

- $A_i < S_j$ pour les attributs numériques (par une discrétisation dynamique)
- $A_i = V_j$ pour les attributs symboliques

où $S_j$ représente un seuil automatiquement choisi par l'algorithme et $V_j$ est une des valeurs discrètes de l'attribut $A_i$. Le nombre d'hypothèses testées ($t_i$) correspond généralement au nombre de valeurs possibles pour les attributs symboliques et au nombre de seuils évalués pour les attributs numériques. Dans le cas particulier des attributs binaires, $t_i$ prend la valeur 1, les deux valeurs possibles de l'attribut générant des hypothèses équivalentes.

L'algorithme ne comporte **ni étape distincte ni fonction d'évaluation différente** pour l'élagage : la spécialisation de l'arbre est arrêtée au moment où aucune des hypothèses $H_{ij}$ n'est plus probable que l'hypothèse $H_0$.

Nous allons donc confronter IDF, qui repose entièrement sur une seule fonction d'évaluation, à C4.5, dont l'heuristique de recherche comprend :

- le calcul du gain informationnel, en utilisant des fonctions différentes pour les attributs numériques et symboliques;
- l'élimination des tests qui proposent un nombre de branches ayant plus de deux exemples inférieur à deux;
- la sélection du test qui maximise le rapport de gain parmi les tests avec un gain informationnel supérieur à la moyenne (ce rapport de gain qui divise deux mesures d'information entre elles est difficilement interprétable du point de vue théorique);
- l'élagage de l'arbre de décision sur la base du pouvoir prédictif estimé de ses sous-arbres (en utilisant un niveau de confiance choisi empiriquement).

| Base | IDF Moy | ± | C4.5 Moy | ± | > | < | = | t | p | Rapport | Meilleur | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 5,61 | 1,65 | 5,59 | 1,82 | 24 | 23 | 3 | 0,12 | 0,909 | 1 | C45 | |
| *bupa* | 38,29 | 5,72 | 35,81 | 5,95 | 35 | 15 | 0 | 3,22 | 0,002 | 1,07 | **C45** | Oui |
| *diabetes* | 26,31 | 3,51 | 25,65 | 3,93 | 30 | 20 | 0 | 1,41 | 0,166 | 1,03 | C45 | |
| *glass* | 33,86 | 7,52 | 34,15 | 6,25 | 29 | 21 | 0 | -0,24 | 0,801 | 0,99 | IDF | |
| *ionosphere* | 10,23 | 3,51 | 10,83 | 3,62 | 19 | 31 | 0 | -1,32 | 0,191 | 0,94 | IDF | |
| *iris* | 4,13 | 4,5 | 4,46 | 4,14 | 23 | 13 | 14 | -1,12 | 0,263 | 0,93 | IDF | |
| *letter* | 11,83 | 0,66 | 12,52 | 0,54 | 8 | 42 | 0 | -7,9 | **<0,001** | 0,95 | **IDF** | Oui |
| *new-thyroid* | 5,21 | 3,56 | 7,55 | 4,14 | 10 | 38 | 2 | -3,88 | **<0,001** | 0,69 | **IDF** | Oui |
| *segment* | 3,45 | 0,94 | 3,4 | 0,83 | 27 | 23 | 0 | 0,42 | 0,68 | 1,01 | C45 | |
| *sonar* | 26,1 | 8,63 | 25,62 | 8,79 | 30 | 20 | 0 | 0,47 | 0,638 | 1,02 | C45 | |
| *vehicle* | 28,98 | 3,39 | 27,02 | 3,22 | 36 | 14 | 0 | 3,75 | **<0,001** | 1,07 | **C45** | Oui |
| *vowel* | 21,02 | 2,56 | 22,3 | 3,23 | 20 | 30 | 0 | -3,09 | 0,003 | 0,94 | **IDF** | Oui |
| *waveform* | 23,41 | 1,47 | 23,68 | 1,16 | 24 | 26 | 0 | -1,31 | 0,192 | 0,99 | IDF | |
| *waveform40* | 23,78 | 1,22 | 24,58 | 1,34 | 17 | 33 | 0 | -3,94 | **<0,001** | 0,97 | **IDF** | Oui |
| *wine* | 7,44 | 4,74 | 7,67 | 4,43 | 25 | 24 | 1 | -0,46 | 0,64 | 0,97 | IDF | |
| *SCORE* | | | | | | | | | | 0,97 | 9-6 | 4-2 |
| *mushroom* | 0 | 0 | 0 | 0 | 0 | 0 | 50 | --- | 1 | 1 | *Egalité* | |
| *crx* | 18,97 | 6,62 | 19 | 7,4 | 31 | 19 | 0 | -0,07 | 0,936 | 1,00 | IDF | |
| *german* | 29,6 | 3,17 | 28,71 | 3,06 | 31 | 19 | 0 | 1,79 | 0,079 | 1,03 | C45 | |
| *kr-vs-kp* | 0,48 | 0,3 | 0,71 | 0,36 | 8 | 42 | 0 | -5,73 | **<0,001** | 0,68 | **IDF** | Oui |
| *krkopt* | 24,88 | 0,63 | 44,45 | 0,69 | 0 | 50 | 0 | -175,04 | **<0,001** | 0,56 | **IDF** | Oui |
| *postoperative* | 28,78 | 7,25 | 28,66 | 7,21 | 25 | 25 | 0 | 1,02 | 0,309 | 1,00 | C45 | |
| *promoters* | 25,88 | 10,01 | 25,8 | 10,15 | 24 | 26 | 0 | 0,06 | 0,956 | 1,00 | C45 | |
| *splice* | 6,38 | 1,05 | 6,07 | 0,84 | 31 | 19 | 0 | 1,93 | 0,059 | 1,05 | C45 | |
| *tic-tac-toe* | 26,64 | 16,03 | 29,18 | 12,99 | 18 | 32 | 0 | -3,71 | **<0,001** | 0,91 | **IDF** | Oui |
| *zoo* | 6,1 | 7,27 | 8,1 | 6 | 12 | 32 | 6 | -4,85 | **<0,001** | 0,75 | **IDF** | Oui |
| *SCORE* | | | | | | | | | | 0,9 | 5-4 | 4-0 |

**Figure 3-17** Comparaison taux d'erreur IDF-C4.5

Les performances de classification et les tailles des arbres de décision construits ont été évaluées sur les bases de données numériques de l'UCI utilisées dans les expériences précédentes auxquelles ont été ajoutées 10 nouvelles bases de données, fournies par le même serveur, comportant aussi des attributs symboliques. Le protocole expérimental et la présentation des résultats sont identiques à ceux utilisés dans la section 3.2.2.

L'analyse des résultats expérimentaux présentés dans la figure 3-17 met en évidence des comportements assez proches des deux algorithmes, avec un léger avantage du côté de IDF aussi bien en considérant le nombre total de meilleures classifications ou le nombre de meilleures classifications validées statistiquement qu'en se basant sur la moyenne du rapport des taux d'erreur des deux algorithmes.

| Base | IDF Moy | ± | C4.5 Moy | ± | > | < | = | t | p | Rapport | Moins complexe | Statis$^{ent}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 25,52 | 5,3 | 21,36 | 4,99 | 35 | 11 | 4 | 4,41 | **<0,001** | 1,19 | **C45** | Oui |
| *bupa* | 8,48 | 3,81 | 42,8 | 10,18 | 0 | 50 | 0 | -22,65 | **<0,001** | 0,2 | **IDF** | Oui |
| *diabetes* | 15,72 | 3,63 | 38,84 | 15,4 | 3 | 46 | 1 | -11,1 | **<0,001** | 0,4 | **IDF** | Oui |
| *glass* | 44,48 | 8,43 | 42,84 | 4,16 | 28 | 18 | 4 | 1,59 | 0,118 | 1,04 | C45 | |
| *ionosphere* | 19 | 5,21 | 22,76 | 4,36 | 14 | 31 | 5 | -4,21 | **<0,001** | 0,83 | **IDF** | Oui |
| *iris* | 8,28 | 1,75 | 7,84 | 1,22 | 12 | 4 | 34 | 2,29 | 0,026 | 1,06 | **C45** | Oui |
| *letter* | 3454,8 | 43,88 | 2165,9 | 35,09 | 50 | 0 | 0 | 155,84 | **<0,001** | 1,6 | **C45** | Oui |
| *new-thyroid* | 15,76 | 2,91 | 13,4 | 2,65 | 34 | 8 | 8 | 4,98 | **<0,001** | 1,18 | **C45** | Oui |
| *segment* | 88,12 | 6,87 | 77,2 | 6,8 | 41 | 6 | 3 | 8,55 | **<0,001** | 1,14 | **C45** | Oui |
| *sonar* | 26,08 | 3,6 | 25,12 | 2,87 | 24 | 13 | 13 | 1,54 | 0,13 | 1,04 | C45 | |
| *vehicle* | 96,64 | 7,52 | 116,8 | 17,69 | 6 | 42 | 2 | -7,15 | **<0,001** | 0,83 | **IDF** | Oui |
| *vowel* | 216,16 | 9,04 | 177,16 | 8,74 | 50 | 0 | 0 | 20,71 | **<0,001** | 1,22 | **C45** | Oui |
| *waveform* | 281,2 | 18,29 | 485,36 | 29,07 | 0 | 50 | 0 | -39,67 | **<0,001** | 0,58 | **IDF** | Oui |
| *waveform40* | 320,48 | 22,62 | 534,32 | 20,31 | 0 | 50 | 0 | -47,59 | **<0,001** | 0,6 | **IDF** | Oui |
| *wine* | 11,76 | 1,85 | 9,92 | 2,03 | 34 | 4 | 12 | 5,71 | **<0,001** | 1,19 | **C45** | Oui |
| *SCORE* | | | | | | | | | | 0,94 | 6-9 | 6-7 |
| *mushroom* | 17 | 0 | 31,1 | 2,09 | 0 | 50 | 0 | -47,64 | **<0,001** | 0,55 | **IDF** | Oui |
| *crx* | 9,2 | 5,87 | 13,48 | 14,4 | 16 | 17 | 17 | -2,58 | 0,013 | 0,68 | **IDF** | Oui |
| *german* | 47,76 | 17 | 104,04 | 22,01 | 0 | 50 | 0 | -14,34 | **<0,001** | 0,46 | **IDF** | Oui |
| *kr-vs-kp* | 76,48 | 4,96 | 51,88 | 5,13 | 50 | 0 | 0 | 23,8 | **<0,001** | 1,47 | **C45** | Oui |
| *krkopt* | 5050 | 54,49 | 5006,3 | 117,2 | 31 | 19 | 0 | 2,32 | 0,024 | 1,01 | **C45** | Oui |
| *postoperative* | 71,22 | 7,25 | 71,34 | 7,21 | 25 | 25 | 0 | -1,03 | 0,309 | 1,00 | IDF | |
| *promoters* | 13,12 | 5,99 | 13,16 | 6,31 | 13 | 10 | 27 | -0,16 | 0,868 | 1,00 | IDF | |
| *splice* | 190,8 | 9,89 | 322,6 | 27,99 | 0 | 50 | 0 | -32,8 | **<0,001** | 0,59 | **IDF** | Oui |
| *tic-tac-toe* | 28,84 | 30,84 | 51,76 | 57,55 | 8 | 34 | 8 | -5,59 | **<0,001** | 0,56 | **IDF** | Oui |
| *zoo* | 16,4 | 1,58 | 15,16 | 3,35 | 32 | 13 | 5 | 2,76 | 0,008 | 1,08 | **C45** | Oui |
| *SCORE* | | | | | | | | | | 0,84 | 7-3 | 5-3 |

**Figure 3-18** Comparaison taille IDF-C4.5

L'analyse de la complexité des représentations produites (figure 3-18) suggère que la prise en compte explicite de l'hypothèse $H_0$ au cours de la construction de l'arbre de décision permet un élagage efficace de l'espace de recherche, comparable (sinon légèrement meilleur) à celui réalisé dans une étape distincte par C4.5.

Pour cette comparaison, il n'est pas indispensable de mener une étude expérimentale des temps d'exécution. En effet, en utilisant une procédure plus simple pour l'évaluation des tests, en arrêtant plus tôt la spécialisation des arbres de décision et en éliminant complètement la phase de post-élagage, IDF est indiscutablement plus rapide que C4.5 pour des réalisations informatiques comparables.

### 3.3.6 Autres variantes et extensions possibles de IDF

L'analyse présentée dans les sections précédentes a éludé, pour simplifier l'exposé, certaines solutions alternatives qui pourraient être prises en compte dans la construction de la fonction d'évaluation de IDF.

Un premier aspect qui mérite plus d'attention est l'allocation des probabilités *a priori* aux tests sur les attributs numériques. Nous avons utilisé dans la version présentée dans la section précédente une solution qui a l'avantage de rejoindre l'analyse de [Quinlan, 1996] et de garder la symétrie avec les attributs symboliques : des probabilités *a priori* inversement proportionnelles au nombre de seuils considérés. On peut reprocher à cette solution, à la différence du cas des attributs symboliques, que ce nombre n'est pas connu **avant** la présentation des exemples et ne devrait donc pas figurer dans l'expression des probabilités ***a priori***.

Une autre solution à ce problème, qui présente l'avantage d'être indépendante par rapport aux exemples, consiste dans l'allocation d'une distribution continue de probabilité *a priori* du choix du seuil sur le domaine entier de variation de l'attribut considéré, $p(v,i)$. Pour réaliser les desiderata présentés dans la section 3.3.3, $p(v,i)$ doit être constante pour un attribut $i$ et $\int_{min_i}^{max_i} p(v,i)dv = P(H_0)$. On obtient alors : $\frac{p(v,i)}{P(H_0)} = \frac{1}{max_i - min_i}$.

La probabilité qui sera effectivement considérée dans l'évaluation d'un test correspondra, dans cette version, à l'intégration de cette distribution entre les deux valeurs de l'attribut qui délimitent la marge de liberté dans le choix du seuil (la portion inoccupée entre deux projections consécutives $v_j$ et $v_{j+1}$ des exemples sur l'axe respectif) :

$$\frac{P(H_{ij})}{P(H_0)} = \frac{\int_{v_j}^{v_{j+1}} p(v,i)dv}{P(H_0)} = \int_{v_j}^{v_{j+1}} \frac{p(v,i)}{P(H_0)} dv = \frac{v_{j+1} - v_j}{max_i - min_i}$$

La fonction d'évaluation qui en résulte va donc préférer les partitions contrastées, ce qui constitue une façon de prendre en compte l'hypothèse de similarité dans l'induction des arbres de décision. Une version de IDF qui utilise cette nouvelle fonction d'évaluation (IDF.C) a été évaluée sur les bases de données numériques et les résultats sont présentés dans la figure 3-19.

| Base | IDF.C Moy | ± | C4.5 Moy | ± | > | < | = | t | p | Rapport | Meilleur | Statis[ent] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 5,63 | 1,62 | 5,59 | 1,82 | 21 | 25 | 4 | 0,17 | 0,863 | 1,01 | C45 | |
| *bupa* | 38,52 | 5,6 | 35,81 | 5,95 | 37 | 13 | 0 | 3,13 | 0,003 | 1,08 | **C45** | Oui |
| *diabetes* | 26,47 | 3,21 | 25,65 | 3,93 | 33 | 17 | 0 | 1,67 | 0,102 | 1,03 | C45 | |
| *glass* | 31,4 | 6,73 | 34,15 | 6,25 | 23 | 27 | 0 | -2,4 | 0,02 | 0,92 | **IDF.C** | Oui |
| *ionosphere* | 9,21 | 3,27 | 10,83 | 3,62 | 13 | 37 | 0 | -3,48 | 0,001 | 0,85 | **IDF.C** | Oui |
| *iris* | 3,8 | 4,21 | 4,46 | 4,14 | 19 | 17 | 14 | -2,61 | 0,012 | 0,85 | **IDF.C** | Oui |
| *letter* | 11,65 | 0,64 | 12,52 | 0,54 | 3 | 47 | 0 | -10,42 | **<0,001** | 0,93 | **IDF.C** | Oui |
| *new-thyroid* | 5,12 | 2,9 | 7,55 | 4,14 | 10 | 38 | 2 | -4,98 | **<0,001** | 0,68 | **IDF.C** | Oui |
| *segment* | 3,16 | 0,89 | 3,4 | 0,83 | 20 | 30 | 0 | -1,94 | 0,056 | 0,93 | IDF.C | |
| *sonar* | 23,95 | 8,62 | 25,62 | 8,79 | 23 | 27 | 0 | -1,48 | 0,142 | 0,93 | IDF.C | |
| *vehicle* | 28,84 | 3,28 | 27,02 | 3,22 | 34 | 16 | 0 | 3,57 | **<0,001** | 1,07 | **C45** | Oui |
| *vowel* | 17,9 | 2,79 | 22,3 | 3,23 | 6 | 44 | 0 | -7,97 | **<0,001** | 0,8 | **IDF.C** | Oui |
| *waveform* | 23,34 | 1,36 | 23,68 | 1,16 | 23 | 27 | 0 | -1,43 | 0,155 | 0,99 | IDF.C | |
| *waveform40* | 24,1 | 1,39 | 24,58 | 1,34 | 20 | 28 | 2 | -2,29 | 0,026 | 0,98 | **IDF.C** | Oui |
| *wine* | 8,56 | 4,76 | 7,67 | 4,43 | 26 | 23 | 1 | 1,48 | 0,145 | 1,11 | C45 | |
| *SCORE* | | | | | | | | | | 0,94 | 9-5 | 7-2 |

**Figure 3-19** Comparaison taux d'erreur IDF.C-C4.5

| Base | IDF.C Moy | ± | C4.5 Moy | ± | > | < | = | t | p | Rapport | Moins complexe | Statisent |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *breast-w* | 25,56 | 4,43 | 21,36 | 4,99 | 37 | 9 | 4 | 4,43 | **<0,001** | 1,2 | **C45** | Oui |
| *bupa* | 7,2 | 4,13 | 42,8 | 10,18 | 0 | 50 | 0 | -22,97 | **<0,001** | 0,17 | **IDF.C** | Oui |
| *diabetes* | 20,56 | 4,2 | 38,84 | 15,4 | 6 | 43 | 1 | -8,27 | **<0,001** | 0,53 | **IDF.C** | Oui |
| *glass* | 52,96 | 6,38 | 42,84 | 4,16 | 44 | 3 | 3 | 9,26 | **<0,001** | 1,24 | **C45** | Oui |
| *ionosphere* | 24,88 | 4,63 | 22,76 | 4,36 | 30 | 13 | 7 | 2,51 | 0,015 | 1,09 | **C45** | Oui |
| *iris* | 8,36 | 1,31 | 7,84 | 1,22 | 13 | 0 | 37 | 4,15 | **<0,001** | 1,07 | **C45** | Oui |
| *letter* | 3454,7 | 45,75 | 2165,9 | 35,09 | 50 | 0 | 0 | 156,27 | **<0,001** | 1,6 | **C45** | Oui |
| *new-thyroid* | 16 | 2,11 | 13,4 | 2,65 | 34 | 7 | 9 | 5,43 | **<0,001** | 1,19 | **C45** | Oui |
| *segment* | 94,08 | 7,57 | 77,2 | 6,8 | 46 | 3 | 1 | 12,77 | **<0,001** | 1,22 | **C45** | Oui |
| *sonar* | 29,52 | 4,4 | 25,12 | 2,87 | 39 | 3 | 8 | 6,51 | **<0,001** | 1,18 | **C45** | Oui |
| *vehicle* | 97,36 | 7,01 | 116,8 | 17,69 | 7 | 41 | 2 | -7,11 | **<0,001** | 0,83 | **IDF.C** | Oui |
| *vowel* | 224,48 | 9,9 | 177,16 | 8,74 | 50 | 0 | 0 | 23,15 | **<0,001** | 1,27 | **C45** | Oui |
| *waveform* | 312,84 | 26,07 | 485,36 | 29,07 | 0 | 50 | 0 | -30,58 | **<0,001** | 0,64 | **IDF.C** | Oui |
| *waveform40* | 360,16 | 31,3 | 534,32 | 20,31 | 0 | 50 | 0 | -32,57 | **<0,001** | 0,67 | **IDF.C** | Oui |
| *wine* | 12,28 | 2,05 | 9,92 | 2,03 | 41 | 5 | 4 | 8,15 | **<0,001** | 1,24 | **C45** | Oui |
| *SCORE* | | | | | | | | | | 1,01 | 5-10 | 5-10 |

**Figure 3-20** Comparaison taille IDF-C4.5

On peut constater un gain de précision de IDF.C par rapport à IDF, surtout sur les bases de données pour lesquelles, traditionnellement, la similarité "géométrique" des exemples est considérée comme importante. Sur ces bases de données PmBc réalise aussi des résultats meilleurs que C4.5, ce qui semble confirmer l'importance de la prise en compte de la similarité numérique dans la construction des fonctions d'évaluation, au moins pour une certaine catégorie de tâches d'apprentissage.

Les arbres de décision construits par IDF.C sont légèrement plus complexes que ceux construits par C4.5 (dans 10 des 15 tests, mais pour une augmentation globale de taille de seulement 1%, comme le montre la figure 3-20).

Par ses conséquences sur le choix des tests, cette modification de la fonction d'évaluation pour les attributs numériques peut être rapprochée des travaux présentés dans [Van de Merckt, 1993] et mentionnés dans la section 3.2.3. Ces travaux proposent la prise en compte de la similarité numérique dans la construction des arbres de décision par une combinaison entre l'entropie et une mesure du contraste de la partition issue des techniques d 'apprentissage non-supervisé. La formule proposée est, en utilisant les notations précédentes :

$CE(H_i) = \dfrac{(p_1 + n_1) \times (p_2 + n_2) \times (m_1 - m_2)^2}{(p+n) \times I(p_1, n_1, p_2, n_2)}$ où $m_1$ et $m_2$ représentent les moyennes des valeurs de l'attribut $i$ pour les deux branches du test considéré.

Par rapport à cette proposition, difficilement interprétable du point de vue probabiliste, notre approche présente l'avantage de découler directement de l'analyse bayesienne du choix des hypothèses.

Une autre extension possible de IDF concerne les tests sur les attributs symboliques. Les tests binaires construits actuellement par IDF semblent assez efficaces mais d'autres possibilités pourraient également être considérées, notamment la partition des exemples sur des branches correspondant à chacune des valeurs possibles de l'attribut considéré (comme dans C4.5). Nous avons essayé de généraliser directement la fonction d'évaluation construite pour les tests binaires, ce qui revient à remplacer le coefficient binomial $C_{p+n}^{p_1+n_1}$ par le coefficient multinomial correspondant à une partition multiple et à attribuer à $t_i$ la valeur 1 (une seule hypothèse considérée par attribut). Cette nouvelle variante de l'algorithme (IDF.M) s'est avérée moins performante que la version originale, ce qui suggère qu'une analyse plus approfondie de ce type de test devra être réalisée.

Enfin, dans la version actuelle de IDF, la décision de ramifier une branche et le choix du test se font en accord avec l'hypothèse la plus probable par rapport aux exemples. Cette prise de décision pourrait être effectuée d'une façon plus sophistiquée, en s'appuyant sur les risques associés aux différentes décisions. L'algorithme pourrait donc préférer parfois prendre une décision en accord avec une hypothèse moins probable mais qui minimise le risque global par rapport à la distribution de probabilité *a posteriori* des différentes hypothèses. Les risques pourraient être identifiés avec les taux d'erreur estimés ou prendre en compte aussi d'autres critères de qualité.

### 3.3.7 Conclusion

Nous avons proposé dans cette section un cadre d'analyse bayesienne du processus de choix des tests dans la spécialisation d'un arbre de décision. La fonction d'évaluation construite à partir de cette analyse fait apparaître les ingrédients principaux des heuristiques utilisées par C4.5, combinés d'une façon différente. Cette approche présente, selon nous, deux avantages importants :

- des fondements théoriques rigoureux qui permettent une dissection fine des composants de la fonction d'évaluation et ouvrent des nouvelles perspectives (pour un meilleur traitement des attributs numériques, par exemple);
- des performances de classification comparables (et même légèrement meilleures) à celles de C4.5 en éliminant le besoin d'une étape distincte et d'une fonction d'évaluation différente pour l'élagage;
- absence de tout paramètre d'exécution à réglage laborieux.

## 3.4 COPAC : coopération expert-machine pour l'acquisition des connaissances

### 3.4.1 Problématique

Les recherches présentées dans cette section visent à créer le cadre d'une véritable coopération expert-machine au cours de l'acquisition de connaissances. Avant de présenter les modèles de coopération étudiés, nous allons analyser succinctement les avantages et les faiblesses des deux méthodes classiques généralement utilisées dans ce domaine.

Le plus souvent, les connaissances qui sont à la base de l'architecture des Systèmes Experts sont acquises auprès des experts humains (*élicitation de connaissances*). L'ingénierie des connaissances et les recherches psychologiques ont abouti au développement d'un certain nombre de méthodes et d'outils pour l'élicitation des connaissances. Malheureusement, ces recherches ont aussi révélé les limitations de cette approche. Dans certains cas, les experts ne sont pas capables d'exprimer les connaissances qu'ils utilisent dans leurs processus de décision.

Et même quand ces connaissances sont finalement verbalisées et formalisées, elles peuvent s'avérer incomplètes ou imparfaites.

Des méthodes de vérification statique peuvent corriger certains types d'erreurs d'ordre syntactique dans la formulation des connaissances. La validation des bases de connaissances passe aussi par le test des systèmes experts sur des ensembles d'exemples représentatifs. Ces systèmes disposent généralement des mécanismes de génération d'explications qui permettent une analyse détaillée des traces de leur exécution et, théoriquement, la détection des disfonctionnements et la correction des erreurs. Dans la pratique, l'utilité de cette méthode est souvent restreinte par les limitations humaines dans l'analyse des cas nombreux, décrits par un nombre important de paramètres. Un certain biais culturel ou contextuel peut aussi empêcher l'expert de trouver les meilleures solutions.

Comme nous l'avons vu dans les sections précédentes, les algorithmes d'apprentissage automatique peuvent également être utilisés pour l'acquisition de connaissances. Ils cherchent une "bonne" généralisation des exemples disponibles dans l'espace de toutes les représentations possibles générées par leur formalisme de représentation des connaissances. Comme ils disposent d'habitude de peu ou, le plus souvent, d'aucune connaissance du domaine, les algorithmes d'apprentissage sont extrêmement sensibles au nombre, à la distribution et à la représentation des exemples disponibles.

Une recherche complète de la meilleure représentation est généralement prohibitive à cause de son coût calculatoire. Pour cette raison ces algorithmes réalisent des modifications incrémentales d'une représentation courante, fondées sur une évaluation heuristique, locale et indépendante du domaine, de leur qualité. Ceci introduit un biais qui rend la recherche faisable mais peut aussi mener les algorithmes d'apprentissage à omettre des solutions utiles et à rester prisonniers des optima locaux.

Pour pallier leurs limitations respectives, ces deux approches peuvent être intégrées dans des méthodologies unitaires, qui assurent les moyens d'un véritable dialogue expert-machine pendant l'acquisition de connaissances. Ces méthodologies sont fondées sur l'observation d'une certaine complémentarité entre les avantages et les limitations des méthodes classiques :

- les algorithmes d'apprentissage peuvent fournir à l'expert :

  - des hypothèses initiales, obtenues par induction sur un grand nombre de données, pour décrire des phénomènes initialement mal connus par l'expert;
  - des améliorations et des compléments de leurs connaissances, à l'aide des techniques de révision de théorie;

- l'expert peut fournir aux algorithmes d'apprentissage :
  - des "bonnes" initialisations (les connaissances élicitées) qui peuvent faciliter leur convergence vers des solutions optimales;
  - des solutions pour les régions de l'espace des exemples qui sont peu représentées dans l'ensemble d'apprentissage.

Finalement, nous soutenons que la coopération expert-machine peut aider les deux partenaires à surpasser les limitations induites par leurs biais respectifs.

Pour rendre possible une telle coopération, le formalisme utilisé pour la représentation des connaissances doit être simultanément accessible à l'expert et à la machine (comme ceux utilisés dans l'Intelligence Artificielle symbolique : règles, arbres de décision ou objets, par exemple). En plus de la possibilité de communiquer sa connaissance, chaque partenaire doit être capable de la défendre et, éventuellement, de changer les opinions de l'autre. Une forme de justification simple (souvent utilisée dans les discussions entre les experts eux-mêmes) consiste dans l'invocation d'exemples (pour défendre ses propres propositions) et de contre-exemples (pour infirmer les propositions de l'autre).

### 3.4.2 Exploitation des connaissances du domaine en apprentissage automatique

L'exploitation des connaissances du domaine constitue une préoccupation constante des recherches en apprentissage automatique. Les différentes approches proposées diffèrent par le statut des connaissances utilisées (sûres ou sujettes à la révision) et par l'importance des connaissances et des interventions de l'expert au cours de l'apprentissage. Nous allons nous limiter ici à une présentation succincte des approches issues de l'apprentissage symbolique supervisé.

Une première catégorie d'approches est représentée par les algorithmes d'apprentissage spécialisés dans un domaine d'activité bien délimité. C'est le cas, par exemple, du système

Meta-Dendral [Dietterich & Michalski., 1983], conçu pour la découverte des règles de spectroscopie de la masse moléculaire. Les représentations des connaissances et les stratégies de recherche utilisées par ce type d'algorithme reposent substantiellement sur les connaissances du domaine mais leur applicabilité est généralement restreinte à des problèmes très spécifiques.

Une autre catégorie de recherches s'est orientée vers l'utilisation des connaissances du domaine déjà confirmées (*background knowledge*) dans des algorithmes d'apprentissage d'applicabilité générale. La plupart des algorithmes de programmation logique inductive exploitent ce type de connaissances, représentées sous forme de relations explicites (FOIL [Quinlan, 1990]) ou implicites (GOLEM [Muggleton & Feng, 1992], FOIDL [Mooney & Califf, 1995]).

Nous distinguons aussi une troisième catégorie d'approches, qui visent à remédier aux défauts des connaissances obtenues auprès des experts humains. C'est le cas des algorithmes de révision de théorie présentés dans la section 2.4. Dans [Munteanu et al., 1995] et [Munteanu et al., 1996] nous avons proposé l'intégration de ce type d'algorithmes dans un système d'acquisition de connaissances par coopération homme-machine, fondé sur des révisions itératives d'une base de connaissances commune, effectuées tour à tour par les deux partenaires (figure 3-21).

La base de connaissances initiale peut être fournie par l'expert ou obtenue par induction automatique, à partir d'un ensemble d'exemples sélectionnés par lui. Ensuite, chaque partenaire peut proposer des modifications et les soumettre à l'appréciation de l'autre. Le dernier peut accepter ces propositions, les refuser ou les accepter en partie, en proposant d'autres modifications.

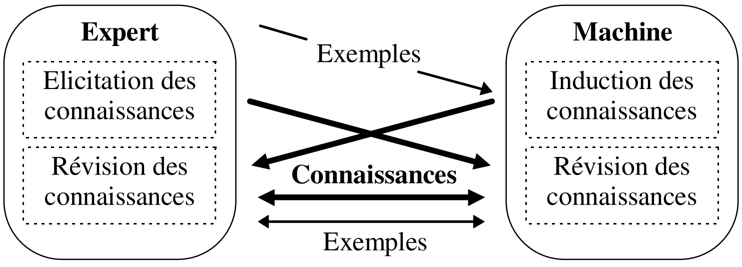



**Figure 3-21** Coopération expert-machine en acquisition de connaissances

Nous allons présenter ici une autre variante de ce modèle, qui préconise une intervention humaine directe au cours de toutes les étapes de l'induction même : génération et évaluation des hypothèses, arrêt de la recherche, élagage de la représentation finale, etc.

### 3.4.3 Présentation générale de l'environnement COPAC

COPAC est un environnement de découverte et mise au point interactive de connaissances, conçu pour atteindre les objectifs suivants :

1. proposer une méthodologie d'analyse rigoureuse et en même temps facile à appréhender;

2. permettre un contrôle humain direct et précis de l'analyse;

3. offrir des primitives automatisées pour décharger l'utilisateur des opérations répétitives ou laborieuses.

Compte tenu du premier objectif nous nous sommes orientés vers une représentation des connaissances à base de règles. Par rapport à d'autres formalismes de représentation des connaissances utilisés en apprentissage (arbres de décision, listes de décision), les règles présentent l'avantage d'être auto-explicatives et donc, potentiellement, plus facilement compréhensibles par les experts (voir section 2.3.2).

Une grande partie des algorithmes d'induction automatique de règles utilisent une stratégie de recherche de type "séparer pour régner". Dans COPAC nous avons adopté une stratégie descendante, similaire à celle utilisée par FOIL. Nous considérons que cette méthodologie, développée initialement pour l'apprentissage automatique, peut s'avérer aussi extrêmement utile pour structurer une analyse supervisée par un expert humain. A chaque itération, COPAC crée une règle vide (*SI vrai ALORS classe*) et rajoute successivement des conditions au corps de la règle, pour enlever des exemples négatifs couverts. Après avoir terminé la construction d'une règle, COPAC enlève tous les exemples positifs couverts de l'ensemble d'apprentissage et démarre, éventuellement, une nouvelle règle pour couvrir les exemples positifs restants. Cette stratégie est appliquée successivement pour les différentes classes et produit donc des représentations de classes qui peuvent se recouvrir partiellement.

Les décisions de contrôle principales dans une stratégie de type "séparer pour régner" descendante sont :

- le choix de la condition à rajouter au corps de la règle courante;
- la décision d'arrêter la spécialisation d'une règle ou la création des nouvelles règles;
- la décision d'abandonner une voie de recherche et de retourner en arrière, pour essayer une autre variante (*backtracking*).

En accord avec le deuxième objectif visé, l'utilisateur peut directement intervenir dans COPAC au niveau de toutes ces décisions de contrôle. Une stratégie de contrôle automatique est également disponible dans COPAC, chaque proposition du système étant soumise à l'acceptation de l'utilisateur.

Pour atteindre le troisième objectif, COPAC offre à l'utilisateur la possibilité d'effectuer automatiquement différentes opérations. Ces opérations sont encapsulées à différents niveaux de granularité qui vont du choix de la meilleure condition à rajouter au corps d'une règle en cours de construction jusqu'à l'induction d'une base complète de règles pour la tâche de classification considérée.

Du point de vue informatique, l'environnement COPAC est constitué de plusieurs modules interconnectés, réalisés en C++ :

- une bibliothèque de classes qui modélisent les concepts de base de la méthodologie de découverte de connaissances : exemples, conditions, règles, bases de règles, etc. et les opérations primitives sur ces concepts;
- un interpréteur de commandes extensible;
- plusieurs interfaces utilisateur (le prototype actuel dispose d'une interface en ligne de commande ergonomique avec vérification de la syntaxe et auto-complétion, rappel des commandes, etc.).

### 3.4.4 Principales commandes disponibles

Nous allons présenter dans cette section uniquement les commandes directement liées à l'acquisition de connaissances. Le système dispose en outre des commandes auxiliaires pour le chargement du contexte, les sauvegardes, la consultation de l'état courant, les demandes d'aide, etc.

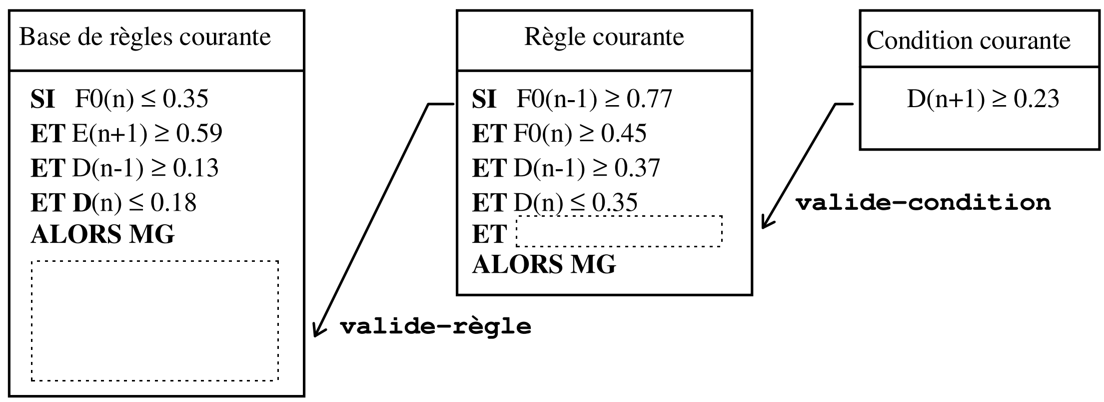



**Figure 3-22** Structures temporaires et validation dans COPAC (exemple d'induction interactive de règles d'étiquetage prosodique pour les mots grammaticaux)

***Construction des conditions***

Le choix de la condition à rajouter au corps de la règle en cours de construction est l'opération de base dans COPAC. Pour apprécier la qualité d'une condition, COPAC se base sur un critère de qualité préalablement choisi par l'utilisateur. Le gain informationnel (tel qu'il est défini dans FOIL pour l'induction de règles) est utilisé par défaut mais d'autres critères comme l'estimateur de Laplace (utilisé dans CN2) peuvent être invoqués.

Une information plus détaillée est fournie à l'utilisateur (qui est responsable de la décision finale) : la valeur du critère de qualité, les distributions inter-classes des exemples couverts, l'évolution de ces distributions par rapport à la situation d'avant l'ajout de la condition à évaluer (dans le reste de cette section nous allons appeler ces informations "les statistiques"). L'utilisateur dispose en outre de la possibilité de consulter les exemples positifs (**`montre-exemples-couverts`**) ou négatifs (**`montre-contre-exemples-couverts`**), couverts par la règle courante.

Pour choisir une condition, l'utilisateur explore généralement plusieurs pistes à l'aide de la commande **essaie**. En fonction du degré d'automatisation souhaité, plusieurs variantes de cette commande sont disponibles :

**essaie ATTRIBUT OPERATEUR VALEUR**

**essaie ATTRIBUT OPERATEUR ~VALEUR**

**essaie ATTRIBUT OPERATEUR ***

**essaie ATTRIBUT ***

**essaie ***

Le système cherche et fournit les statistiques pour la meilleure condition qui satisfait les éventuelles contraintes imposées par l'utilisateur. L'étoile signale les parties de la condition qui peuvent être librement choisies par COPAC. Pour les conditions numériques, le symbole **~** indique à COPAC une valeur approximative autour de laquelle il doit chercher la valeur la plus proche qui réalise un maximum local du critère de qualité (cette facilité est très utile pour l'affinement des seuils numériques, une opération qui s'avère difficile pour les humains).

L'utilisateur peut demander d'autres propositions, dans l'ordre décroissant du critère de qualité, à l'aide de la commande **suivante**. La commande **meilleure** lui permet de reprendre la consultation de la liste des propositions à partir de la meilleure condition trouvée par COPAC.

Après avoir évalué différentes variantes à l'aide de la commande **essaie**, la condition considérée meilleure par l'utilisateur est rajoutée au corps de la règle courante par la commande **valide-condition**. Le retour arrière peut être demandé par l'intermédiaire de la commande inverse : **invalide-condition**.

***Construction de règles***

Les règles peuvent être construites d'une manière interactive (comme décrit en haut) ou automatiquement par la commande **propose_règle**. Au moment où l'utilisateur est satisfait par la règle courante, celle-ci est rajoutée au paquet de règles par la commande **valide_règle** (le retour arrière peut être demandé par l'invocation de la commande inverse : **invalide-règle**). L'induction automatique d'un paquet entier de règles de classification est réalisé par la commande **propose_paquet**. Aucune mesure n'est prise pour éviter le recouvrement partiel des règles correspondant aux différentes classes, que nous considérons nécessaire dans certaines situations (voir discussion sur le recouvrement des règles dans PmBc, section 3.2.1).

Il n'est généralement pas recommandé de continuer la spécialisation (ou le rajout) de règles jusqu'à la modélisation parfaite de tous les exemples d'apprentissage disponibles à cause du risque de sur-spécialisation, qui diminue le pouvoir prédictif des représentations obtenues. Dans COPAC, nous proposons deux stratégies d'arrêt complémentaires pour diminuer ce risque. Le principe MDL, tel qu'il est utilisé dans FOIL (section 2.3.4), contrôle l'arrêt de la spécialisation au cours de l'induction automatique et prévient l'utilisateur du risque de sur-spécialisation au cours de l'induction interactive. Dans l'induction interactive, la décision d'arrêter la spécialisation est prise par l'utilisateur, qui a également la possibilité d'examiner les contre-exemples pour apprécier s'il s'agit des cas atypiques ou bruités qui peuvent être ignorés ou des cas demandant un affinement supplémentaire de la modélisation.

***Modification de la description des exemples***

Les résultats de l'analyse peuvent suggérer à l'expert certaines modifications dans la description des exemples qui pourraient faciliter l'induction de règles ou simplifier leur représentation. Pour réaliser ces modifications il dispose des commandes :

**`ignore ATTRIBUT`**

**`considère ATTRIBUT_IGNORÉ`**

**`crée-attribut-virtuel FORMULE(ATTRIBUTS)`**

où les formules utilisables sont répertoriées dans une bibliothèque prédéfinie.

Pour l'instant, l'initiative pour la modification de la description des exemples revient exclusivement à l'utilisateur. Des techniques d'induction constructive pourraient être envisagées pour permettre au système de proposer automatiquement de telles modifications.

***Evaluation de la représentation***

Les performances de classification du paquet de règles construit sont évaluées sur un ensemble indépendant d'exemples par les commandes **`test_paquet`** qui fournit des informations synthétiques (matrice de confusion, taux de détection et fiabilité pour chaque classe) sur le paquet entier ou **`test_règle`** qui permet la focalisation sur les performances d'une règle particulière.

### 3.4.5 Conclusion

COPAC a été conçu pour permettre aux experts une intervention plus directe au cours d'un processus d'induction de connaissances assisté par ordinateur. Ces travaux reposent sur la supposition que l'apport des connaissances et du raisonnement des experts peut contribuer à améliorer les performances globales d'un algorithme d'induction.

Il est évident que la validation expérimentale d'une telle approche est beaucoup plus difficile que dans le cas des algorithmes d'apprentissage purement automatique. A titre d'exemple, mentionnons que les résultats présentés dans le chapitre suivant (induction interactive d'une base de règles pour la discrimination des mots grammaticaux sur des critères prosodiques) ont été obtenus dans une séance de travail de plusieurs heures avec l'expert.

L'utilisation de l'environnement COPAC pour différents problèmes pratiques de recherche en parole (voir section 4.1.6) a fait apparaître deux autres applications intéressantes. Premièrement, en imposant une méthodologie d'analyse rigoureuse et des outils d'exploration automatique, COPAC permet aux chercheurs d'aborder des problèmes peu étudiés auparavant (pour lesquels même les connaissances expertes sont pratiquement inexistantes) et d'acquérir de nouvelles connaissances. Nous avons observé que la participation directe au processus d'induction facilite l'acceptation et la compréhension des résultats par les experts.

Deuxièmement, cette dissection fine d'un processus d'induction, examiné pendant son exécution sur des tâches pratiques, peut s'avérer intéressante pour la pédagogie et pour la recherche en apprentissage automatique. Dans la réalisation informatique de cet environnement nous avons recherché une extensibilité maximale autour d'un noyau d'opérations de base en apprentissage, ce qui fait de COPAC une plate-forme d'expérimentation et de diagnostic ergonomique pour de nouvelles idées dans ce domaine.

# Chapitre 4 :
# APPLICATIONS EN PAROLE

Pour évaluer les méthodes proposées dans le chapitre précédent, il nous a semblé absolument nécessaire de les confronter à des problèmes concrets et représentatifs de recherche en parole. Nous avons donc participé activement à des études ayant comme but l'acquisition de connaissances prosodiques pour l'amélioration de l'accès lexical et la discrimination des consonnes occlusives sourdes.

## 4.1 Acquisition de connaissances prosodiques pour l'amélioration de l'accès lexical

Le travail présenté dans cette section a été réalisé en étroite collaboration avec des chercheurs de l'équipe GEOD du laboratoire CLIPS-IMAG (l'ancienne équipe Reconnaissance et Dialogue de l'ICP), intéressés par le test et la mise au point d'un agent d'étiquetage prosodique dans le cadre du développement d'un système multi-agents de compréhension de la parole (MICRO).

Nous allons nous intéresser ici essentiellement aux problèmes liés à l'acquisition de connaissances et à l'utilisation de l'apprentissage automatique. Mais, pour la bonne compréhension du cadre de ces recherches, nous allons décrire sommairement le système MICRO de compréhension de la parole et les agents prosodiques (qui constituent l'objet d'une thèse de doctorat [Caillaud, 1996]), dans les paragraphes suivants.

### 4.1.1 Le système MICRO de compréhension de la parole

L'architecture du système MICRO de compréhension de la parole (figure 4-1) suit le paradigme multi-agents et fait apparaître deux voies distinctes de traitement des informations extraites de la parole :

- une voie analytique, classique dans les systèmes de reconnaissance de la parole, qui analyse la parole aux niveaux :

  - *Acoustique* : ce groupe d'agents correspond au système auditif humain et fournit aux étages supérieurs des traits acoustiques extraits du signal de parole;

- *Phonétique* : ce groupe d'agents réalise un processus stochastique de reconnaissance de la parole et fournit des hypothèses phonétiques au groupe linguistique;

- *Linguistique* : ce groupe d'agents construit des représentations conceptuelles des phrases pour faciliter leur interprétation;

- une voie holistique (le groupe d'agents *Prosodie*) qui doit fournir rapidement à la voie analytique les informations apportées par la prosodie. Elle est constituée par les agents :

  - *noyaux vocaliques* qui réalise la détection des noyaux vocaliques et la stylisation correspondante des paramètres prosodiques (section 4.1.2);

  - *pseudo-syllabes* qui stocke les données produites par l'agent *noyaux vocaliques* et les tient à la disposition des agents *prosodie des mots* et *intonation*;

  - *prosodie des mots* qui doit fournir rapidement des informations sur les frontières et les catégories lexicales des mots;

  - *intonation* qui produit des informations de haut niveau (comme la classification en actes de dialogue, par exemple) sur la base de la prosodie du dialogue.

Enfin, ces deux voies de traitement convergent vers le groupe d'agents de *Dialogue* qui gère les stratégies de dialogue en accord avec les informations de nature pragmatique.

Dans les sections suivantes nous allons nous focaliser sur la présentation des deux agents prosodiques qui sont directement concernés par notre étude.

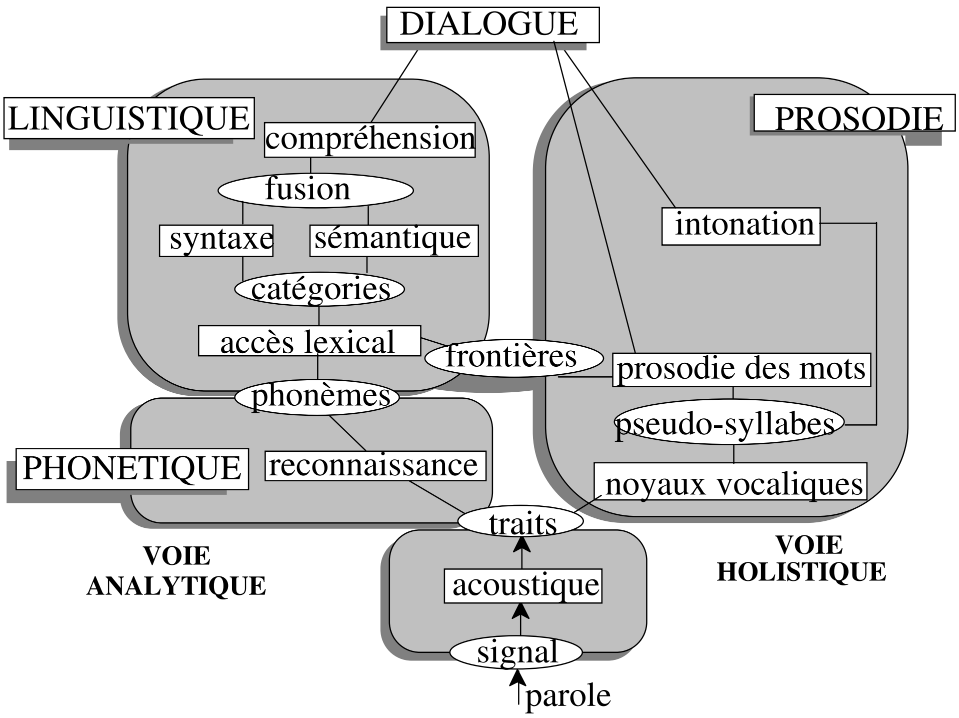



**Figure 4-1** Architecture du système MICRO

### 4.1.2 L'agent de détection des noyaux vocaliques

La prosodie est associée à l'évolution temporelle de trois paramètres différents du signal de parole : la fréquence fondamentale, l'énergie, la durée des syllabes et des pauses. Bien que la fréquence fondamentale et l'énergie évoluent continuellement dans la parole, il est généralement admis que ces paramètres peuvent être représentés d'une façon discrète et compressée, par une stylisation, sans perte importante d'information. Il subsiste cependant des différences importantes quant au choix de l'unité de stylisation, au calcul des valeurs pertinentes pour l'unité choisie, aux méthodes de normalisation inter- et intra-locuteurs et à la taille de la fenêtre d'analyse.

Le rôle de la voie holistique dans MICRO impose le choix d'une unité de stylisation dont la localisation ne demande pas l’intervention préalable du décodage acoustico-phonétique. Pour cette raison, les concepteurs de ce système ont décidé d'utiliser les noyaux vocaliques pour segmenter le signal dans des pseudo-syllabes, qui représentent des approximations acceptables des syllabes réelles, et de calculer les paramètres prosodiques à partir de ces unités de stylisation.

La détection des noyaux vocaliques est fondée sur la recherche des zones d’énergie maximale dans le signal, correspondant aux centres des voyelles. Le principe de l'algorithme de détection des noyaux vocaliques et la correspondance entre les pseudo-syllabes ainsi délimitées et les syllabes réelles sont présentés dans la figure 4-2.

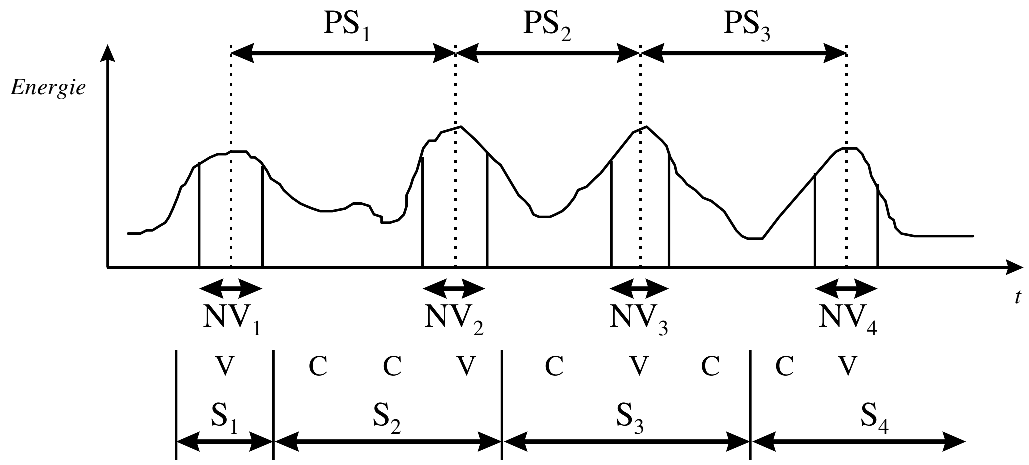



**Figure 4-2** Correspondance entre syllabes et pseudo-syllabes ($NV_i$ : noyaux vocaliques, $PS_i$ : pseudo-syllabes, $S_i$ : syllabes, C: consonne et V : voyelles).

Après de nombreuses expérimentations (détaillées dans [Caillaud, 1996]), la taille de la fenêtre d'analyse a été fixée à trois noyaux vocaliques successifs et le choix des paramètres prosodiques qui seront utilisés pour la démarcation lexicale a porté sur :

- les fréquences fondamentales maximales des trois noyaux vocaliques, codées par rapport aux droites de régression et normalisées sur une phrase entière;
- les différences entre la fréquence fondamentale du noyau central et celles de ses deux voisins;
- les énergies des trois noyaux, mesurées au 2/3 de leur largeur et normalisées par phrase;
- les durées des pseudo-syllabes dont les frontières gauches correspondent aux trois noyaux considérés, normalisées par phrase;
- les durées des pauses (éventuelles) consécutives aux trois noyaux vocaliques, normalisées sur le corpus entier disponible pour un locuteur.

Cette forme de stylisation associe donc à chaque noyau vocalique quatorze paramètres prosodiques (cinq pour la fréquence fondamentale, trois pour l'énergie, trois pour la durée et trois pour les pauses).

### 4.1.3 L'agent de démarcation prosodique

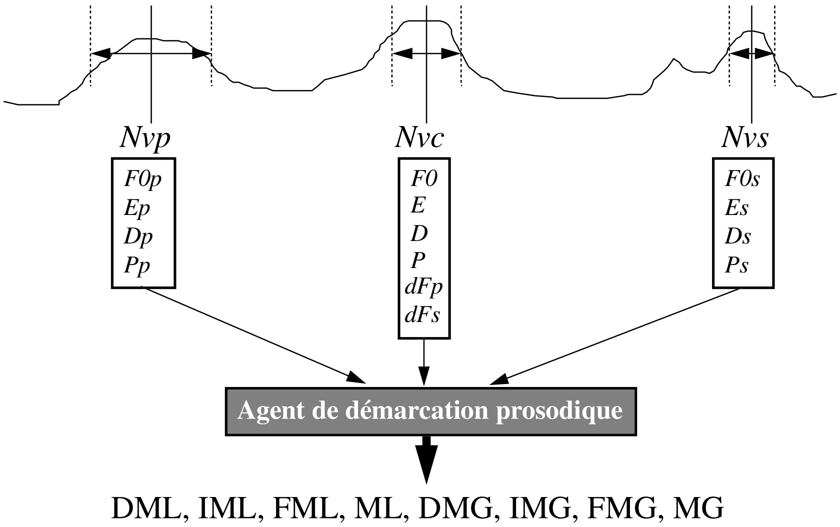



**Figure 4-3** La tâche de l'agent de démarcation

Le but de cet agent est de produire des informations sur le découpage lexical de l'énoncé. A partir des informations prosodiques stylisées produites par l'agent *noyaux vocaliques*, cet agent doit associer aux noyaux vocaliques des marqueurs indiquant le type d'événement détecté (figure 4-3) :

- DML : noyau correspondant au début d'un mot lexical,
- IML : noyau correspondant à une syllabe interne de mot lexical,
- FML : noyau correspondant à la fin d'un mot lexical,
- ML : noyau correspondant à un mot lexical monosyllabique (ne comportant qu'un seul noyau),
- DMG : noyau correspondant au début d'un mot grammatical,
- IMG : noyau correspondant à une syllabe interne d'un mot grammatical.

- FMG : noyau correspondant à la fin d’un mot grammatical,
- MG : noyau correspondant à un mot grammatical monosyllabique,

Ces informations peuvent guider la voie analytique du système MICRO de deux manières distinctes :

- en tant que **point d’ancrage** : ces informations constitueraient le squelette de la voie analytique. Même si des erreurs intervenaient au cours de l'analyse, le système pourrait reprendre en toute sécurité le traitement après le point d'ancrage suivant, à condition que celui-ci ait été correctement détecté. En outre, l'étiquetage produit par cet agent pourrait invalider certaines conclusions de la voie analytique et la forcer à explorer d'autres possibilités. Il est donc évident que, pour pouvoir être utilisé de cette façon, l'agent prosodique doit produire des informations très fiables, même si elles ne sont pas nombreuses.
- pour le **classement** des hypothèses lexicales : dans cette variante, les informations fournies par cet agent pourraient renforcer la priorité des hypothèses lexicales compatibles avec la prosodie et permettre au système d’explorer en premier les hypothèses ayant le plus de chances d’aboutir à la solution. Les contraintes de fiabilité pourraient donc être affaiblies dans ce cas, sans toutefois renoncer à l'apport des informations prosodiques pour l'analyse.

La figure 4-4 montre la façon dont l'agent de démarcation peut infirmer (ou rendre moins probable) une hypothèse lexicale (Hypothèse 1), qui semble en désaccord avec les indices prosodiques, pour la deuxième démarcation de mots proposée.

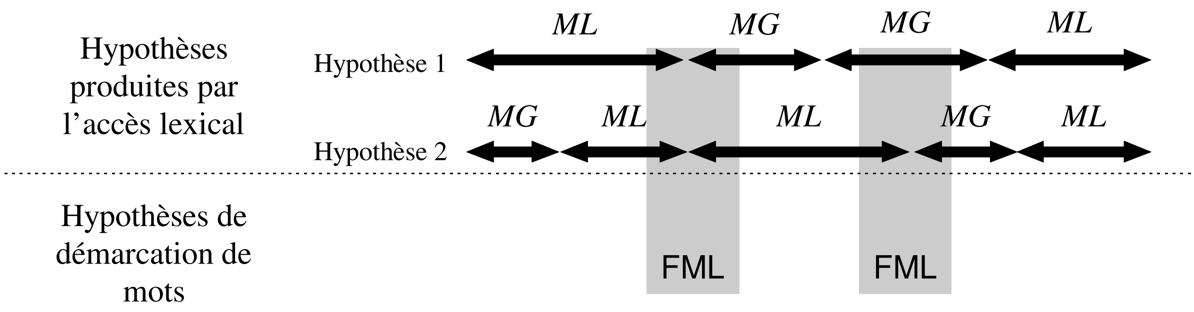



**Figure 4-4** Mise en correspondance des informations produites par l’accès lexical et l’agent prosodique de démarcation de mots

### 4.1.4 Description du corpus de référence

L'agent de démarcation réalise essentiellement une tâche de classification. Pour produire et pour valider les connaissances nécessaires à cet agent nous avons utilisé un corpus de signaux de parole extrait de BDSONS [Descout et al., 1986]. Ce corpus correspond à du texte lu (cinq phrases), prononcé par neuf locuteurs différents (six hommes et trois femmes).

Pour estimer la qualité intrinsèque des informations de démarcation lexicale qui peuvent être extraites à partir des indices prosodiques (sans tenir compte des performances de l'agent de détection des noyaux vocaliques), nous avons directement utilisé les voyelles, comme unités de stylisation, qui peuvent être repérées avec précision à partir des étiquettes phonétiques disponibles dans BDSONS. Les paramètres prosodiques mentionnés dans la section 4.1.2 ont été calculés pour chacune de ces voyelles.

La transcription orthographique du corpus a été étiquetée par l'expert en prosodie avec des informations de démarcation lexicale. Ensuite, une procédure semi-automatique d'alignement a été utilisée pour associer à chaque voyelle une étiquette de démarcation lexicale. On peut observer dans la figure 5-5 que les distributions des différentes classes d'étiquettes, dans le corpus de référence, sont très variées et que l'étiquette IMG est absente (il n'existe pas de mot grammatical avec plus de deux syllabes dans ce corpus).

| DML | IML | FML | ML | DMG | FMG | MG | Total |
|---|---|---|---|---|---|---|---|
| 345 | 141 | 345 | 150 | 27 | 27 | 556 | 1591 |

**Figure 4-5** Distributions des classes d'étiquettes de démarcation lexicale

### 4.1.5 Induction de connaissances prosodiques à partir des exemples

Sur l'ensemble d'exemples ainsi produits nous avons mené une étude expérimentale sur différentes méthodes de classification automatique. A part l'objectif direct d'évaluer la qualité des informations de démarcation lexicale pouvant être obtenues à partir des indices prosodiques, nous nous sommes intéressés aussi à la comparaison des performances de classification de ces différentes approches. Dans cette étude, nous avons rajouté à la batterie d'algorithmes déjà présentés (C4.5, CN2, PmBc, IDF.C), deux autres méthodes de classification automatique, largement utilisées dans la recherche en parole : CGL et NOPT.

CGL (pour classificateur gaussien linéaire) est une méthode statistique de classification paramétrique simple mais qui s'avère suffisamment performante pour de nombreuses applications pratiques [Gascuel & Gallinari, 1994]. Elle est fondée sur l'hypothèse que les exemples appartenant à une même classe apparaissent aléatoirement, en suivant une distribution de probabilité gaussienne autour d'un prototype. A partir des exemples d'apprentissage, cette méthode estime les paramètres des distributions de probabilité des exemples correspondant aux différentes classes et utilise le théorème de Bayes pour affecter des nouveaux exemples à la classe la plus probable. Cette procédure de classification induit généralement des frontières entre les classes d'ordre quadratique. Une simplification des calculs peut être réalisée en se basant sur l'hypothèse que les matrices de covariances sont égales pour les différentes classes. Les classificateurs qui en résultent réalisent alors une discrimination linéaire. Nous avons utilisé dans ces expériences l'implémentation d'un classificateur gaussien, proposée dans le paquet de méthodes statistiques de classification TOOLDIAG [Rauber *et al.,* 1993].

NOPT est un algorithme connexioniste fondé sur l'optimisation du gradient conjugué, développé dans le cadre de recherches en parole [Barnard & Cole, 1989]. Comme dans la plupart des approches connexionistes, la structure du réseau de neurones joue un rôle important dans les performances de classification de NOPT. Nous avons décidé d'utiliser une structure classique de réseau à trois couches et, pour choisir un nombre d'unités cachées approprié pour chaque tâche de classification, nous avons mené des essais préliminaires, en utilisant des découpages particuliers du corpus d'exemples en ensembles d'apprentissage et ensembles de test.

Comme nous n'avons pas de raison de croire que toutes les classes d'étiquettes de démarcation lexicale sont repérables avec une égale facilité à partir des indices prosodiques, nous avons préféré réaliser plusieurs expériences de classification binaire correspondant à chacune de ces classes. Au cours des expérimentations, deux regroupements d'étiquettes se sont avérés bénéfiques pour la classification : *ML+FML* (*ML'*), *MG+DMG* (*MG'*). Le protocole expérimental est similaire à ceux utilisés dans le chapitre précédent : pour chacune des neuf tâches de classification, les différents algorithmes ont été entraînés sur 80% des exemples choisis aléatoirement parmi tous les exemples disponibles et testés sur le reste de 20%. Cette procédure a été répétée 50 fois, pour des découpages différents des exemples, et les taux d'erreur résultants sont présentés dans la figure 5-6 en termes de moyenne et d'écart-type.

| Tâche | PmBc | | IDF.C | | C45 | | CN2 | | NOPT | | CGQ | | CGL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± |
| *DML* | 20,27 | 2,15 | 22,08 | 2,15 | 22,06 | 2,18 | 20,93 | 2,237 | 21,05 | 2,12 | 26,05 | 2,58 | 21,82 | 2,00 |
| *IML* | 8,36 | 1,48 | 9,38 | 1,75 | 9,94 | 2,03 | 8,91 | 1,638 | 9,11 | 1,87 | 27,94 | 2,734 | 9,70 | 1,70 |
| *FML* | 15,40 | 2,02 | 17,32 | 2,19 | 17,10 | 1,90 | 18,82 | 2,294 | 16,97 | 1,78 | 32,44 | 2,783 | 18,81 | 1,92 |
| *ML* | 9,05 | 1,33 | 9,45 | 1,21 | 9,68 | 1,60 | 9,21 | 1,499 | 9,69 | 1,62 | 26,19 | 2,855 | 9,90 | 1,47 |
| *DMG* | 1,71 | 0,56 | 1,89 | 0,63 | 1,95 | 0,64 | 1,72 | 0,566 | 1,71 | 0,57 | 4,67 | 1,139 | 1,85 | 0,64 |
| *FMG* | 1,72 | 0,67 | 1,84 | 0,74 | 2,23 | 0,86 | 1,72 | 0,689 | 1,72 | 0,69 | 3,47 | 1,099 | 2,24 | 0,82 |
| *MG* | 23,22 | 2,06 | 25,30 | 2,00 | 24,98 | 2,27 | 25,49 | 2,371 | 22,82 | 2,06 | 39,65 | 2,636 | 27,76 | 2,27 |
| *ML'* | 13,18 | 1,63 | 14,14 | 2,00 | 15,16 | 1,67 | 16,58 | 1,563 | 14,45 | 1,88 | 20,73 | 2,463 | 17,01 | 1,47 |
| *MG'* | 22,75 | 2,40 | 24,78 | 2,07 | 24,00 | 2,03 | 25,3 | 2,22 | 22,40 | 1,88 | 39,29 | 3,206 | 26,48 | 2,13 |

**Figure 4-6** Résultats (taux d'erreur) des différents algorithmes

Les différences entre les sept algorithmes ne sont pas spectaculaires et il faut s'assurer qu'elles présentent une signification statistique, en utilisant un test approprié. Un travail important de mise au point de la représentation et du codage optimal des exemples a été nécessaire pour arriver à ces scores. Dans ces tâches, une différence de quelques pourcents devient donc très importante, si elle est validée statistiquement.

Pour analyser leurs performances relatives, nous avons réalisé des tests de Student (échantillons appariés) de comparaison des moyennes pour chaque paire d'algorithmes ($p<0.05$). Nous proposons la méthodologie suivante pour présenter ces résultats d'une façon synthétique : le meilleur algorithme et tous ceux dont les taux d'erreur ne diffèrent pas statistiquement des siens sont placés dans un premier groupe. Le premier algorithme qui s'avère statistiquement moins performant commence un deuxième groupe et ainsi de suite. Les résultats obtenus en suivant cette méthodologie sont présentés dans la figure 4-7. La dernière ligne du deuxième tableau indique aussi les moyennes des rapports des taux d'erreur entre chaque algorithme et PmBc (rubrique MTR/PmBc).

| Tâche | Algorithmes | (Légende : | **1er groupe** | 2ème groupe | 3ème groupe | 4ème groupe | **5ème groupe** |
|---|---|---|---|---|---|---|---|
| *DML* | **PmBc** | CN2 | NOPT | CGL | C45 | IDF.C | CGQ |
| *IML* | **PmBc** | CN2 | NOPT | IDF.C | CGL | C45 | **CGQ** |
| *FML* | **PmBc** | NOPT | C45 | IDF.C | CGL | CN2 | CGQ |
| *ML* | **PmBc** | **CN2** | IDF.C | C45 | NOPT | CGL | CGQ |
| *DMG* | **NOPT** | **PmBc** | CN2 | CGL | IDF.C | C45 | CGQ |
| *FMG* | **NOPT** | **PmBc** | **CN2** | IDF.C | C45 | CGL | CGQ |
| *MG* | **NOPT** | **PmBc** | C45 | IDF.C | CN2 | CGL | CGQ |
| *ML'* | **PmBc** | IDF.C | NOPT | C45 | CN2 | CGL | **CGQ** |
| *MG'* | **NOPT** | **PmBc** | C45 | IDF.C | CN2 | CGL | **CGQ** |

| | PmBc (I) | NOPT (II) | CN2 (III) | IDF.C (IV) | C4.5 (V) | CGL (VI) | CGQ (VII) |
|---|---|---|---|---|---|---|---|
| **1er groupe** | **9** | **4** | **2** | | | | |
| 2ème groupe | | 5 | 4 | 5 | 4 | | |
| 3ème groupe | | | 2 | 4 | 4 | 6 | |
| 4ème groupe | | | 1 | | 1 | 3 | 6 |
| **5ème groupe** | | | | | | | **3** |
| MTR(/PmBc) | 1,00 | 1,04 | 1,09 | 1,09 | 1,13 | 1,18 | 2,15 |

**Figure 4-7** Comparaison des algorithmes d'apprentissage

PmBc s'avère indiscutablement le plus performant algorithme d'apprentissage pour l'étiquetage prosodique, en dépassant le réseau de neurones (NOPT), qui vient en deuxième position. En effet, il est le seul à faire partie du groupe de tête pour toutes les tâches de classification étudiées. IDF.C dépasse légèrement C4.5 aussi bien en nombre d'apparitions dans le deuxième groupe qu'en termes de taux d'erreur moyens mais il est à son tour dépassé par CN2 en nombre d'apparitions dans les premiers groupes. Enfin, les classificateurs gaussiens, surtout la version quadratique, semblent les moins performants pour ces tâches.

Le taux global d'erreur est un indicateur synthétique très utile pour les comparaisons inter-algorithmes sur une même tâche de classification. Il ne peut cependant pas être utilisé directement pour comparer les performances de classification entre les différentes tâches, à cause des distributions des exemples qui varient beaucoup entre les classes d'étiquettes prosodiques. A sa place, les spécialistes du domaine se basent sur deux autres indicateurs de performance, définis comme suit :

le taux de détection $\dfrac{P_p}{P_p + P_n}$

la fiabilité des détections $\dfrac{P_p}{P_p + N_p}$

où $P_p$ et $N_n$ désignent les nombres d'exemples positifs et négatifs bien classés, et $P_n$, $N_p$ représentent les nombres d'exemples positifs et négatifs mal classés. Les moyennes et les écarts types de ces deux indicateurs sont présentés dans la figure 4-8 pour les classificateurs proposés par PmBc.

| Tâche | Taux de détection | | Fiabilité | |
|---|---|---|---|---|
| | Moy | ± | Moy | ± |
| *DML* | 22,05 | 6,319 | 59,45 | 9,757 |
| *IML* | 12,47 | 5,939 | 66,05 | 25,297 |
| *FML* | 46,69 | 6,692 | 73,20 | 8,550 |
| *ML* | 13,03 | 5,688 | 56,74 | 19,183 |
| *DMG* | 3,65 | 15,172 | 92,00 | 27,405 |
| *FMG* | 2,90 | 14,493 | 94,00 | 23,990 |
| *MG* | 52,69 | 5,513 | 73,92 | 4,886 |
| ***ML+FML*** | **70,59** | **5,046** | **84,49** | **3,575** |
| ***MG+DMG*** | **56,04** | **5,715** | **76,04** | **5,739** |

**Figure 4-8** Comparaison des tâches de classification (résultats de PmBc)

Il en résulte que la détection des différentes classes d'étiquettes de démarcation lexicale pose des problèmes de difficultés variées aux méthodes de classification automatique. Les taux de détection très bas des DMG et des FMG peuvent être expliqués en partie par la faible représentation des mots grammaticaux plurisyllabiques dans le corpus de référence. Une explication similaire pourrait être envisagée aussi pour les ML et les IML, en gardant les proportions. Il est cependant évident que les distributions des classes ne peuvent pas expliquer complètement les différences observées. En effet, PmBc détecte deux fois plus de FML que de DML, avec une plus grande fiabilité, en dépit de leurs distributions identiques dans le corpus de référence. Il en résulte donc que le marquage prosodique est plus accentué pour certaines classes d'informations de démarcation lexicale.

Le gain de précision des classificateurs obtenus pour les regroupements d'étiquettes ML+FML et MG+DMG par rapport à chacune des classes initiales fait ressortir une autre conclusion intéressante : certaines de ces classes, définies uniquement sur des critères lexicaux, partagent un encodage prosodique similaire. En effet, les mots lexicaux monosyllabiques se comportent, du point de vue prosodique, comme les fins de mots lexicaux plurisyllabiques, tandis que les mots grammaticaux monosyllabiques semblent plutôt proches des débuts des mots grammaticaux plurisyllabiques. Ces observations sont en accord avec les prévisions des experts. Pour simplifier l'exposé nous nous sommes limités ici à l'étude des résultats de PmBc mais l'analyse des résultats des autres méthodes de classification utilisées conduit à des conclusions similaires.

Du point de vue du projet MICRO, l'utilisation des informations prosodiques pour le classement des hypothèses lexicales semble déjà réalisable. En effet, même si on se limite aux classes les plus marquées prosodiquement (*ML+FML* et *MG+DMG*), on obtient un taux approximatif de 45% de voyelles qui sont étiquetées d'une manière assez fiable. Ce type d'utilisation pourrait donc accélérer considérablement la vérification des hypothèses lexicales, surtout pour des applications avec un vocabulaire important. L'utilisation de ces informations en tant que points d'ancrage pour la voie analytique demanderait encore plus de fiabilité, que nous ne sommes pas encore en mesure d'obtenir sans une réduction importante du taux de détection.

Les algorithmes d'apprentissage symbolique, et notamment PmBc, ont été également utilisés pour évaluer les différents codages possibles des paramètres prosodiques, l'utilisation directe des noyaux vocaliques à la place des voyelles et l'influence de chacun des locuteurs du corpus d'apprentissage sur les résultats des classificateurs obtenus. Ces expériences, nécessaires pour la mise au point des agents prosodiques et pour l'évaluation de l'apport de la prosodie dans la reconnaissance de la parole, n'apportent pas d'élément nouveau pour notre étude. Elles sont décrites dans la thèse consacrée à cette application [Caillaud, 1996].

### 4.1.6 Perspectives en acquisition de connaissances prosodiques par coopération expert-machine

Une autre piste d'investigation que nous avons poursuivie pour l'acquisition des connaissances prosodiques concerne l'apport potentiel des connaissances des experts en prosodie. Nous allons fournir dans cette section des résultats expérimentaux obtenus dans le cadre de cet étude. Il faut préciser cependant que la validation statistique de ces résultats est beaucoup plus problématique que dans le cas des méthodes purement automatiques. Ces expériences demandent beaucoup de temps et d'effort de la part des experts, ce qui rend difficile leur répétition sur la même tâche de classification, avec des méthodologies différentes. Notre objectif est donc plutôt de fournir des indices pour soutenir l'intérêt des approches coopératives dans l'acquisition de connaissances que d'apporter des preuves irréfutables de leur supériorité par rapport aux techniques automatiques.

Après un certain nombre d'expériences peu concluantes, axées sur l'élicitation des connaissances de l'expert et la mise au point "manuelle" sur un corpus d'exemples (travaux décrits dans mon mémoire de D.E.A. [Munteanu, 1993]), nous nous sommes orientés vers la collaboration entre l'expert et les méthodes d'apprentissage automatique. Ces expériences ont été menées avec le concours de Mme Geneviève Caelen-Haumont, Chargée de Recherches au CNRS, spécialisée dans l'étude de la prosodie [Caelen-Haumont, 1991], [Caelen-Haumont, 1993].

Une première expérience, qui a été réalisée au début de nos travaux sur l'acquisition de connaissances prosodiques, a consisté en la révision par l'expert des règles extraites à partir des arbres de décision produits par ID3 [Munteanu *et al.,* 1995]. A ce moment, nous utilisions directement les noyaux vocaliques détectés automatiquement et un codage moins performant des paramètres prosodiques (qui n'incluait pas, par exemple, les différences de fréquences fondamentales), ce qui explique en partie les résultats inférieurs des algorithmes d'apprentissage : un taux d'environ 75% de bonnes classifications obtenu par ID3 dans la tâche *ML'* (par rapport au taux de 86% réalisé par C4.5 actuellement sur la même tâche). Les règles proposées à l'expert ont cependant été suffisamment intelligibles pour lui permettre une révision efficace, qui a augmenté leurs performances de classification sur cette tâche de 5%. Les questions et les commentaires de l'expert lors de cette première expérience nous ont suggéré les avantages potentiels d'une plus forte implication de sa part dans la phase d'induction proprement dite. Ces réflexions ont abouti à la conception de l'environnement d'induction interactive de règles, COPAC, décrit dans la section 3.2.

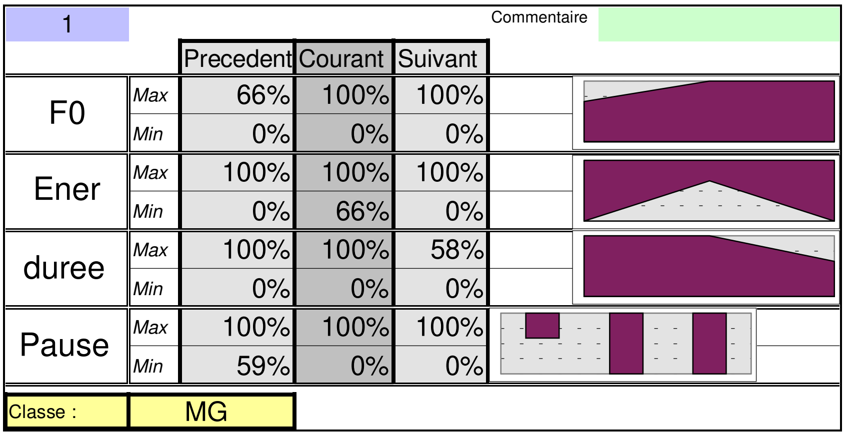



**Figure 4-9** Interface spécialisée de COPAC pour la présentation des règles prosodiques sous forme de gabarits de trajectoires

Dans une deuxième expérience, nous avons donc demandé à l'expert de proposer, à l'aide de COPAC, des règles pour la discrimination des voyelles appartenant aux mots grammaticaux par rapport à ceux appartenant aux mots lexicaux. 1200 exemples du corpus de référence ont été utilisés pour l'induction de règles, le reste de 391 exemples étant réservé pour le test. Dans cette expérience, nous nous sommes intéressés uniquement aux règles qui décrivent les mots grammaticaux (un système expert dédié à cette tâche de classification pourrait également contenir des règles pour les mots lexicaux et une stratégie de résolution des conflits appropriée).

Le paquet de règles ainsi obtenu est composé de **9** règles (Annexe E), simples et intelligibles, qui réalisent un taux de réussite de 74% sur l'ensemble d'exemples de test. Pour la comparaison, il faut mentionner que FOIL [Quinlan, 1990], l'algorithme d'induction de règles qui est à la base de la méthodologie proposée dans COPAC, réalise un taux de bonnes classifications inférieur (71%) à l'aide de **50** règles, beaucoup trop nombreuses pour être comprises et éventuellement améliorées par l'expert.

## 4.2 Etude des transitions formantiques et du bruit d'occlusion pour la caractérisation des occlusives sourdes

Cette section présente les premiers résultats d'une collaboration, plus récente, avec des chercheurs de l'équipe "Synthèse" de l'ICP, qui concerne l'étude des transitions formantiques et du bruit d'occlusion pour la caractérisation des occlusives sourdes. Ces recherches font l'objet d'une thèse en cours [Neagu, 1996]. Leur origine se situe donc dans le domaine de la synthèse de la parole mais elles pourraient également fournir des solutions aux problèmes de décodage acoustico-phonétique des occlusives sourdes.

Les expériences auxquelles nous avons participé visent à évaluer l'apport des deux sources d'information qui sont traditionnellement exploitées pour la discrimination des trois consonnes occlusives sourdes du français (p, t et k) :

- le bruit d'occlusion qui marque l'ouverture du conduit vocal après l'occlusion;
- les transitions des formants vers les voyelles qui suivent ces consonnes.

### 4.2.1 Description du corpus de référence

Le corpus de référence utilisé dans nos expériences se compose de 138 éléments de signal de parole correspondant à toutes les combinaisons possibles $V_1CV_2$ (120) et $V_2CV_2$ (18), où $V_1 \in \{a, i, u, y\}$ (les voyelles cardinales), $C \in \{p, t, k\}$ (les occlusives sourdes) et $V_2 \in \{a, i, u, y, ø, œ, e, ɛ, o, O\}$ (les voyelles orales). Ces éléments de signal ont été soumis à deux chaînes de traitement différentes en vue de l'obtention de paramètres synthétiques pour la caractérisation du bruit d'occlusion et des transitions formantiques.

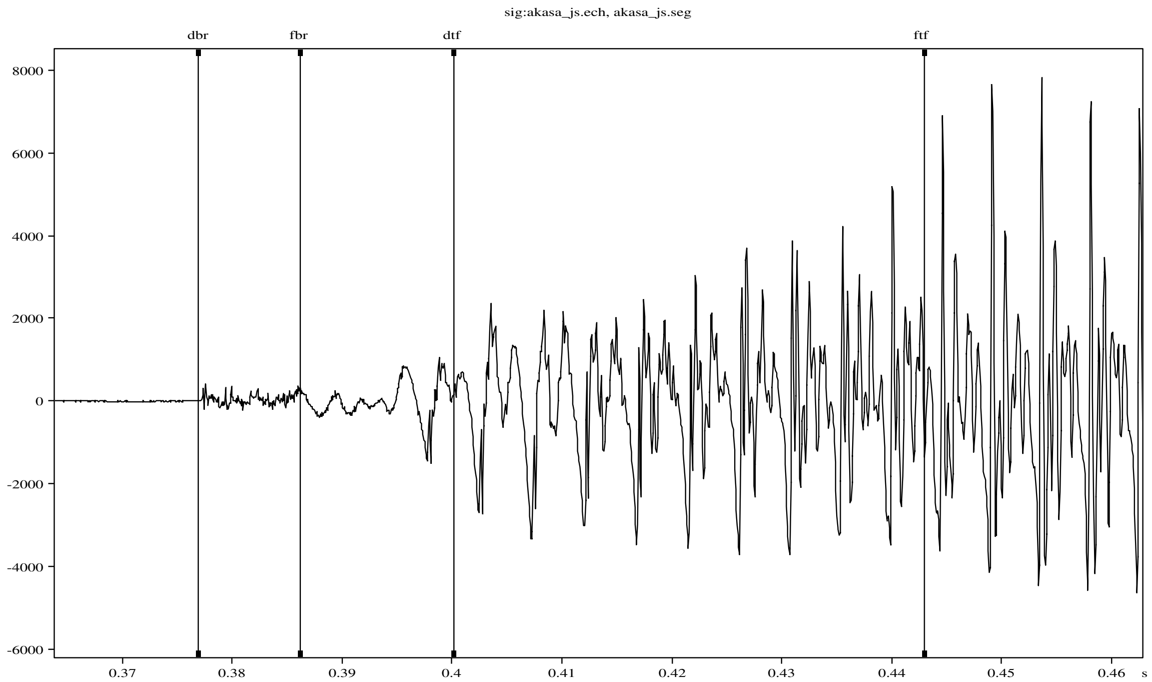



**Figure 4-10** Localisation temporelle du bruit d'occlusion et de la transition sur un élément de signal de parole (/akasa/). Les étiquettes *dbr*, *fbr*, *dtf* et *ftf* délimitent les portions de signal correspondant à ces deux phénomènes (*br* = burst, *tf* = transition formantique) pour l'occlusive k.

Les limites du bruit d'occlusion ont été identifiées par expertise et le spectre correspondant a été calculé avec un fenêtrage de Hanning asymétrique, d'une longueur de 10 ms, centré sur le moment du relâchement. Ce spectre a été ensuite lissé par une LPC (*Linear Prediction Coding*) d'ordre 12. Trois procédures de codages différentes ont été utilisées dans nos expériences :

- codage linéaire en 10 échantillons (*lin_10*);
- codage en Barks (une transformation de nature logarithmique qui présente une certaine plausibilité perceptive), 10 échantillons (*brk_10*);
- codage par la transformée en cosinus discret (qui devrait réaliser un maximum de compression de l'information), 7 échantillons (*dct_7*).

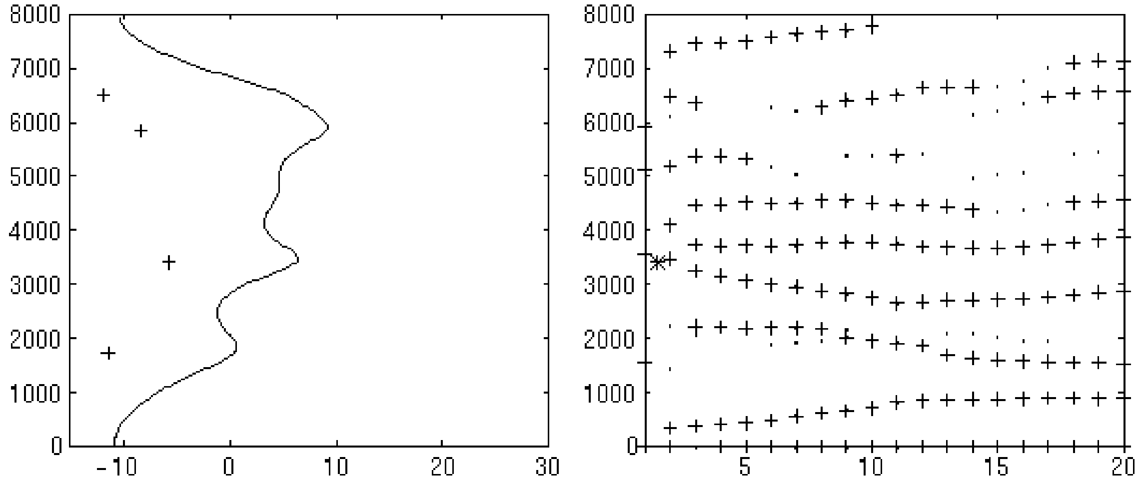



**Figure 4-11** Spectre brut du bruit d'occlusion et formants au cours de la transition pour le signal /aka/

La caractérisation des transitions formantiques a été fondée sur une procédure semi-automatisée qui a eu comme objectif l'identification des points de mesure correspondant au centre et à la périphérie (le dernier endroit pour lequel on peut encore calculer les formants) de la voyelle qui suit la consonne étudiée. Dans nos expériences nous avons évalué deux codages différents des transitions :

- $F_{1v}$, $F_{2v}$, $F_{3v}$, $F_{1v}$-$F_{1c}$, $F_{2v}$-$F_{2c}$, $A_{2c}$-$A_{3c}$ (*cv_diff*);
- $F_{1v}$, $F_{2v}$, $F_{3v}$, $F_{1c}$, $F_{2c}$, $F_{3c}$, $A_{1v}$, $A_{2v}$, $A_{3v}$, $A_{1c}$, $A_{2c}$, $A_{3c}$ (*cv_all*)

où F et A dénotent les fréquences et les amplitudes des formants, le premier niveau d'indices (1,2,3) représente l'ordre des formants et le deuxième niveau d'indices indique le point de mesure : v = au centre de la voyelle, c = à la périphérie de la voyelle.

Enfin, nous avons également évalué des représentations qui regroupent le codage *cv_diff* des transitions et chacun des codages utilisés pour le bruit d'occlusion : *fusion_lin*, *fusion_brk* et *fusion_dct*.

Il faut préciser que dans ces expériences nous avons pu utiliser des éléments de signal prononcés par deux locuteurs (276 exemples) pour lesquels le codage du bruit d'occlusion a été calculé, alors que pour l'évaluation des représentations qui utilisent les transitions formantiques nous ne disposions que des éléments de signal traités correspondant à un seul locuteur (138 exemples).

### 4.2.2 Induction automatique de classificateurs

Le problème abordé revient à l'étude des classificateurs pour les trois catégories d'occlusives sourdes (p, t et k), produits automatiquement en utilisant différentes représentations des exemples.

Nous allons présenter les résultats obtenus avec les six méthodes de classification automatique utilisées pour l'induction de connaissances prosodiques : PmBc, IDF.C, C4.5, CN2, NOPT et CGQ (le classificateur gaussien quadratique obtient sur ces tâches de meilleurs résultats que CGL). Compte tenu du nombre réduit d'exemples, nous avons restreint le nombre d'apprentissages/tests à 20. Les moyennes et les écarts types des taux d'erreur obtenus par ces méthodes de classification automatique (sur les 8 tâches de classification correspondant aux différents paramètres et procédures de codage décrites dans la section précédente) sont présentés dans la figure 4-12.

| Tâche | PmBc | | IDF.C | | C45 | | CN2 | | NOPT | | CGQ | | CGL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± | Moy | ± |
| *brk_10* | 16,87 | 5,07 | 22,68 | 5,28 | 23,30 | 5,99 | 25,44 | 7,88 | 15,53 | 3,71 | 20,62 | 4,39 | 22,05 | 4,72 |
| *dct_7* | 18,04 | 4,41 | 25,71 | 6,64 | 20,61 | 3,44 | 32,23 | 4,90 | 19,11 | 7,05 | 14,47 | 3,12 | 26,97 | 6,07 |
| *lin_10* | 18,39 | 5,09 | 24,91 | 7,15 | 19,84 | 7,20 | 30,54 | 5,56 | 15,36 | 5,66 | 16,16 | 4,62 | 22,41 | 6,52 |
| *cv_all* | 13,39 | 5,66 | 20,00 | 5,95 | 25,02 | 7,57 | 30,19 | 9,24 | 18,57 | 9,24 | 12,32 | 2,72 | 28,93 | 6,44 |
| *cv_diff* | 11,43 | 4,11 | 16,96 | 8,27 | 24,31 | 7,56 | 33,19 | 8,45 | 7,86 | 6,82 | 7,86 | 5,51 | 33,93 | 11,85 |
| *fusion_brk* | 11,25 | 2,90 | 15,00 | 4,43 | 19,47 | 3,91 | 19,12 | 3,11 | 3,21 | 3,26 | 6,96 | 4,56 | 14,82 | 7,17 |
| *fusion_dct* | 15,71 | 8,06 | 15,89 | 4,84 | 17,33 | 4,53 | 14,81 | 7,37 | 3,03 | 3,53 | 2,68 | 2,28 | 9,46 | 3,72 |
| *fusion_lin* | 16,25 | 9,16 | 17,14 | 7,99 | 15,88 | 6,30 | 22,86 | 9,80 | 5,36 | 4,10 | 2,14 | 3,36 | 7,86 | 7,47 |

**Figure 4-12** Résultats (taux d'erreur) des différents algorithmes

En utilisant la méthodologie décrite dans la section 4.1.5 on obtient le classement suivant des différents algorithmes utilisés :

| Tâche | Algorithmes | (Légende : | **1er groupe** | 2ème groupe | 3ème groupe | 4ème groupe | **5ème groupe** |
|---|---|---|---|---|---|---|---|
| *brk_10* | **NOPT** | **PmBc** | CGQ | CGL | IDF.C | C45 | CN2 |
| *dct_7* | **CGQ** | PmBc | NOPT | C45 | IDF.C | CGL | **CN2** |
| *lin_10* | **NOPT** | **CGQ** | PmBc | C45 | CGL | IDF.C | **CN2** |
| *cv_all* | **CGQ** | **PmBc** | NOPT | IDF.C | C45 | CGL | CN2 |
| *cv_diff* | **NOPT** | **CGQ** | PmBc | IDF.C | C45 | **CN2** | **CGL** |
| *fusion_brk* | **NOPT** | CGQ | PmBc | CGL | IDF.C | **CN2** | **C45** |
| *fusion_dct* | **CGQ** | **NOPT** | CGL | CN2 | PmBc | IDF.C | C45 |
| *fusion_lin* | **CGQ** | NOPT | CGL | C45 | PmBc | IDF.C | CN2 |

| | CGQ (I) | NOPT (II) | PmBc (III) | IDF.C (IV) | CGL (V) | C4.5 (VI) | CN2 (VII) |
|---|---|---|---|---|---|---|---|
| **1er groupe** | **6** | **5** | **2** | | | | |
| 2ème groupe | 2 | 3 | 3 | 1 | 2 | 1 | |
| 3ème groupe | | | 3 | 4 | 4 | 5 | 2 |
| 4ème groupe | | | | 3 | 1 | 1 | 2 |
| **5ème groupe** | | | | | **1** | **1** | **4** |
| MTR(/PmBc) | 0,68 | 0,71 | 1,00 | 1,31 | 1,44 | 1,43 | 1,77 |

**Figure 4-13** Comparaison des algorithmes d'apprentissage

A la différence des tâches d'étiquetage prosodique, présentées dans la section 4.1, cette comparaison met en évidence un net avantage du classificateur gaussien quadratique et du réseau de neurones pour la discrimination des occlusives sourdes. En effet, ces méthodes se placent dans les deux premiers groupes pour toutes les tâches étudiées. Nous considérons que les résultats des quatre dernières tâches indiquent des distributions de classes très proches de trois lois gaussiennes à matrices de covariance différentes, ce qui génère des frontières optimales difficilement approximables par les représentations hyperrectangulaires propres aux algorithmes d'apprentissage symbolique utilisés.

Parmi ces derniers, PmBc (dont la stratégie de classification permet une meilleure approximation des frontières non-orthogonales) s'avère encore le plus performant (mais avec environ 32% de taux d'erreur en plus par rapport à CGQ) en réussissant même à dépasser chacune des deux premières méthodes de classification sur au moins une des tâches étudiées (il dépasse CGQ sur *brk_10* et NOPT sur *dct_7* et *cv_all*). Les résultats de IDF.C, de CGL et de C4.5 sont très proches, mais ces expériences confirment encore une fois le léger avantage de IDF.C par rapport à C4.5. Enfin, CN2 semble l'algorithme d'apprentissage le moins performant pour ces tâches.

Du point de vue des recherches sur la caractérisation des occlusives sourdes, ces résultats permettent la formulation d'un certain nombre de conclusions provisoires :

- Les deux caractérisations des occlusives sourdes (bruit d'occlusion et transitions formantiques) apportent des informations pertinentes pour la discrimination des consonnes p, t et k.

- Le choix du meilleur codage du spectre du bruit d'occlusion dépend de la méthode de classification utilisée. Toutes les méthodes semblent s'accorder sur une utilisation du codage *cv_diff* pour les transitions formantiques.

- Les transitions formantiques semblent plus informatives que le bruit d'occlusion (cette affirmation doit être pondérée par l'apport beaucoup plus important de l'expertise humaine dans la procédure utilisée pour l'obtention des paramètres correspondant aux transitions)

- L'utilisation conjointe des deux sources d'information dans les expériences de "fusion" apporte un gain de performances par rapport à leur exploitation indépendante pour la plupart des méthodes de classification.

Cette étude est encore loin d'être finie. Des travaux sont en cours pour l'élargissement du corpus de référence à d'autres locuteurs. Il est également très important d'étudier les prédictions incorrectes des classificateurs pour typer les erreurs et évaluer l'influence du contexte vocalique sur les différentes représentations des occlusives. Nous pensons que, malgré leurs performances de classification inférieures à d'autres méthodes sur ces tâches, les algorithmes d'apprentissage symbolique peuvent être utilisés avec intérêt pour cette direction de recherche, car ils produisent des représentations facilement interprétables.

### 4.2.3 Perspectives en coopération expert-machine

Certains résultats semblent déjà confirmer l'intérêt de la coopération expert-machine dans la réalisation de classificateurs efficaces qui reposent sur des connaissances explicites et facilement interprétables.

Au début de cette étude sur les occlusives sourdes, nous avons utilisé FFOIL [Quinlan, 1995] (une version de FOIL mieux adaptée pour les tâches de classification) pour l'induction de règles ordonnées (représentées sous forme de clauses Prolog utilisant l'opérateur *cut*) à partir des spectres des bruits d'occlusion codés en Barks. A ce moment nous ne disposions que des exemples correspondant à un seul locuteur. Pour notre application, FFOIL présentait l'intérêt de pouvoir inclure dans le corps des règles des tests de comparaison relative ($\leq$ ou >) entre les valeurs des différents attributs, ce qui rendait les représentations obtenues plus facilement interprétables par les chercheurs.

La discrimination des consonnes p, t et k à partir du spectre du bruit d'occlusion représente essentiellement une tâche de reconnaissance de formes. Vu la diversité impressionnante des formes de ces spectres il s'agit de mettre en évidence un nombre d’événements (occurrence d'un maximum ou d'un minimum sur la courbe du spectre, par exemple) qui permettraient la classification.

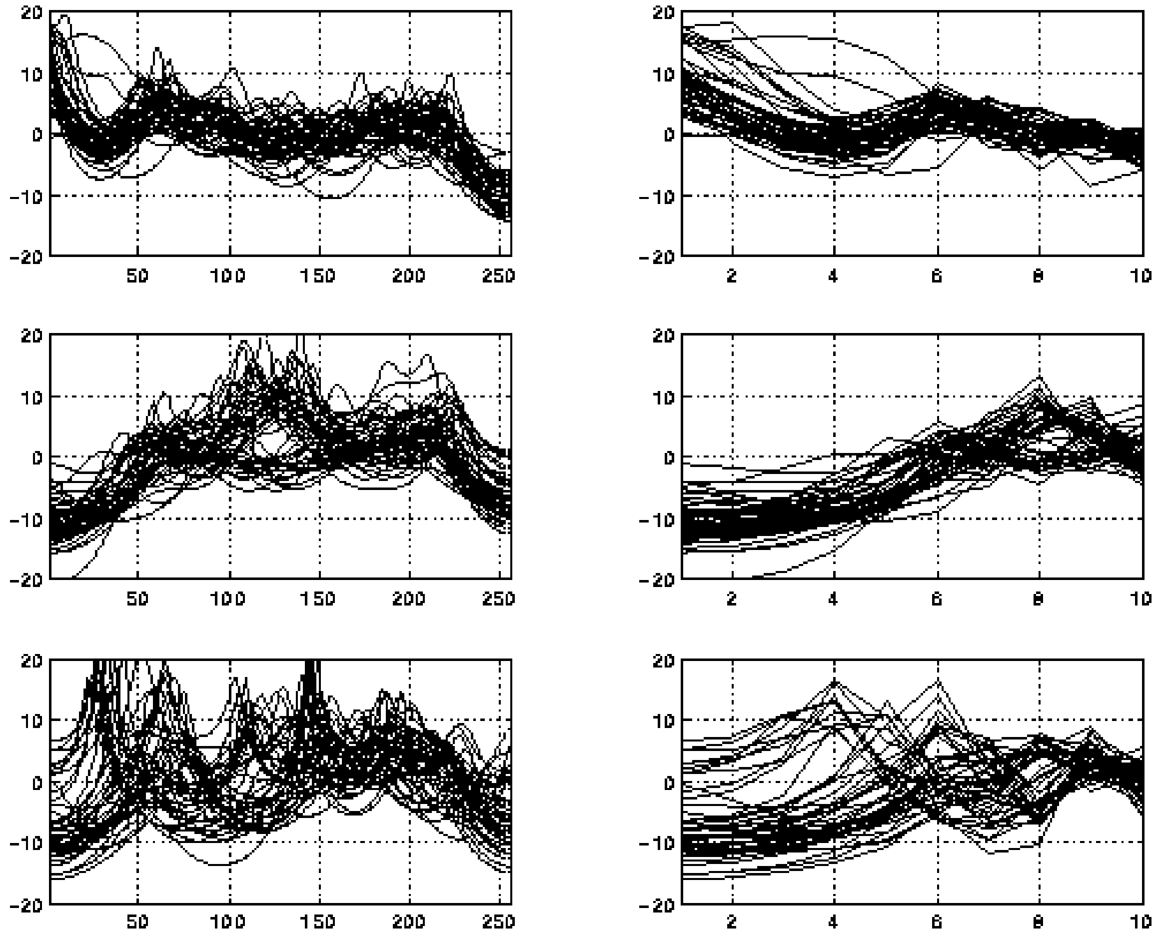



**Figure 4-14** Spectres bruts et codés en Barks correspondant aux trois classes d'occlusives sourdes (dans l'ordre : p, t et k)

Les règles produites par FFOIL ont été révisées par le chercheur spécialisé dans cette problématique (Adrian Neagu) qui s'est déclaré intéressé par quelques-unes des propositions de l'algorithme mais en désaccord avec d'autres et a effectué les modifications correspondantes. Il a également réduit le nombre de règles de 7 à 6. D'après l'expert, les règles proposées par FFOIL lui ont surtout été utiles pour la mise en évidence des événements porteurs d'information sur la courbe spectrale et les modifications qu'il a proposées ont eu pour but de rendre plus robuste leur détection. Suite à ces modifications, le taux d'erreur réalisé par le paquet de règles sur l'ensemble d'exemples d'apprentissage a été augmenté légèrement (de 7,07% à 7,25%) mais l'expert a considéré qu'il s'agissait, dans la plupart de ces erreurs, d'exemples aberrants qui devaient être considérés comme du bruit et donc ignorés.

La disponibilité récente d'un nouveau corpus d'exemples correspondant à un autre locuteur nous a permis d'évaluer la qualité des deux paquets de règles (celui proposé par FFOIL et celui révisé par l'expert) sur un ensemble de test indépendant. Le paquet de règles originales est affecté par une chute brutale de performances sur ces nouveaux exemples (**31,88%** d'erreurs). Au contraire, les règles révisées réalisent un résultat supérieur aux moyennes de la plupart des algorithmes automatiques utilisés dans les expériences présentées dans la section précédente : seulement **16,67%** d'erreurs. En plus, il faut noter que les méthodes automatiques ont eu accès à plus d'exemples d'apprentissage, correspondant aux deux locuteurs. Les deux paquets de règles sont présentés dans l'Annexe F.

## 4.3 Conclusion

L'expérimentation des techniques d'apprentissage symbolique dans deux problèmes concrets de recherche en parole renforce les observations présentées dans le chapitre 3. Nous retrouvons dans ces deux applications l'importance des attributs numériques, la nécessité de gérer le bruit causé par les imperfections des pré-traitements et l'apport des connaissances scientifiques et des connaissances des experts dans l'amélioration des résultats de l'apprentissage.

Les résultats des trois algorithmes d'apprentissage que nous avons élaborés (PmBc, IDF.C et COPAC) sont similaires à ceux obtenus sur des tâches de classification issues de domaines différents. Les bonnes performances de PmBc par rapport à d'autres méthodes d'apprentissage symbolique sont confirmées dans les deux applications. IDF.C reste très proche de C4.5 mais conserve son léger avantage. Les résultats obtenus à l'aide de COPAC semblent également prometteurs.

Dans leur ensemble, les performances des algorithmes d'apprentissage symbolique par rapport à d'autres méthodes de classification automatique, plus utilisées actuellement dans le domaine de la parole, varient beaucoup entre les deux applications. Pour l'acquisition de connaissances prosodiques les algorithmes symboliques réalisent des résultats comparables ou même supérieurs (PmBc) à ceux du réseau de neurones et surpassent les deux classificateurs gaussiens. Par contre, ils sont beaucoup moins performants que NOPT et CGQ dans l'utilisation de certains codages pour la discrimination des occlusives sourdes. Sur ces tâches, où les distributions des classes sont en accord évident avec ses hypothèses statistiques, le classificateur gaussien quadratique s'avère très performant (il l'est beaucoup moins quand ses hypothèses ne sont pas confirmées, comme c'est le cas dans les tâches prosodiques). Le réseau de neurones s'est montré le plus robuste face aux différences des distributions mais son pouvoir de discrimination supérieur est réalisé au prix du réglage laborieux de ses paramètres, de sa lenteur et du manque d'intelligibilité de ses représentations internes.

Les différentes catégories de méthodes de classification automatique présentent donc des avantages et des inconvénients différents mais souvent complémentaires. Nous sommes convaincus que les algorithmes d'apprentissage symbolique peuvent apporter une contribution considérable dans le travail de recherche en parole, surtout dans le travail d'exploration et d'expérimentation, par leur efficacité d'exécution et par l'intelligibilité directe de leurs résultats.

# Chapitre 5 :
# INTEGRATION DES TECHNIQUES D'EXTRACTION DE CONNAISSANCES DANS LE SIDOC

Nous avons présenté, dans l'introduction de cette thèse, les motivations qui sont à la base de nos recherches sur l'applicabilité des méthodes d'apprentissage automatique symbolique en parole. Ces recherches ont émergé dans le cadre du projet SIDOC et doivent aboutir à l'intégration des techniques d'extraction de connaissances dans cet environnement. Nous allons présenter dans ce chapitre le projet SIDOC et nos contributions à l'évolution de ce projet.

## 5.1 Le projet SIDOC

Une des principales difficultés que rencontrent les chercheurs dans le domaine de la reconnaissance et de la synthèse de parole est l'immense variabilité, inter- et même intra-locuteur, du signal de la parole. D'où la nécessité de disposer de larges corpus de parole enregistrés par de nombreux locuteurs, ce qui a poussé, depuis une douzaine d'années, les divers laboratoires ou organismes de recherche à fournir un effort considérable pour l'enregistrement de larges bases de données. Ces bases de données se sont avérées d'une grande importance pour le progrès général des recherches en parole et, particulièrement, pour l'évaluation et la mise au point des systèmes à entrées / sorties vocales et pour les études du langage.

Ces activités ont démarré aux Etats-Unis, par la production des bases comme TI-DIGITS [Leonard, 1984]ou TIMIT [Price *et al.,* 1988] qui sont encore aujourd'hui largement utilisées pour l'évaluation des systèmes de reconnaissance de la parole. Plus récemment, des initiatives européennes et plus particulièrement des projets ESPRIT (*European Strategic Programme for Research and Developpement in Information Technology*) ont abouti à l'enregistrement de bases de données multi-lingues comme EUROM0 [Fourcin *et al.,* 1989] (huit langues) ou EUROM1 [Zeiliger *et al.,* 1992] (onze langues). De nombreuses bases de données francophones ont également été produites : BDSONS [Descout *et al.,* 1986], BREF [Lamel *et al.,* 1991], BD-BRUIT [Zeiliger *et al.,* 1994], etc. Notre laboratoire a participé activement à l'enregistrement de bases de données françaises et européennes (EUROM1, BDSONS, BD-BRUIT) et à la définition de standards pour les méthodologies et les outils d'évaluation des systèmes à entrées / sorties vocales [Fourcin & Dolmazon, 1991].

SIDOC (*Système d'Intégration de DOnnées et de Connaissances en Parole*) est un projet à long terme de notre laboratoire, visant au développement d'une plate-forme intégrée qui faciliterait l'utilisation, d'une manière systématique, de ces bases de données pour la recherche en parole. Les objectifs principaux de ce projet sont :

- la modélisation uniforme des entités hétérogènes manipulées par les chercheurs en parole [Cervantes, 88], [Fernandez, 89];
- le stockage et la gestion efficaces des grandes quantités de données parole [Cervantes, 88];
- le développement d'interfaces ergonomiques pour la présentation, l'exploration et la récupération des données [Belbachir, 1991];
- l'accès transparent pour l'utilisateur à des programmes de traitement du signal variés [Neagu, 1994];
- **le développement d'outils d'acquisition, test et mise au point de connaissances assistés par ordinateur** [Fernandez, 89], [Zeiliger, 1992], [Munteanu, 1993];
- le contrôle de l'intégrité syntaxique et sémantique des données [Belbachir, 1991].

L'information traitée par les chercheurs en parole est caractérisée par une grande hétérogénéité, une grande complexité et des fortes interconnexions (les sections suivantes présenteront une analyse plus détaillée de ces aspects). En outre, les données parole sont manipulées d'une manière différente en fonction de leur type : les signaux acoustiques sont écoutés, les signaux vidéo sont visualisés, les programmes de traitement du signal sont exécutés, etc. Les concepteurs du SIDOC ont considéré, dès le début de ce projet, que la possibilité d'associer structure et comportement, dans la représentation de ces entités, faciliterait la conception du système et se sont donc dirigés vers les approches orientées objets.

Aux débuts du projet SIDOC (fin des années quatre-vingt), le concept de Système de Gestion de Bases de Données Orientées Objets (SGBDOO) venait d'apparaître et la plupart des réalisations informatiques de SGBDOO étaient encore des prototypes de recherche. Les premières versions du SIDOC ont donc été fondées sur une implémentation originale du modèle objet au dessus d'un gestionnaire relationnel de bases de données, GURU[1], qui apportait la sécurité d'un système commercial puissant, bénéficiant de développements et de mises à jour périodiques.

Depuis, la situation a beaucoup changé : des SGBDOO commerciaux sophistiqués, disponibles sur des plates-formes variées et validés dans de nombreuses applications sont actuellement disponibles. Dans les conclusions de mon mémoire de D.E.A. [Munteanu, 1993], j'ai souligné les avantages que le projet SIDOC pouvait tirer de ces avancées et j'ai proposé l'utilisation d'un véritable gestionnaire de bases de données orientées objets pour la suite de ces recherches. Ces propositions ont été concrétisées, en début de ma thèse, par le choix du SGBDOO GemStone[2] pour le développement d'une nouvelle version du SIDOC.

Les progrès des SGBDOO ont été également accompagnés par une certaine stabilisation des concepts de base de "l'orienté objets" et par la consécration des méthodologies d'analyse et de conception appropriées [Booch, 1991], [Coad & Yourdon, 1991], [Rumbaugh *et al.,* 1991]. Nous avons donc considéré que le passage du SIDOC à un nouvel environnement de développement constituait aussi l'occasion de reconsidérer certains aspects de la modélisation des objets parole.

## 5.2 Systèmes de gestion de bases de données orientées objets

Le concept d'**objet** constitue un point de rencontre remarquable entre des idées issues indépendamment de plusieurs disciplines informatiques (langages de programmation, interfaces, I.A., bases de données). Ces origines multiples impriment encore certaines particularités aux différentes acceptions de cette notion (l'utilité de l'encapsulation est, par exemple, remise en question par des approches issues de l'I.A. ou des bases de données), mais

---

[1] GURU est une marque déposée de Micro Data Base Systems Inc.

[2] GemStone est une marque déposée de Servio Corporation

la plupart des définitions actuelles semblent s'accorder sur un grand nombre de caractéristiques communes (identité, classes et héritage, par exemple).

Dans le domaine des bases de données, le règne absolu des modèles relationnels (qui ont à leur tour succédé aux systèmes réseaux ou hiérarchiques des années 60-70) semble aujourd'hui menacé par une nouvelle génération de SGBD, conçus pour répondre aux exigences des applications de la "prochaine génération" : conception et ingénierie logicielle assistées par ordinateur, systèmes d'information multimédia et hypermedia, bases de données intelligentes. Le développement de la technologie objet dans le domaine des bases de données a été marqué par des débats, parfois très animés, entre les avocats d'une extension plus "douce" du modèle relationnel [Stonebraker *et al.,* 1990] et les défenseurs d'une approche fondamentalement orientée objets [Atkinson *et al.,* 1989]. Des produits issus des deux écoles ont finalement pénétré le marché des bases de données et on assiste aujourd'hui à une évolution croissante du nombre et de la variété des applications qui les choisissent comme support de développement.

Pour essayer de caractériser les SGBDOO, nous allons nous rapporter aux treize "règles d'or", proposées dans [Adiba & Collet, 1993], que doivent respecter les gestionnaires de bases de données pour revendiquer ce "label de qualité". Ces règles combinent des exigences d'ordre général pour les SGBD (possibilité de gestion de **grandes quantités** de données **persistantes**, en assurant leur **fiabilité** et en facilitant leur **partage** et leur **interrogation**) avec des desiderata propres aux systèmes à objets (support pour l'**encapsulation**, les **objets composites**, les **identificateurs** d'objets, les **classes,** les **types** et les **méthodes**, l'**héritage**, la **surcharge** des méthodes, un **langage de programmation complet** et l'**extensibilité**).

Pour choisir le futur environnement de développement pour le SIDOC, nous nous sommes retrouvés en 1993 devant une gamme de SGBDOO déjà extrêmement diversifiée : GemStone, $O_2$, Ontos, ObjectStore, Objectivity/DB, UniSQL/X, Versant et Itasca pour ne citer que quelques-uns des produits commerciaux les plus connus et sans parler des prototypes de recherche disponibles dans le domaine public. Compte tenu des particularités et des objectifs de recherche du projet SIDOC, nous nous sommes orientés vers le SGBD GemStone, qui réunissait, selon nous, le plus grand nombre des caractéristiques souhaitées.

Apparu sur le marché en 1987, GemStone de Servio Corporation est le plus ancien des SGBDOO commerciaux. Il a été conçu pour une exploitation multi-utilisateurs et multi-plates-formes dans un environnement client-serveur. Les atouts principaux de ce système sont :

- base de données active : GemStone permet aux développeurs d'associer aux classes des méthodes qui sont stockées et exécutées directement dans la base de données. Ces méthodes sont accessibles soit en interne soit par des applications clients externes.

- support concurrent pour plusieurs langages de programmation : Smalltalk, C++ et C. Toutes les applications développées dans ces langages ont simultanément accès aux objets de la base de données.

- contrôle des transactions, flexible et multi-utilisateurs.

- sécurité des informations : des droits d'accès différents peuvent être assignés à n'importe quel objet de la base.

- support pour les modifications dynamiques du schéma et pour l'évolution des objets : versions des classes, migration automatique des instances, garantie de l'intégrité référentielle.

Servio Corporation diffuse aussi GeODE (*GemStone Object Development Environment*), un environnement ergonomique de développement d'applications pour GemStone. Il est composé de modules pour la conception et la représentation graphiques des schémas (*Visual Schema Designer*), pour le développement rapide des formulaires de présentation des données (*Visual Forms Designer*), pour la programmation graphique des interfaces (*Visual Program Designer*) et de différents outils d'inspection et de navigation, d'organisation des classes, de déverminage et d'édition d'image. Pour le développement du SIDOC, il présente l'immense avantage de rendre réutilisables par les programmeurs toutes les composantes de cet environnement (et beaucoup d'autres) dans une bibliothèque de plus de 480 classes et de milliers de méthodes.

### 5.3 Modélisation objet des données-parole

Dans cette section, nous allons présenter succinctement la démarche que nous avons suivie dans la modélisation objet des données-parole gérées par le SIDOC. Notre tâche a consisté essentiellement dans la structuration globale et de haut niveau du schéma du SIDOC, les aspects particuliers correspondant à des domaines d'étude spécifiques, comme l'articulatoire, étant abordés dans d'autres recherches effectuées dans notre laboratoire [Bensaber et *al.,* 1996]. Pour cette activité de modélisation nous avons adopté la méthodologie OMT (*Object*

*Modeling Technique*) [Rumbaugh *et al.*, 1991]. Tous les schémas présentées dans les paragraphes suivants suivent la notation graphique proposée dans cette méthodologie.

### 5.3.1 Les données parole

Cette section présente quelques particularités des données utilisées dans la recherche en parole qui imposent des contraintes à la modélisation et à la gestion. Une analyse plus complète des problèmes généraux liés au développement des bases de données scientifiques est réalisée dans [French *et al.,* 1990].

Les données gérées par les bases de données scientifiques peuvent être classées en fonction de leur origine et de leur niveau d'abstraction. Les différentes catégories de données demandent des solutions différentes de modélisation et de gestion.

La partie la plus importante des données gérées par le SIDOC provient des bases de données parole déjà existantes. Nous allons classer ces *données primaires* en *données brutes* (signaux) et *données descriptives* (informations sur les caractéristiques des signaux et sur le protocole d'enregistrement).

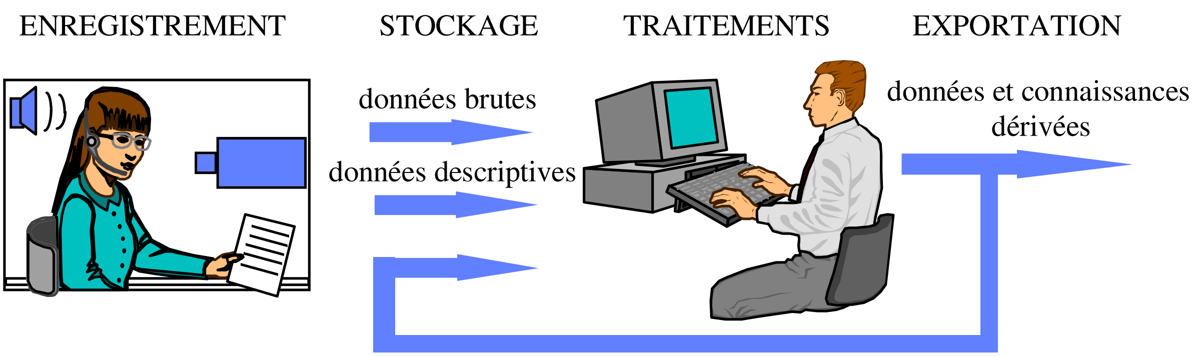



**Figure 5-1** Cycle de vie des données dans le SIDOC

Les bases de données parole qui alimentent le SIDOC avec des données primaires ont été enregistrées à des moments différents, par des organismes différents, pour des raisons différentes. La variété de leur origine implique des différences importantes dans leur représentation.

Nous pouvons observer des structures communes dans ces données : toutes les sources fournissent certains types de signaux physiques, des informations sur les locuteurs, les corpus linguistiques et les conditions d'enregistrement. A un niveau d'analyse plus fin, la diversité devient, cependant, apparente. Les descriptions des locuteurs (âge, poids, sexe, caractéristiques linguistiques, etc.) et des conditions d'enregistrement ne sont presque jamais les mêmes. Le nombre, le type et la signification des signaux enregistrés varient aussi entre les différentes sources (signaux audio en provenance d'un ou plusieurs microphones, images vidéo, films radiologiques, signaux laryngographiques, etc.).

Une autre source de données est l'activité de recherche. Les données brutes sont traitées par les experts et les outils d'analyse pour produire des caractérisations plus élaborées - *des données et connaissances dérivées* - à différents niveaux d'abstraction : acoustique, phonétique, prosodique, syntaxique, sémantique et pragmatique. Les données dérivées sont souvent stockées à leur tour dans les bases de données pour deux raisons principales : éviter la répétition des traitements, d'habitude coûteux, impliqués dans leur production et les rendre disponibles pour les autres chercheurs de la communauté scientifique.

Même si les données primaires sont caractérisées par des variations importantes dans leur représentation, leur contenu reste stable et objectif. Les données et les connaissances dérivées qui sont élaborées dans l'activité de recherche évoluent et, souvent, ne rallient pas le consensus à l'intérieur de l'entière communauté scientifique.

La coexistence des points de vue multiples et parfois divergents est reflétée dans la production des résultats différents et même contradictoires (pour se convaincre de cette réalité il suffit de regarder deux étiquetages proposés par deux experts pour une même portion de signal de parole). Dans le SIDOC nous devons donc gérer des estimations différentes d'un même concept et fournir les moyens pour une identification correcte de leurs auteurs et des données originales qui ont été traitées.

Par ailleurs, les résultats scientifiques évoluent continuellement. Un système de gestion approprié pour ce type de bases de données doit s'adapter à cette évolution tout en garantissant la préservation de la cohérence syntaxique et sémantique de leur contenu. La gestion de cette dynamicité est rendue encore plus difficile par l'existence des réseaux complexes d'interdépendances entre les résultats partiels obtenus aux différents niveaux de l'analyse.

### 5.3.2 Modélisation et évolutivité

La variété structurelle et l'évolutivité des données scientifiques ont été prises en compte dès les premières étapes de modélisation. Nous distinguons deux types d'évolutions auxquelles peuvent être soumises les données parole : l'évolution des classes (changements dans la structure des classes existantes, création de nouvelles classes, etc.) et l'évolution des objets (reconsidération des valeurs de certains objets, destruction d'objets, etc.). SIDOC doit gérer ces changements en préservant la cohérence globale des bases de données.

Certains mécanismes pour le contrôle de l'intégrité syntaxique des bases de données sont directement fournis par GemStone (versions de classes et migration d'instances, maintenance de l'intégrité référentielle par la préservation indéfinie des objets référencés). D'autres types de problèmes de maintien de la cohérence, posés par l'existence des dépendances sémantiques entre les données dérivées, les agents de traitements et les données traitées, doivent être résolus par le SIDOC lui-même.

Nous considérons, cependant, que la meilleure solution pour limiter les problèmes posés par l'évolution structurelle consiste dans une modélisation initiale appropriée. Les versions des classes offrent des moyens puissants pour gérer des changements arbitraires dans la structure des classes, mais la migration des instances correspondantes est une opération lente et délicate. La spécialisation constitue certainement un mécanisme plus simple et plus efficace pour l'extension du schéma. Un modèle initial qui anticipe les éventuels besoins de changement, en fournissant des classes abstraites appropriées pour des futures spécialisations, limiterait donc l'appel à des techniques plus sophistiquées mais plus coûteuses et faciliterait la maintenance du système.

### 5.3.3 Modélisation des données primaires

Les données brutes représentent la trace des locutions des différents locuteurs, placés dans des conditions particulières et qui ont prononcé un énoncé spécifié dans un corpus linguistique, pendant une séance d'enregistrement. Dans le jargon parole, on dit que ces signaux décrivent une réalisation particulière d'un corpus linguistique. Ce concept est modélisé par la classe *Réalisation*, qui permet d'établir la correspondance entre les signaux représentant les différents aspects d'une réalisation. Elle garde aussi l'ensemble des informations globales qui décrivent le protocole expérimental (*Locuteur*, *Corpus* et *ConditionsEnregistrement*).

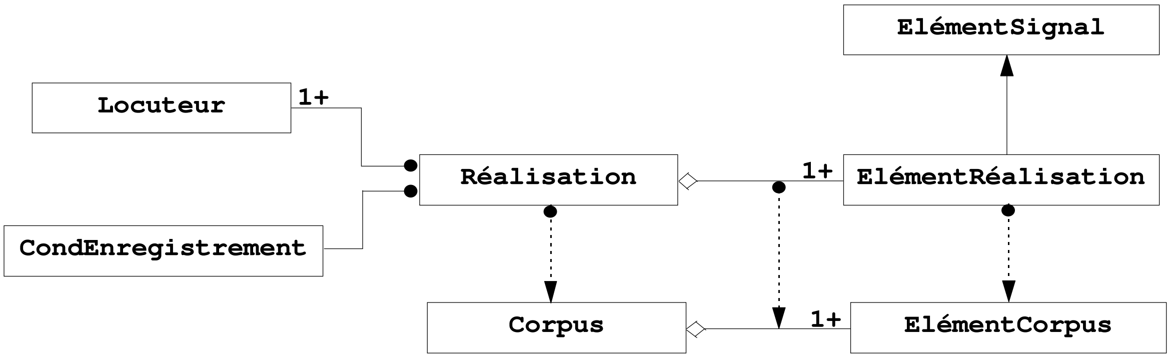



**Figure 5-2** Modélisation des données primaires (1)

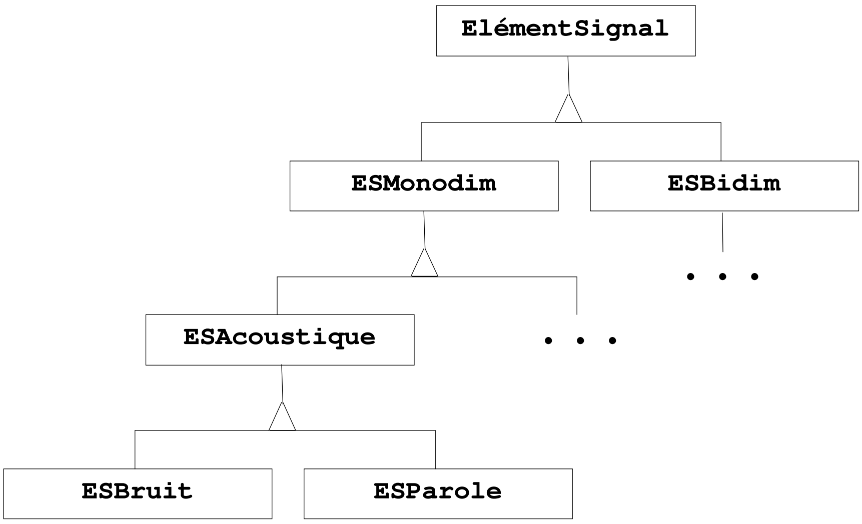



**Figure 5-3** Modélisation des données primaires (2)

Les différentes catégories de signaux sont modélisées par la hiérarchie de sous-classes de la classe *ElémentSignal* (la figure 5-3 présente une portion de ce graphe). Cette taxonomie factorise les similarités structurelles et fonctionnelles des signaux dans les classes abstraites (*ESMonodim*, *ESBidim*, *ESAcoustique*, etc.), en haut de la hiérarchie. Les catégories concrètes de signaux manipulés dans la recherche en parole sont représentées au niveau des feuilles terminales (*ElémentSignalParole*, *ElémentSignalBruit*, etc.).

### 5.3.4 Modélisation des données dérivées

Au plus haut niveau du schéma, les données dérivées sont modélisées dans le SIDOC par la classe *ObjetDérivé* et ses sous-classes : *Paramètre*, qui décrit les estimations périodiques d'une valeur continue au long d'un élément de signal, et *Etiquette*, qui décrit les occurrences discrètes d'événements à des endroits particuliers dans le signal. Ces classes abstraites sont ensuite spécialisées davantage pour fournir la structure et le comportement particulier des étiquettes (phonétiques, prosodiques, articulatoires, etc.) ou des paramètres (fréquence fondamentale, énergie, spectre, etc.) concrets manipulés par les chercheurs en parole.

A part les valeurs effectives des données dérivées, ces classes fournissent des structures pour garder la trace détaillée des objets traités et des traitement utilisés (les *Agents*). Nous distinguons deux grandes catégories d'agents : les experts humains et les programmes d'ordinateur. Ce modèle permet la coexistence d'estimations multiples pour le même concept, obtenues par différents agents à partir des mêmes données initiales. Ceci facilite les comparaisons entre ces estimations et l'évaluation des outils automatiques sur la base des références fournies par les experts humains.

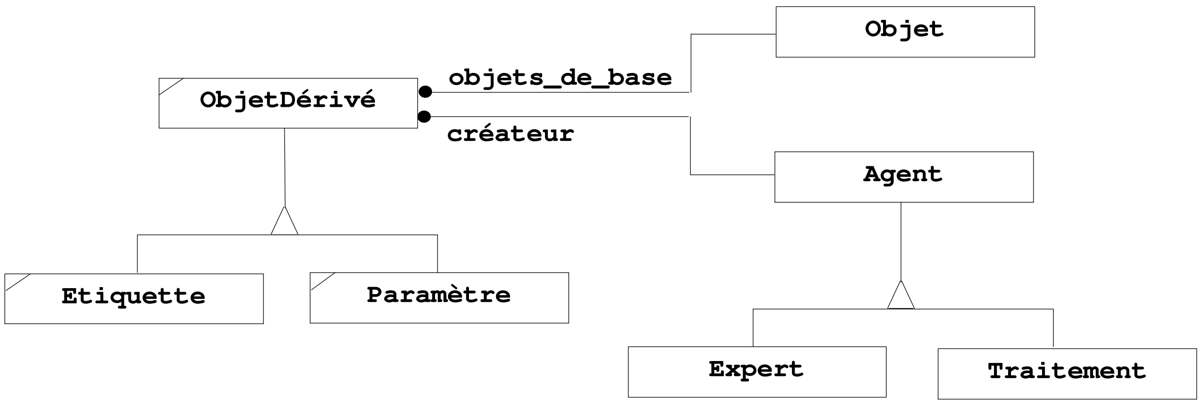



**Figure 5-4** Modélisation des données dérivées

Selon nous, les recherches sur le maintien de la cohérence dans le SIDOC pourront profiter de ce cadre de modélisation pour réaliser des mécanismes d'invalidation ou d'actualisation des données dérivées à la suite de modifications dans les objets traités ou dans les agents de traitement. Comme les données dérivées sont souvent impliquées dans la production d'autres données dérivées, à des niveaux d'abstraction plus élevés, ces liens produisent un réseau d'interdépendances par lequel les messages d'invalidation devront être propagés pour maintenir la cohérence globale des données.

### 5.3.5 L'organisation des données dans des collections

En plus de la modélisation des entités parole et des interconnexions sémantiques entre elles, nous avons dû concevoir aussi la façon de regrouper les instances correspondantes dans des collections et la structure macroscopique du système. Nous avons décidé de préserver le regroupement original des données primaires dans des "bases de données" pour trois raisons principales.

Premièrement, nous avons considéré que cette catégorisation sémantique rendra plus rapide l'accommodation des chercheurs avec ce nouveau système et facilitera l'exploration des données à la recherche d'informations intéressantes.

Deuxièmement, le volume de données gérées par le SIDOC est très grand et continuera à croître. Cette partition dans des ensembles réduits (*BasesDonnées*), qui correspondent aux bases de données originales, accélérera le traitement des requêtes sur chacune de ces collections, en ralentissant les requêtes globales. Comme le premier type de requêtes (qui se focalisent sur une base de données thématique) semble beaucoup plus fréquent que le deuxième, nous considérons que cette partition va améliorer les performances globales du système.

Troisièmement, la diversité de contenu des données primaires, en provenance de bases de données différentes, crée des problèmes dans l'interprétation des requêtes (interprétation des valeurs inconnues, par exemple). Comme les bases de données originales présentent une structure plus homogène, nous pouvons leur rattacher des méta-données qui permettent une meilleure interprétation de leur contenu et restreignent la formulation des requêtes pour garantir leur pertinence.

### 5.3.6 Réalisation informatique du SIDOC

A partir de cette modélisation objet des données parole, une première version opérationnelle du système SIDOC sous GemStone a été réalisée. J'ai participé à l'encadrement de deux stages dans notre laboratoire qui ont eu comme objectifs :

- l'implémentation du schéma du SIDOC (Annexe G) et l'importation des données correspondant à cinq bases de données parole (BDSONS, BDSONS3, BD-BRUIT, EUROM1 et PSH) depuis les anciennes versions relationnelles [Cristofor, 1995];

- la réalisation d'une interface de consultation qui permet la sélection des éléments de réalisation à partir des transcriptions orthographiques ou phonétiques, des caractéristiques des locuteurs et des conditions d'enregistrement, l'écoute et la visualisation des signaux acoustiques correspondants [La, 1995].

Des travaux menés en parallèle [Bensaber et *al.,* 1996] ont abouti à l'extension du schéma du SIDOC pour la prise en compte des aspects articulatoires, à l'importation de données articulatoires (films vidéo et radiologiques accompagnés des données dérivées correspondantes en provenance de BD-ART [Bensaber *et al.,* 1995]) et à l'adaptation de l'interface pour la consultation de ce nouveau type de données.

## 5.4 Le module d'extraction de connaissances

Nous revenons dans cette section sur les techniques d'extraction de connaissances abordées dans les chapitres précédents, pour proposer un cadre unitaire d'intégration de ces méthodes dans le système SIDOC.

### 5.4.1 Motivation

Avant tout, on peut se demander s'il est vraiment utile de rajouter ce type de fonctionnalité à un système déjà très complexe comme le SIDOC. La plupart des algorithmes d'apprentissage automatique mentionnés dans les chapitres précédents sont déjà disponibles pour la communauté scientifique en tant que logiciels isolés et nos propres contributions (PmBc, IDF, COPAC) ont été implémentées sous forme de programmes C++, directement utilisables par les chercheurs.

L'intérêt de regrouper plusieurs méthodes d'apprentissage automatique et de rendre homogène leur utilisation est évident. Aucune de ces méthodes n'a encore prouvé sa totale suprématie par rapport aux autres et la majorité des problèmes pratiques d'apprentissage demandent l'expérimentation simultanée de plusieurs approches différentes. Comme la plupart des implémentations existantes utilisent des descriptions des données, des modalités d'appel et des représentations des résultats distinctes (Annexe B), une grande partie du temps d'expérimentation est consommé par des tâches laborieuses et pas directement productives comme les conversions de formats. En outre, ces tâches demandent certaines connaissances en informatique qui ne sont pas forcément à la portée de tous les utilisateurs potentiels.

Selon nous, des éléments de standardisation seraient donc nécessaires pour simplifier les études comparatives, aussi bien pour la recherche en apprentissage automatique que pour l'utilisation en pratique de ces méthodes. On peut citer dans ce sens l'effort d'homogénéisation des bases de données fournies par l'UC-Irvine [Murphy & Aha, 1993] ou l'émergence de standards *de facto* comme le format C4.5 [Quinlan, 1993] de description des contextes de travail.

D'autres travaux vont dans le sens d'une intégration dans des plates-formes complètes d'expérimentation homogènes de plusieurs algorithmes d'apprentissage. C'est le cas, par exemple, des prototypes destinés à la recherche et disponibles dans le domaine public, tels que la plate-forme d'expérimentation écrite en Lisp par R. Mooney ou celle réalisée à l'Université Stanford dans le cadre du projet MLC++ [Kohavi *et al.,* 1994]. Un public plus large est visé par des environnements qui mettent plus l'accent sur l'ergonomie des interfaces : WEKA [Garner, 1995], SIPINA [Zighed, 1992] ou CLEMENTINE (Integral Solutions Ltd.), ces deux derniers étant des produits commerciaux.

Notre approche d'intégration de différents outils d'extraction de connaissances dans le SIDOC est fondée sur des motivations similaires mais apporte une dimension nouvelle par rapport à ces travaux : la connexion forte avec un système de gestion de bases de données. Selon nous, l'utilisation des fonctionnalités d'un SGBD pour le développement d'une telle application présente les avantages suivants :

- facilite et accélère les opérations très fréquentes effectuées par l'utilisateur sur les données pour la définition ou la modification des espaces de travail;
- facilite le contrôle du partage et de l'accès concurrent aux données par des mécanismes propres aux SGBD;
- permet une gestion systématique des résultats et du contexte des différentes expérimentations, extrêmement nécessaire pour la recherche;
- pourrait faciliter l'optimisation de certaines opérations internes d'accès aux données des algorithmes d'apprentissage.

### 5.4.2 Analyse du module d'extraction de connaissances

L'analyse des fonctionnalités qui devront être assurées par le module d'extraction de connaissances est fondée sur le modèle de coopération proposé dans la section 3.4.2 (figure 3-21) et nous conduit au scénario suivant d'interaction entre l'utilisateur et le système :

- **Définition de l'espace de travail**

  L'espace de travail doit contenir un ensemble d'exemples de test et, éventuellement, un ensemble d'exemples d'apprentissage et/ou une base de connaissances. Pour le construire, l'utilisateur peut récupérer une définition d'un espace de travail déjà existante ou peut définir un nouvel espace de travail en effectuant des sélections dans la base de données, par exemple.

- **Séance d'expérimentation**

  A chaque instant, l'utilisateur a le choix entre les opérations suivantes :

  - éditer la base de connaissances ou les ensembles d'exemples de test ou d'apprentissage;

  - tester la base de connaissances courante;

  - produire une nouvelle base de connaissances par induction automatique;

  - demander la révision automatique de la base de connaissances courante sur l'ensemble d'exemples d'apprentissage;

  - demander des justifications (sous forme d'exemples ou de contre-exemples) pour certains résultats du dernier processus d'induction, révision ou test;

  - enregistrer la configuration et les résultats de la dernière expérimentation.

La représentation schématique de ce scénario est présentée dans la figure 5-5 sous la forme d'un diagramme de flux de données.

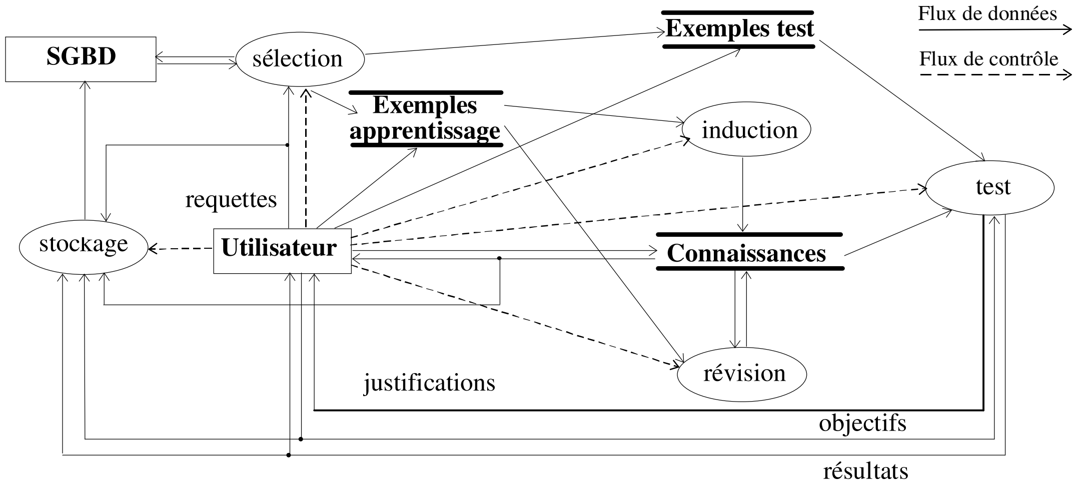



**Figure 5-5** Les composantes du module d'extraction de connaissances et leur interaction avec l'utilisateur et le SGBD

### 5.4.3 Conception des espaces de travail

Un espace de travail contient les informations persistantes (d'une séance d'expérimentation à une autre) sur le contexte d'une tâche de classification. Plusieurs espaces de travail peuvent être regroupés dans des "projets" correspondant à des applications similaires, pourvus des droits d'accès qui permettent l'exclusivité ou le partage de ces informations.

Nous avons accordé une attention particulière à l'obtention des exemples (pour l'apprentissage et pour le test), un aspect souvent négligé dans les réalisations similaires. Une application pratique d'apprentissage utilise rarement la totalité des objets d'une certaine classe (ou d'une certaine collection de la base de données) pour l'extraction ou le test des connaissances. Les différents paramètres, qui doivent être rigoureusement contrôlés dans ce type d'activité scientifique, imposent des restrictions (codées sous la forme de requêtes adressées au SGBD) qui délimitent des sous-ensembles d'objets pertinents pour l'étude en cours. D'habitude, ces sous-ensembles sont de taille substantiellement inférieure à celle des collections initiales. Les chercheurs doivent également disposer de la possibilité d'indiquer, parmi tous les traits qui caractérisent ces objets, ceux qui sont considérés comme nécessaires pour la tâche de classification analysée.

Il existe donc des différences importantes entre la structure et l'organisation des objets stockés dans la base de données et celles des exemples qui seront effectivement utilisés pour l'apprentissage ou pour le test d'une base de connaissances. La translation entre ces deux formes de représentation est une opération relativement coûteuse en temps de réponse. En tenant compte du fait que les mêmes ensembles d'exemples seront d'habitude utilisés dans un grand nombre de séances d'expérimentation, nous avons décidé de créer dans l'espace de travail des structures de stockage appropriées qui gardent les exemples (d'apprentissage et de test), une fois produits à partir des objets de la base de données. Ceci introduit une duplication des données (et des problèmes de maintien de la cohérence associés) mais améliore les temps de réponse à l'initialisation des séances d'expérimentation répétitives. Pour les espaces de travail moins souvent utilisés, l'utilisateur peut commander le remplacement des ensembles d'exemples stockés explicitement par des définitions implicites (des requêtes) qui permettent leur génération ultérieure.

La structure complexe des données représentées dans un formalisme à objets induit une légère complication supplémentaire. Comme la plupart des algorithmes d'apprentissage demandent que les exemples soient représentés sous la forme de tableaux d'enregistrements "plats", cette structure complexe des objets doit être convertie dans une représentation linéaire. L'interface qui permet cette conversion est schématisée dans la figure 5-6. Une solution différente est proposée dans [Manago & Kodratoff, 1991], où un algorithme d'apprentissage (ID3) est adapté pour une prise en compte directe de la représentation complexe des objets.

Pour réaliser ce type d'interface, nous avons dû concevoir un mécanisme de navigation entre les classes du SIDOC. Le langage de définition et de manipulation des données de GemStone est fondé sur un dialecte de Smaltalk (un langage de programmation à objets non-typé). Pour explorer les classes reliées à une classe donnée, GemStone favorise la navigation au long des références réalisées par les attributs (*instance variable*) de la classe source (et ceci seulement dans la mesure où ces références ont été pourvues de contraintes de type à la création de cette classe).

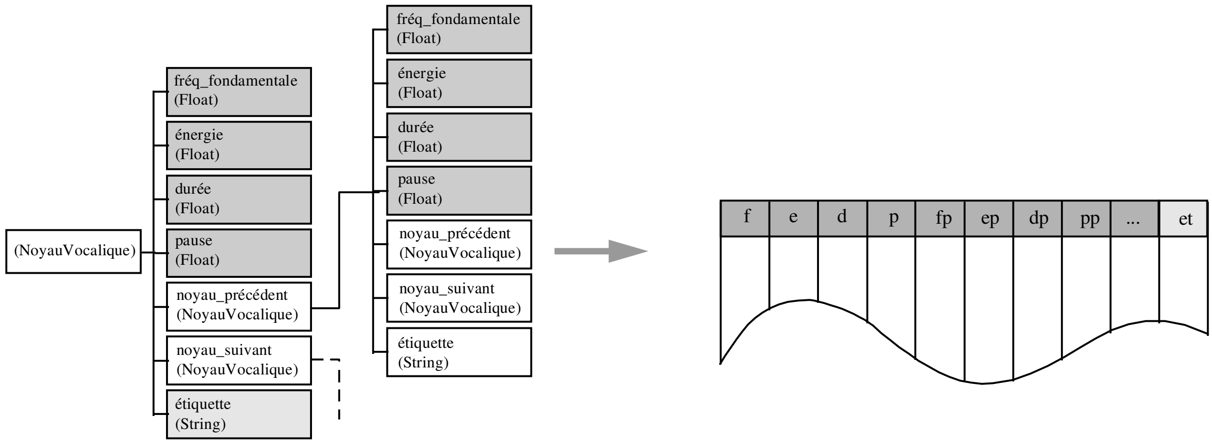



**Figure 5-6** Translation entre objets complexes et exemples
Ici, l'utilisateur indique l'inclusion de certains attributs de l'**objet** Noyau Vocalique courant et de son prédécesseur dans la description de l'**exemple** correspondant

Il est évident que l'utilisation directe de cette forme de navigation briserait l'encapsulation des objets et mettrait l'utilisateur en face de tous les détails de l'implémentation des objets. La navigation entre les classes au long des méthodes est par ailleurs rendue impossible par l'absence de toute information sur le type des valeurs retournées.

Nous avons préféré offrir à l'utilisateur une *vue* partielle de la base de données qui contient uniquement les classes pertinentes pour la recherche en parole. La navigation peut être réalisée au long d'une catégorie spéciale de méthodes de ces classes, sans paramètre et effet de bord, et pour lesquelles les concepteurs du SIDOC doivent fournir les méta-informations appropriées.

A part les ensembles d'exemples pour l'apprentissage et le test, un espace de travail contient aussi une base de connaissances de classification accessible à l'utilisateur humain et aux algorithmes d'induction ou de révision. Plusieurs formalismes de représentation des connaissances peuvent être envisagés (ils sont modélisés comme des sous-classes de la classe *Catégoriseur*) mais nous avons accordé une attention particulière aux représentations à base de règles, qui s'avèrent les plus accessibles pour la plupart des utilisateurs.

### 5.4.4 Conception des séances d'expérimentation

Un premier aspect à détailler dans la conception des séances d'expérimentation concerne l'intégration des différents algorithmes d'apprentissage dans le module d'extraction de connaissances. Il existe deux solutions différentes à ce problème : l'interfaçage du système avec des algorithmes d'apprentissage externes, déjà existants en tant que logiciels isolés, ou la re-programmation de ces algorithmes en tant que composants internes du module d'extraction de connaissances.

Nous avons adopté une solution mixte, comme dans MLC++. Elle consiste, d'abord, dans la spécification d'un protocole à base de fichiers pour la communication avec tout programme d'apprentissage externe qui pourrait ainsi être facilement interfacé avec notre système. Ensuite, une classe de base (*InducteurInterne*) définit également les caractéristiques des algorithmes d'apprentissage internes et leurs interaction avec les autres composantes du module. Cette deuxième possibilité d'intégration devrait permettre l'expérimentation des bénéfices apportés par l'utilisation des primitives d'un environnement de gestion de bases de données dans la programmation de ce type d'algorithme. Pour l'utilisateur, l'appel des deux catégories d'algorithmes d'apprentissage ne comporte aucune différence.

Pour homogénéiser la présentation des résultats et les mécanismes de contrôle de l'inférence, le test des bases de connaissances, fournies par l'utilisateur ou obtenues par apprentissage, est réalisé par une composante interne du module d'extraction de connaissances. C'est cette composante qui se charge également de la présentation d'exemples ou de contre-exemples pour justifier les résultats des tests.

Enfin, nous avons jugé utile, dans cette phase de conception, d'affiner la sélection des attributs qui figureront dans la description des exemples. Nous avons vu que, au moment de la constitution de l'espace de travail, l'utilisateur doit indiquer tous les traits jugés comme pertinents pour une tâche de classification particulière. Ces informations seront récupérées de la base de données et seront stockées dans les structures persistantes qui gardent les descriptions des exemples. Pour expérimenter l'apport de certains attributs, au cours d'une séance d'expérimentation, nous avons rajouté la possibilité pour l'utilisateur de demander leur inclusion ou leur exclusion dans la description des exemples qui seront effectivement considérés par les algorithmes d'apprentissage.

### 5.4.5 Réalisation informatique

L'implémentation des fonctionnalités d'extraction de connaissances spécifiées dans la section précédente a démarré récemment dans le cadre d'un projet d'ingénieur [Ripeanu, 1996] et un prototype de ce module est actuellement opérationnel dans le SIDOC (figure 5-7). Il intègre plusieurs algorithmes d'apprentissage automatique ou interactif de règles (C4.5+C4.5*rules*, CN2, FOIL, PmBc et COPAC) et permet le déroulement des séances d'expérimentation.

Pour l'instant, tous les algorithmes mentionnés ci-dessus ont été intégrés sous forme de logiciels externes, mais un premier essai a été conduit pour la programmation d'un algorithme d'apprentissage (CN2) dans l'environnement GemStone. [Ripeanu, 1996] fait état d'un gain important d'efficacité dans la programmation apporté par l'utilisation des primitives de manipulation et d'interrogation de données directement disponibles dans cet environnement (environ 500 lignes de code par rapport aux 7000 de l'implémentation originale en C fournie par son auteur) mais remarque aussi la difficulté de réaliser certaines optimisations due à l'inexistence de primitives comme "*count by group*" dans la version actuelle de GemStone.

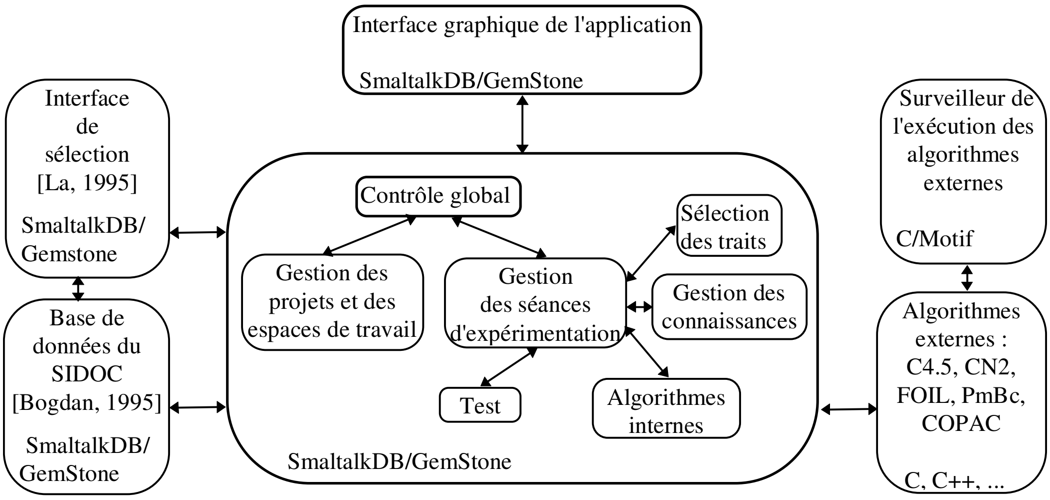



**Figure 5-7** Architecture du module d'extraction de connaissances

## 5.5 Conclusion

Nous avons présenté dans cette section quelques directions récentes explorées dans le cadre du projet SIDOC en vue de l'intégration des techniques d'extraction de connaissances dans un environnement homogène de gestion de données et de connaissances sur la parole. Dans un premier temps, ces travaux ont abouti à la définition d'un modèle objet général qui a permis l'intégration dans un même environnement de plusieurs bases de données parole avec des origines et des structures variées et la réalisation d'une application de gestion appropriée.

Nous avons présenté ensuite un cadre d'intégration des techniques d'extraction de connaissances dans cet environnement. L'architecture et la méthodologie proposées mettent l'accent sur l'apport du SGBD pour l'organisation et pour la gestion des espaces de travail et des résultats expérimentaux. La conception des séances d'expérimentation vise également à créer le cadre pour une véritable coopération entre l'utilisateur et les processus d'apprentissage pour l'extraction de connaissances dans les bases de données parole.

# Chapitre 6 :
# Conclusions et perspectives

L'objectif principal de cette thèse est d'apporter des arguments, d'ordre théorique et expérimental, pour soutenir l'intérêt des méthodes d'apprentissage symbolique dans la recherche en communication parlée proche des signaux physiques.

Dans un premier temps nous avons essayé de dresser un panorama succinct mais représentatif de la problématique abordée dans l'apprentissage symbolique supervisé, en privilégiant surtout les aspects qui pourraient intéresser plus particulièrement la communauté de recherche en parole.

Nous avons ensuite mis en évidence certaines caractéristiques de la plupart des tâches d'apprentissage issues de l'étude des signaux physiques de parole : prépondérance des attributs numériques, présence du bruit, importance de la prise en compte des connaissances scientifiques et expertes. Nos travaux de recherche en apprentissage automatique se sont donc orientés essentiellement dans ces directions.

Un des fils conducteurs de nos recherches a été la conviction que la similarité entre les exemples joue un rôle considérable dans l'apprentissage et que des progrès importants restent possibles dans cette direction, surtout en ce qui concerne le traitement des attributs numériques. Il s'agit principalement d'étudier la possibilité d'inclure, pour la résolution de tâches supervisées de classification, des critères d'évaluation des hypothèses apparentées à celles utilisées dans l'apprentissage non-supervisé. Plusieurs travaux de recherche, publiés récemment, montrent un intérêt croissant des chercheurs en apprentissage symbolique pour cette problématique.

A partir de ces réflexions, nous proposons un algorithme d'apprentissage original pour les espaces numériques, PmBc, qui combine des heuristiques issues de l'apprentissage non-supervisé, des techniques de classification fondées sur les plus proches voisins et de l'induction de règles. Des expériences sur plusieurs bases de données d'apprentissage, concernant des domaines d'application très variés, sont présentées pour appuyer la supériorité des performances de classification de PmBc par rapport à d'autres méthodes d'apprentissage automatique, fondées sur des principes similaires ou très différents.

A la différence de cette approche, plutôt empirique, les recherches qui ont été concrétisées par la conception de l'algorithme IDF d'induction d'arbres de décision sont fondées sur une analyse théorique de l'évaluation des hypothèses. Cette analyse vise à mettre en évidence et, éventuellement, à améliorer les composantes du processus d'induction, responsables de la prise en compte du bruit et de la similarité entre les exemples. Les résultats expérimentaux obtenus avec cet algorithme tentent de prouver que nous avons atteint le premier objectif visé par ces recherches : expliquer d'une façon rigoureuse certaines heuristiques performantes utilisées dans l'induction d'arbres de décision.

Une autre direction de recherche, qui a été abordée dans cette thèse, vise à améliorer les possibilités de coopération entre les méthodes d'apprentissage automatique et les experts humains. Dans cet esprit, nous avons conçu une méthodologie et un environnement informatique d'induction interactive de règles de classification, COPAC, qui permet l'intervention directe de l'utilisateur au niveau de toutes les étapes et de toutes les décisions de contrôle du processus d'apprentissage. Le degré d'automatisation de l'analyse peut être adapté d'une manière souple en fonction des connaissances disponibles et de la complexité de la tâche de classification.

Plusieurs méthodes de classification (parmi lesquelles les algorithmes que nous avons conçus) ont été expérimentées sur deux problèmes pratiques et représentatifs, issus des recherches en reconnaissance et synthèse de la parole menées dans notre laboratoire. Ces études ont mis en évidence les atouts et les faiblesses relatives des différentes catégories d'approches (symboliques, statistiques ou connexionistes), ont confirmé les performances escomptées de la part de nos algorithmes par rapport à d'autres méthodes d'apprentissage symbolique et ont apporté de nouveaux arguments en faveur de la coopération expert-machine en acquisition de connaissances.

Les travaux sur l'acquisition de connaissances prosodiques, nécessaires à l'agent de démarcation lexicale du système MICRO de compréhension de la parole (qui constitue le sujet d'une thèse de doctorat menée en parallèle avec nos travaux [Caillaud, 1996]), ont exploité intensément les techniques d'apprentissage symbolique et surtout l'algorithme PmBc issu de cette collaboration. L'utilisation d'un formalisme de représentation de connaissances facilement compréhensible a facilité de beaucoup l'interprétation des résultats obtenus par induction et la mise au point d'une configuration optimale de paramètres prosodiques. La méthodologie et les outils proposés dans COPAC ont reçu également une appréciation très positive de la part des experts concernés. De par leur caractère incrémental et interactif, ils ont facilité l'acceptation des résultats proposés par les techniques automatiques d'analyse, ils ont rendu possible l'amélioration par les experts de certains de ces résultats et même, d'après leurs affirmations, ils ont contribué au progrès de leurs propres connaissances dans le domaine.

Même si elles se sont avérées moins performantes que certaines méthodes de discrimination plus utilisées dans les recherches en parole, les techniques d'apprentissage symbolique ont joué un rôle important pour la caractérisation explicite des occlusives sourdes /p/, /t/ et /k/ (d'après le chercheur concerné : [Neagu, 1996], thèse en cours). Ceci lui a permis de mettre en évidence (par révision, à partir de règles produites par apprentissage) des prototypes de spectres du bruit d'occlusion assez différents de ceux utilisés généralement dans le passé. Les conclusions auxquelles il est arrivé à partir de cette étude (les spectres de /k/ en contexte /i/ sont plus proches de ceux de /t/ en contexte /y/ que de ceux de la même consonne /k/ en contexte /y/, par exemple) pourraient, selon lui, relancer le débat fondamental sur la nature des entités sous-lexicales indivisibles (unités phonétiques ou syllabiques).

L'utilisation effective des méthodes d'apprentissage symbolique pour les recherches en parole est conditionnée également par la réalisation de systèmes informatiques adaptés aux besoins et au profil des utilisateurs visés. L'objectif principal du projet SIDOC, auquel nous avons essayé d'apporter notre contribution, vise l'intégration, dans un environnement homogène et ergonomique, d'une large palette d'outils de gestion et de traitement des informations en évolution continuelle (données et connaissances), produites et exploitées par la recherche en parole. Nous sommes convaincus que l'apprentissage symbolique et les méthodes coopératives d'acquisition de connaissances ont un rôle important à jouer dans ce contexte. Dans l'esprit du projet SIDOC, la conception du module d'extraction de connaissances que nous proposons privilégie les aspects de modélisation et de gestion des données, souvent ignorés ou négligés dans les réalisations similaires.

Les perspectives de ces travaux de recherche nous permettent d'envisager, dans un premier temps, l'amélioration et l'extension des propositions présentées dans cette thèse. Il s'agit d'étendre l'applicabilité de PmBc par la prise en compte des attributs symboliques nominaux et hiérarchisés, des valeurs manquantes et éventuellement des tâches de révision de théorie. Nous envisageons aussi d'approfondir l'analyse théorique qui est à la base de IDF, pour la recherche de meilleures fonctions d'évaluation. Il est également très important d'évaluer ces méthodes sur d'autres applications de recherche en parole et d'accumuler de nouveaux résultats pour appuyer l'importance de la coopération expert-machine en acquisition de connaissances.

L'intégration des méthodes d'extraction de connaissances dans l'environnement SIDOC nous a aussi suggéré une autre direction de recherche possible : l'adaptation des algorithmes d'apprentissage symbolique pour une prise en compte directe du modèle objet. Nous considérons que les objets pourraient apporter à l'apprentissage automatique, d'une part, un formalisme expressif et facilement compréhensible pour la représentation des données et des connaissances et, d'autre part, la possibilité d'utiliser directement les opérateurs et les méthodes spécifiques aux différentes catégories de données. Finalement, les objets offrent tous les avantages de compatibilité d'un paradigme de modélisation qui semble recueillir actuellement le consensus de la plupart des disciplines issues de l'informatique.

# Bibliographie


[Adiba & Collet, 1993] M. Adiba, C. Collet, *Objets et bases de données - le SGBD $O_2$*, Hermès, 1993.

[Aha *et al.*, 1991] D.W. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, *Machine Learning*, Kluwer Academic Publishers, vol.6, p.37-66, 1991.

[Aha, 1995] D.W. Aha, An implementation and experiment with the nested generalized exemplars algorithm, *Technical Report AIC-95-003*, Navy Center for Applied Research in Artificial Intelligence, Washington D.C., 1995.

[Anthony & Biggs, 1992] M. Anthony, N. Biggs, *Computational learning theory: an introduction*, Cambridge Tracts in Theoretical Computer Science, vol.30, Cambridge University Press, 1992.

[Atkinson *et al.*, 1989] M. Atkinson, F. Bancilhon, D. De Witt, K. Dittrich, D. Mayer, S. Zdonik, The Object-Oriented Database System Manifesto, *Proceedings 1st DOOD*, Kyoto, Japon, 1989.

[Barnard & Cole, 1989] E. Barnard, R. Cole, A neural-net training program based on conjugate-gradient optimization, *Technical Report CSE 89-014*, Oregon Graduate Institute, 1989.

[Belbachir, 1991] H. Belbachir, *Maintien de la cohérence et interface utilisateur dans une base orientée objet. Application à la parole*, Thèse de Doctorat d'Etat de l'Université d'Oran Es Senia, 1991.

[Bensaber *et al.*, 1995], K. Bensaber, J.F. Serignat, P. Perrier, BD-ART : Multimedia Articulatory Database, *Conference on Linguistic Databases*, Groningen, 1995.

[Bensaber *et al.*, 1996], K. Bensaber, P. Munteanu, J.F Serignat, P. Perrier, Object-Oriented Modeling of Articulatory data for Speech Research Information Systems, *First International Conference On Database Systems For Advanced Applications*, Melbourne, Australie (soumis).

[Blumer *et al.*, 1989] A. Blumer,, A. Ehrenfeucht, D. Haussler, M.K. Warmuth, Learnability and the Vapnik-Chervonenkis dimension, *J. ACM*, 36(4), p. 929-965, 1989.

[Booch, 1991] G. Booch, *Object-Oriented Design with applications*, Benjamin/Cummings, 1991.

[Bourlard & Morgan, 1995] H. Bourlard, N. Morgan, Connectionist Techniques, *dans [Cole, 1995]*, p. 412-418, 1995.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and regression trees*, Wadsworth International Group, Belmont, 1984.

[Brunk & Pazzani, 1991] C.A. Brunk, M.J. Pazzani, An investigation of noise-tolerant relational concept learning algorithmes, *Proceedings of the 8th International Workshop on Machine Learning*, p..389-393, Evanston, Illinois, 1991.

[Cai *et al.*, 1991] Y. Cai, N. Cercone, J. Han, Attribute-oriented induction in relational databases, dans *[Piatesky-Shapiro & Frawley ,1991]*, p.213-228, 1991.

[Caelen-Haumont, 1991] G. Caelen-Haumont, *Stratégies des locuteurs en réponse à des consignes de lecture d'un texte : analyse des interactions entre les modèles syntaxiques, sémantiques, pragmatiques et paramètres prosodiques*, Thèse de doctorat d'Etat, Université de Provence Aix Marseille I, Aix-en-Provence, 1991.

[Caelen-Haumont, 1993] G. Caelen-Haumont, , Prosody and Discourse Interpretation, *Proceedings of Eurospeech 93*, ESCA, p. 1991-1994, Berlin, 1993.

[Caillaud, 1996], B. Caillaud, *Apprentissage automatique de connaissances prosodiques pour un système de compréhension automatique de la parole*, Thèse de doctorat INPG, 1996.

[Caillaud *et al.*, 1996] B. Caillaud, P. Munteanu, J.F. Serignat, J. Caelen, Prosodic Knowledge Acquisition for Lexical Access Improvement, *The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology (CC AI)*, Communication and Cognition, Belgique (à paraître).

[Carpenter, 1992] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, « Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps », *IEEE Transactions on Neural Networks*, 3(5), 1992.

[Cervantes, 88] O.Cervantes-Villagomez, *Bases de données et d'objets complexes multimédia pour la recherche sur la parole*, Thèse de doctorat INPG, 1988.

[Cestnik & Bratko, 1991] B. Cestnik, I. Bratko, On estimating probabilities in tree pruning, *Proceedings of EWSL-91*, Y. Kodratoff (ed.), 1991.

[Christiensen, 1992] A.D.Christiensen, Learning to predict in uncertain continuous tasks, *Machine Learning: Proceedings of the Ninth International Workshop*, San Mateo, E.U., 1992.

[Clark & Boswell, 1991] P. Clark, R. Boswell, Rule Induction with CN2: Some Recent Improvements, *Machine Learning - Proceedings of the Fifth European Conference (EWSL-91)*, p.151-163, Springer-Verlag, Berlin, 1991.

[Clark & Niblett, 1989] P. Clark, T. Niblett, The CN2 Induction Algorithm, *Machine Learning*, vol.3, no.4, p.261-283, Kluwer Academic Publishers, 1989.

[Coad & Yourdon, 1991] P. Coad, E. Yourdon, *OOA: Object Oriented Analysis*, Prentice-Hall, 1991.

[Cohen,1993] W.W. Cohen, Efficient pruning methods for separate-and-conquer rule learning systems, *Proceedings of the 13th International Joint Conference on Artificial*

*Intelligence*, p.988-994, Chambery, 1993.

[Cole, 1995] A. Cole (ed.), *Survey of the State of the Art in Human Language Technology*, **http://www.cse.ogi.edu/CSLU/HLTSurvey/HLTSurvey.html**, CSLU, 1995.

[Cover & Hart, 1967], T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, 13, p. 21-27, 1967.

[Cristofor, 1995] L.B Cristofor, *Transfert de bases de données parole d'un modèle relationnel à un modèle orienté objet*, Projet d'ingénieur de l'Ecole d'Automatique et Ordinateurs de Bucarest réalisé à l'ICP, 1995.

[Descout *et al.*, 1986] R. Descout, J.F. Serignat, O. Cervantes, R. Carré, 1986, BDSONS : Une base de données des sons du français, *12-th Int. Congress on Acoustics*, A4-7, Toronto, Canada, 1986.

[De Mori & Brugnara, 1995] R. De Mori, F. Brugnara, HMM Methods in Speech Recognition, *dans [Cole, 1995]*, p. 24-34, 1995.

[Diday *et al.*, 1983] Diday E., Lemaire J., Pouget P., Testu F., Eléments d'analyse de données, Dunod, Paris, 1983.

[Dietterich & Michalski., 1983], T.G. Dietterich, R.S. Michalski, A comparative review of selected methods for learning from examples, *dans [Michalski et al., 1983]*, 1983.

[Fayyad & Irani, 1992] U.M. Fayyad, K.B. Irani, On the Handling of Continuous-Valued Attributes in Decision Tree Generation, *Machine Leaning*, Kluwer Academic Publishers, vol.8, p.87-102, 1992.

[Fayyad & Irani, 1993] U.M. Fayyad, K.B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *Proceedings of the 13th IJCAI*, p. 1022-1027, Chambery, France, 1993.

[Fernandez, 89] Y. Fernandez Ordonez de Villasenor, *Gestion de connaissances pour des applications du domaine de la parole*, Thèse de doctorat INPG, 1989.

[Flexer, 1996] A. Flexer, Statistical Evaluation of Neural Network Experiments: Minimum Requirements and Current Practice, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, p. 1005-1008, Vienne, Autriche, 1996.

[Fourcin *et al.*, 1989] A.J. Fourcin, G. Harland, W. Barry, V. Hazan, *Speech input and output assessment; multilingual methods and standards*, Ellis Horwood, 1989.

[Fourcin & Dolmazon, 1991] A. Fourcin, J.M. Dolmazon, Speech Knowledge, Standards and Assessment, *Proc. of the 12-th International Congress of Phonetic Sciences*, vol.5, p.430-433, August, Aix-en-Provence, 1991.

[Freitas & Lavington, 1996] A.A. Freitas, S.H. Lavington, Using SQL Primitives and Parallel DB Servers to Speed up Knowledge Discovery in Large Relational Databases, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, Vienne, Autriche, 1996.

[French *et al.*, 1990] J.C. French, A.K. Jones, J.L. Pfaltz (eds), *Scientific Database Management*, Invitational NSF Workshop on Scientific Database Management, Technical Reports 90-21 & 90-22, University of Virginia, Charlottesville, E.U., 1990.

[Fürnkranz & Widmer, 1994] J. Fürnkranz, G. Widmer, Incremental Reduced Error Pruning, *Proceedings of the 11th International Conference on Machine Learning*, p.70-77, 1994.

[Gaillat, 1975] G. Gaillat, *Une procédure statistique de décision avec apprentissage et son application à la reconnaissance des caractères manuscrits*, Thèse Paris 6, 1974.

[Garner, 1995] S.R. Garner, WEKA: The Waikato Environment for Knowledge Analysis, *Proc. New Zealand Computer Science Research Students Conference*, Waikato, Nouvelle Zélande, 1995.

[Gascuel & Gallinari, 1994] O. Gascuel, P. Gallinari (coordinateurs), *Méthodes Symboliques-Numériques de Discrimination*, Rapport final d'activité (Projet Inter-PRCs), 1994.

[Goodman & Smyth, 1988] R.M. Goodman, P. Smith, Information-theoretic rule induction, *European Conference on Artificial Intelligence*, 1988.

[Han *et al.*, 1996] J. Han, Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, O.R. Zaiane, DBMiner: A System for Mining Knowledge in Large Relational Databases, *Proceedings of the International Conference on Data Mining and Knowledge Discovery (KDD'96)*, p.250-255, Portland, Oregon, 1996.

[Heath *et al.*, 1993] D.Heath, S. Kasif, S. Salzberg, Induction of Oblique Decision Trees, *Proceedings of the 13th IJCAI*, Chambéry, 1993.

[Helmbold *et al.*, 1989] D.Helmbold, R. Sloan, M. Warmuth, Learning nested differences of intersection closed concept classes, *Proceedings of the 1989 Workshop on Computational Learning Theory*, San Mateo, E.U., 1989.

[Hendler, 1989] J.A. Hendler (ed), Special issue on hybrid systems (symbolic/connectionist), *Connection Science*, 1, 1989.

[Hinton, 1990] G.E. Hinton (ed), Special issue on connectionist symbol processing, *Artificial Intelligence*, 46, 1990.

[Holsheimer & Kersten, 1994] M. Holsheimer, M.L. Kersten, Architectural Support for Data Mining, *Report CS-R9429*, CWI, Amsterdam, 1994.

[Holsheimer & Siebes, 1994] M. Holsheimer, A. Siebes, Data mining: the search for knowledge in databases, CWI, *Technical Report CS-R9406*, Amsterdam, 1994.

[Hun *et al.*, 1996] E. Hun, J. Marin, P.J. Stone, *Experiments in Induction*, Academic Press, New York, E.U., 1966.

[Jaynes, 1994] E.T. Jaynes, *Probability theory: the logic of science*, édition fragmentaire de juin 1994.

[Kaufman *et al.*, 1991] K.A. Kaufman, R.S. Michalski, L. Kerschberg, Mining for Knowledge in Datbases: Goals and General Descritions of the INLEN System, dans *[Piatesky-Shapiro & Frawley ,1991]*, p.449-462, 1991.

[Kerber, 1992] R. Kerber, ChiMerge: Discretization of Numeric Attributes, *Proceedings of the 10th National Conference on Artificial Intelligence*, p.123-128, San José, E.U., 1992.

[Kietz & Wrobel, 1992] J.U. Kietz, S. Wrobel, Controlling the complexity of learning in logic through syntactic and task-oriented models, *Inductive logic Programming*, S. Muggleton (ed.), Academic Press, Londres, 1992.

[Kohavi *et al.*, 1994] R. Kohavi, G. John, R. Long, D. Manley, K. Pfleger, MLC++: A Machine Learning Library in C++, *Tools with Artificial Intelligence*, p.740-743, 1994.

[La, 1995] C.H. La, *Réalisation de l'interface du SIDOC sous GemStone*, Mémoire de DESS Informatique, U.J.F., réalisé à l'ICP, 1995.

[Lacher *et al.*, 1992] R.C. Laucher, S.I. Hruska, D.C. Kuncicky, Back-Propagation Learning in Expert Networks, *IEEE Transactions on Neural Networks*, vol. 3, no. 1, 1992.

[Lamel *et al.*, 1991] L.F. Lamel, J.L. Gauvain, M. Eskenazi, BREF, a Large Vocabulary Spoken Corpus for French, *2-nd European Conf. on Speech Com. and Technology (EUROSPEECH-91)*, vol.2, p.505-508, Gênes, Italie, 1991.

[Leonard, 1984] R.G. Leonard, A database for speaker-independent digit recognition, *Proceedings of the 1984 ICASSP*, vol.3, p.42.11-14, IEEE., 1984.

[Littlestone, 1988] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, *Machine Learning*, vol.2, p.285-318, Kluwer Academic Publishers, 1988.

[Mahoney & Mooney, 1993] J.F. Mahoney, R.J. Mooney, Combining neural and symbolic learning to revise probabilistic rule bases, *Advances in Neural Information Processing Systems*, vol. 5, S. Hanson, J. Cowans, L. Gilles (eds), Morgan Kaufmann, 1993.

[Manago & Kodratoff, 1991] M. Manago, Y. Kodratoff, 1991, Induction of Decision Trees from Complex Structured Data, dans *[Piatesky-Shapiro & Frawley ,1991]*, p. 289-306, 1991.

[Michalski, 1980] R.S. Michalski,, Pattern recognition as rule-guided inductive inference, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.2, p. 349-361, 1980.

[Michalski, 1983] R.S.Michalski, « A Theory and Methodology of Inductive Learning », *dans [Michalski et al., 1983]*, 1983.

[Michalski *et al.*, 1983] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, 1983.

[Michalski *et al.*, 1986] R.S. Michalski, I. Mozetic, J. Hong, N. Lavrac, The multi-purpose incremental learning system AQ15 and its testing application to three medical domains, *Proceedings of the 5th national conference on Artificial Intelligence*, p.1041-1045, Philadelphia, 1986.

[Mitchell, 1977] T.M. Mitchell, Version spaces: A candidate elimination approach to rule learning, *Proceedings of the Fifth IJCAI*, p.305-310, Cambridge, 1977.

[Mooney & Califf, 1995] R.J. Mooney, M.E. Califf, Induction of First-Order Decision Lists: Results on Learning the Past Tense of English Verbs, *Journal of Artificial Intelligence Research*, 3, p.1-24, 1995.

[Muggleton & Feng, 1992] S. Muggleton, C. Feng, Efficient induction of logic programs, *Inductive Logic Programming*, S. Muggleton (ed.), p. 281-298, Academic Press, London, 1992.

[Muggleton & De Raedt, 1994] S. Muggleton, L. De Raedt, Inductive Logic Programming: Theory and Methods, *Journal of Logic Programming*, vol.19, no.20, p.629-679, 1994.

[Munteanu, 1993] P. Munteanu, *Représentation et test des connaissances en communication parlée - paradigme de l'étiquetage prosodique*, Mémoire de D.E.A. de l'INPG, 1993.

[Munteanu & Serignat, 1994a] P. Munteanu; J.F. Serignat, Test et mise au point des paquets de règles de production pour la communication parlée, *Actes des 20-èmes Journées d'Étude sur la Parole*, p.291-296, Trégastel, 1994.

[Munteanu & Serignat, 1994b] P. Munteanu, J.F. Serignat, Mise au point de connaissances sur la parole dans le SIDOC, *Sciences et technologies cognitives, 3ème Colloque du Pôle Rhône-Alpes de Sciences Cognitives*, Lyon, 1994.

[Munteanu *et al.*, 1995] P. Munteanu, B. Caillaud, J.F. Serignat, An Integrated Environment for Speech Expert Systems Tuning, *Seventh International Conference on Artificial Intelligence & Expert Systems Applications (EXPERSYS-95)*, p.3-8, San Francisco, E.U., 1995.

[Munteanu *et al.*, 1996] P. Munteanu, B. Caillaud, J.F. Serignat, An Original Environment for Cooperative Knowledge Test, Revision and Discovery, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, p.979-984, Vienne, Autriche, 1996.

[Murphy & Aha, 1993] P.M. Murphy, D.W. Aha, UCI Repository of Machine Learning Databases, **`http://www.ics.uci.edu/~mlearn/MLRepository.html`**, University of California, Irvine, 1993.

[Neagu, 1994] A. Neagu, *Mise en oeuvre des traitements distribués pour SIDOC - Application à la caractérisation des aspects vocaliques en termes de résonances du conduit vocal*, Mémoire de D.E.A. de l'INPG, 1994.

[Neagu, 1996] A Neagu, *Analyse articulatoire du signal de parole. Suivi des résonances du conduit vocal*, Thèse I.N.P.G. en cours.

[Ourston & Mooney, 1990] D. Ourston, R. Mooney, Changing the rules: A comrehensive approach to theory refinement, *Proceedings of the 8-th national conference on Artificial Intelligence*, p.815-820, Detroit, 1990.

[Parzen , 1962] E. Parzen, On estimation of a probability density function and mode, *Annals of Mathematical Statistics*, 33, p. 1065-1076, 1962.

[Pazzani & Kibler, 1992] M. Pazzani, D. Kibler, The utility of background knowledge in inductive learning, *Machine Learning*, vol.9, p.57-94, Kluwer Academic Publishers, 1992.

[Piatesky-Shapiro & Frawley ,1991] G. Piatesky-Shapiro, W.J. Frawley (eds), *Knowledge Discovery in Databases*, AAAI Press, Menlo Park, 1991.

[Pitt & Valiant, 1988] L. Pitt, L. Valiant, Computational limitations on learning from examples, *J. ACM*, 35(4), p. 965-984, 1988.

[Price *et al.*, 1988] P. Price, W.M. Fisher, J. Bernstein, D.S. Pallet, The DARPA 1000 word Ressource Management Database for Continuous Speech Recognition, *Proceedings ICASSP*, vol.2., p.651-654, April, New York, E.U. , 1988.

[Prince *et al.*, 1991] V.Prince, L. Bochereau, P. Bourgine, G. Deffuant, B. Cahour, P. Falzon, J.M. Hoc, J.P. Krivine, J.M. David, P. Levy, Y.M. Visetti, C. Vogel, *Intellectica - Expertise et Sciences Cognitives*, ARC, 1991.

[Quinlan, 1986] J.R. Quinlan, Induction of decision trees, *Machine Learning*, 1(1), p.81-106, 1986.

[Quinlan, 1988] J.R. Quinlan, Generating production rules from examples, *Tenth International Joint Conference on Artificial Inntelligence*, p.304-307, 1988.

[Quinlan, 1990] J.R. Quinlan, Learning logical definitions from relations, *Machine Learning*, vol.5, p.239-266, Kluwer Academic Publishers, 1990.

[Quinlan, 1993] J.R. Quinlan, *C4.5 : Programs for Machine Learning*, Morgan Kaufmann, 1993.

[Quinlan, 1995] J.R. Quinlan, Learning First-Order Definitions of Functions, *à paraître dans le Journal of Artificial Intelligence Research*.

[Quinlan, 1996] J.R. Quinlan, Improved Use Of Continuous Attributes in C4.5, *Journal of Artificial Intelligence Research*, 4, p. 77-90, 1996.

[Rabaséda & Zighed, 1996] S. Rabaséda, A. Zighed, Extraction et simplification des règles dans les graphes d'induction, *Proceedings ARAE*, 1996.

[Rakotomalala *et al.*, 1996] R. Rakotomalala, A. Zighed, S. Rabaséda, Validation on rules isuued from induction graph, *Proceedings IPMU'96*, p. 1259-1264, Grenade, Espagne, 1996.

[Rauber *et al.*, 1993] T.W. Rauber, M.M. Barata, A.S. Steiger-Garção, A toolbox for analysis and visualisation of sensor data in supervision, *Tooldiag'93*, Toulouse, France, 1993.

[Reilly *et al.*, 1982] D.L. Reilly, L.N. Cooper, C. Erlbaum, A neural network for category learning, *Biological Cybernetics*, vol.45, p. 35-41, 1982.

[Ripeanu, 1996] M. Ripeanu, *Interface pour la mise au point de connaissances dans le SIDOC*, Projet d'ingénieur de l'Ecole d'Automatique et Ordinateurs de Bucarest réalisé à l'ICP, Grenoble, 1996.

[Rissanen, 1978] J. Rissanen, Modeling by shortest data description, *Automatica*, vol.14, p.465-471, 1978.

[Rivest, 1994] R. Rivest, *Machine Learning*, cours M.I.T., Massachusetts, 1994.

[Rumbaugh *et al.*, 1991] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object-oriented Modeling and Design*, Prentice-Hall International, Inc., Englewood Cliffs, E.U., 1991.

[Rumelhart *et al.*, 1986] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, *Parallel Distributed Processing*, vol.1, chap.8, MIT press, Cambridge, 1986.

[Salzberg, 1991] S. Salzberg, A Nearest Hyperrectangle Learning Method, *Machine Learning*, 6, p. 251-276, 1991.

[Sanger, 1991] T.D. Sanger, A tree-structured algorithm for reducing computation in networks with separable basis functions, *Neural Computation*, 3(1), 1991.

[Saporta, 1990] G. Saporta, *Probabilités, analyse des données et statistique*, Editions Technip, Paris, 1990.

[Shapiro, 1983] E.Y. Shapiro, *Algorithmic program debugging*, MIT Press, Massachusetts, 1983.

[Shavlik & Dietterich, 1990], J.W. Shavlik, T.G. Dietterich (eds), *Readings in machine learning*, Morgan Kaufmann, 1990.

[Simpson, 1992] Simpson P.K., « Fuzzy Min-Max Neural Networks—Part 1: Classification », *IEEE Transactions on Neural Networks*, 3(5), 1992.

[Shavlik, 1992] J.W. Shavlik, A Framework for Combining Symbolic and Neural Learning, *Technical Report 1123*, Computer Science Departement, University of Wisconsin, Madison, E.U., 1992.

[Stonebraker *et al.*, 1990] The Comittee for Adavanced DBMS Function, Third-Generation Database System Manifesto, *Technical Report UCB/ERL M90/28*, University of California, Berkeley, 1990.

[Sun & Alexandre, 1995] R. Sun, F. Alexandre (eds), *Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, IJCAI, Montreal, Canada, 1995.

[Sun *et al.*, 1992] R. Sun, L. Bookman, S. Shekhar, *The Working Notes of the AAAI Workshop on Integrating Neural and Symbolic Processes: The Cognitive Dimension*, San Jose, E.U, 1992.

[Terrenoire, 1970] M. Terrenoire, *Un modèle mathématique de processus d'interrogation : les pseudo-questionnaires*, Thèse Grenoble 2, 1970.

[Tounissoux, 1980] Tounissoux D., *Processus séquentiels adaptatifs de reconnaissance de formes pour l'aide au diagnostic*, Thèse Lyon 1, 1980.

[Towell & Shavlik, 1991] G. Towell, J.W. Shavlik, Refining symbolic knowledge using neural networks, *Proceedings of the International Workshop on Multistrategy Learning*, p.257-272, West Valley, 1991.

[Utgoff, 1988] P.E. Utgoff, Perceptron trees: A case study in hibrid concept representations, *Proceedings of the Seventh National Conference on Artificial Intelligence*, p.601-606, St. Paul, MN, 1988.

[Utgoff, 1989] P.E. Utgoff, Incremental induction of decision trees, *Machine Learning*, vol.4, p.161-186, 1989.

[Utgoff & Brodley, 1990] P.E. Utgoff, C.E. Brodley, An incremental method for finding multivariate splits for decision trees, *Proceedings of the Seventh ICML*, p.58-65, Morgan Kaufmann, 1990.

[Valiant, 1984] L.G. Valiant, A theory of the learnable, *Communications of ACM*, vol.27, no.11, p.1134-1142, 1984.

[Van de Merckt, 1993] T. Van de Merckt, Decision Trees in Numerical Attribute Spaces, *Proceedings of the 13th IJCAI*, Chambéry, p.1016-1021, 1993.

[Vue *et al.*, 1991] V. Zue, J. Glass, D. Goodine, L. Hirshman, H. Leung, M. Philips, J. Polifroni, S. Seneff, The MIT ATIS System: Preliminary Development, Spontaneus Speech Data Collection and Performance Evaluation, *2-nd European Conf. on Speech Com. and Technology (EUROSPEECH-91)*, vol.2, p.537-540, Gênes, Italie, 1991.

[Webb, 1996] G.I. Webb, Further Experimental Evidence against the Utility of Occam's Razor, *Journal of Artificial Intelligence Research*, 4, p.397-417, 1996.

[Wettschereck & Dietterich, 1995] D. Wettschereck, T.G. Dietterich, An Experimental Comparaison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms, *Machine Learning*, 19(1), p.5-28, 1995.

[Zeiliger, 1992] J. Zeiliger, *Représentation des connaissances en décodage acoustico-phonétique dans une BDBC-Parole orientée-objet*, Mémoire de D.E.A. de l'INPG, 1992.

[Zeiliger *et al.*, 1992] J. Zeiliger, J.F. Serignat, D. Autesserre, J.M. Dolmazon, Eurom1 : Une base de données parole multilingue - Partie française, *19- èmes Journées d'Étude sur la Parole*, S.F.A., p.303-306, Bruxelles, Belgique, 1992.

[Zeiliger *et al.*, 1994] J. Zeiliger, J.F. Serignat, D. Autesserre, C. Meunier, Bdbruit : Une base de données parole de locuteurs soumis à du bruit, *20-èmes Journées d'Étude sur la Parole*, S.F.A., p.287-290, Trégastel, 1994.

[Zighed, 1985] D.A. Zighed, *Méthodes et outils pour les processus d'interrogation non-arborescents*, Thèse Lyon 1, 1985.

[Zighed, 1992] D.A. Zighed, J.P. Auray, G. Duru, *SIPINA : Méthodes et logiciel*, Lacassagne, 1992.

[Zighed & Rakotomalala, 1996], D.A. Zighed, R. Rakotomalala, A method for non arborescent induction graphs, *Rapport Technique E.R.I.C.*, Université Lyon 2, 1996.

[Zighed *et al.*, 1996] D.A. Zighed, R. Rakotomalala, S. Rabaséda, A Discretisation method of continuous attributes in induction graphs, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, p.997-1002, Vienne, Autriche, 1996.

# Liste des publications concernant ces travaux


*Chapitre dans ouvrage*

P. Munteanu, J.F. Serignat, Apprentissage automatique symbolique, *Commande et supervision des procédés: Aspects théoriques et méthodes numériques*, section "Intelligence artificielle en automatique", S. Gentil, D. Popescu. (eds), Editura Tehnica, Bucarest (accepté en 1996, à paraître).

*Revues avec comité de lecture*

B. Caillaud, P. Munteanu, J.F. Serignat, J. Caelen, Prosodic Knowledge Acquisition for Lexical Access Improvement, *The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology* (CC AI), Communication and Cognition, Belgique (accepté en 1996, à paraître).

P. Munteanu, B. Caillaud, J.F. Serignat, Apprentissage par généralisation dans les domaines numériques basé sur la similarité des exemples, *Revue d'Intelligence Artificielle* (RIA), Editions Hermes (soumis en 1996).

*Conférences internationales avec actes et comité de lecture*

P. Munteanu; J.F. Serignat, Test et mise au point des paquets de règles de production pour la communication parlée, *Actes des 20-èmes Journées d'Étude sur la Parole* (JEP), p. 291-296, Trégastel, juin 1994.

P. Munteanu, B. Caillaud, J.F. Serignat, An Integrated Environment for Speech Expert Systems Tuning, *Seventh International Conference on Artificial Intelligence & Expert Systems Applications* (EXPERSYS), p. 3-8, San Francisco, Etats-Unis, novembre 1995.

P. Munteanu, B. Caillaud, J.F. Serignat, An Original Environment for Cooperative Knowledge Test, Revision and Discovery, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research* (EMCSR), avril 1996, p. 979-984, Vienne, Autriche.

K. Bensaber, P. Munteanu, J.F Serignat, P. Perrier, Object-Oriented Modeling of Articulatory data for Speech Research Information Systems, *First International Conference On Database Systems For Advanced Applications* (DASFAA), Melbourne, Australie (soumis), avril 1997.

*Colloques sans actes*

P. Munteanu, J.F. Serignat, Mise au point de connaissances sur la parole dans le SIDOC, *Sciences et technologies cognitives, 3ème Colloque du Pôle Rhône-Alpes de Sciences Cognitives*, Sciences et Technologies Cognitives, Lyon, novembre 1994.

P. Munteanu, J.F. Serignat, IDF : induction d'arbres de décision par l'approximation des fréquences, *6ème Journée du Groupe de Travail sur l'Apprentissage*, E3I, Tours, septembre 1996.

# Annexes

**Annexe A : Versions Internet des algorithmes d’apprentissage présentés**

**Annexe B : Résultats de l'apprentissage des algorithmes présentés dans la section 2.3 sur la base *breast-w***

**Annexe C : Caractéristiques des bases de données utilisées pour les comparaisons d'algorithmes**

**Annexe D : Tailles (nombres de noeuds) des arbres de décision construits par C4.5 sur les bases de données numériques**

**Annexe E : Règles obtenues à l'aide de COPAC pour les syllabes appartenant aux mots grammaticaux**

**Annexe F : Règles de classification des occlusives sourdes à partir du spectre du bruit d'occlusion codé en Barks**

**Annexe G : Schéma des classes de la base de données de SIDOC**

**Annexe A : Versions Internet des algorithmes d'apprentissage présentés**

*Pour les conditions précises de distribution, consultez les fichiers de copy-right sur le site indiqué.*

- **`ftp://ftp.cs.su.oz.au/pub/ml/patch-R8.tar.Z`**
  (*Release 8* de C4.5)

- **`ftp://ftp.cs.utexas.edu/pub/pclark/cn2.tar.Z`**
  (CN2)

- **`http://www.gmd.de/ml-archive/software/bnge`**
  (NGE et BNGE)

- **`ftp://ftp.cs.su.oz.au/pub/foil6.sh`**
  (FOIL)

- **`ftp://ftp.cs.su.oz.au/pub/ffoil1.sh`**
  (FFOIL)

- **`http://www.cs.utexas.edu/users/ml/neither.html`**
  (NEITHER, une extension de EITHER)

- **`ftp://ftp.uninova.pt/users/tr/soft/tooldiag-2.0.tar.gz`**
  (TOOLDIAG, outils statistiques dont un classificateur gaussien)

- **`http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/`**
  **`areas/speech/systems/ogitools/ogitools.tgz`**
  (outils pour le traitement de la parole dont NOPT)

**Annexe B : Résultats de l'apprentissage des algorithmes présentés dans la section 2.3 sur la base *breast-w***

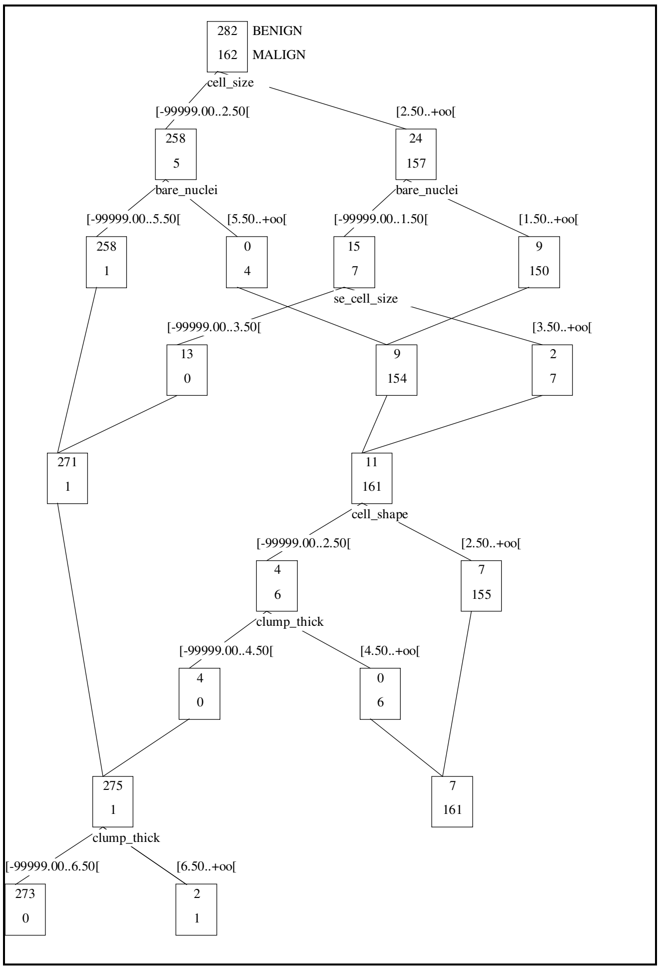
**C4.5 :**

```
Cell_Size <= 2 :
|  Bare_Nuclei <= 2 : BENIGN (244.0)
|  Bare_Nuclei > 2 :
|  |  Cell_Size <= 1 : BENIGN (13.0)
|  |  Cell_Size > 1 : MALIGN (6.0/1.0)
Cell_Size > 2 :
|  Cell_Shape <= 2 :
|  |  Clump_Thick <= 5 : BENIGN (14.0/1.0)
|  |  Clump_Thick > 5 : MALIGN (4.0)
|  Cell_Shape > 2 :
|  |  Bare_Nuclei > 1 : MALIGN (151.0/6.0)
|  |  Bare_Nuclei <= 1 :
|  |  |  SE_Cell_Size <= 3 : BENIGN (4.0)
|  |  |  SE_Cell_Size > 3 : MALIGN (8.0/1.0)
```

**CN2** :

```
IF      Cell_Size < 3.50
  AND  Bare_Nuclei < 1.50
THEN  trait = BENIGN  [250 0]

IF      Clump_Thick < 5.50
  AND  1.50 < Bare_Nuclei < 5.50
  AND  Norm_Nucl < 1.50
THEN  trait = BENIGN  [21 0]

IF      Clump_Thick < 6.50
  AND  SE_Cell_Size < 9.00
  AND  Bare_Nuclei < 4.50
  AND  Bland_Chroma < 3.50
  AND  Mitoses < 2.50
THEN  trait = BENIGN  [259 0]

IF      SE_Cell_Size < 3.50
  AND  Bare_Nuclei < 1.50
THEN  trait = BENIGN  [245 0]

IF      Cell_Size > 4.50
  AND  Marginal_Adh > 1.50
THEN  trait = MALIGN  [0 109]

IF      Clump_Thick > 6.50
  AND  Bare_Nuclei > 1.50
  AND  Norm_Nucl < 7.00
THEN  trait = MALIGN  [0 56]

IF      Norm_Nucl > 8.50
THEN  trait = MALIGN  [0 51]

IF      Bare_Nuclei > 9.00
  AND  Mitoses < 2.50
THEN  trait = MALIGN  [0 63]

IF      Clump_Thick > 6.00
  AND  SE_Cell_Size > 5.50
THEN  trait = MALIGN  [0 35]

IF      Clump_Thick < 4.50
  AND  Bare_Nuclei > 7.50
THEN  trait = MALIGN  [0 15]

(DEFAULT) trait = BENIGN  [282 162]
```

**SIPINA :**




**FOIL :**

is_BENIGN(A,B,C,D,E,F,G,H,I) :- B<=2, F<=3.

is_BENIGN(A,B,C,D,E,F,G,H,I) :- C<=2, A<=4.

is_BENIGN(A,B,C,D,E,F,G,H,I) :- F<=1, E<=3.

is_BENIGN(A,B,C,D,E,F,G,H,I) :- G<=3, A<=6, D<=1, B<>E.

is_BENIGN(A,B,C,D,E,F,G,H,I) :- C<=4, B<=E, H>6, E<=4, F<=7.

où A= Clump_Thick, B= Cell_Size, C= Cell_Shape, D= Marginal_Adh, E= SE_Cell_Size, F=Bare_Nuclei, G= Bland_Chroma, H= Norm_Nucl, I= Mitoses et les variables B et E et, respectivement, F et H ont été déclarées en étant du même type.

## BNGE :

| | | | | | |
|---|---|---|---|---|---|
| MALIGN : | | | | BENIGN : | |
| **Rectangle: 1** | | **Rectangle: 68** | | **Rectangle: 163** | |
| Clump_Thick: | 1.000 10.000 | Clump_Thick: | 5.000 10.000 | Clump_Thick: | 1.000 8.000 |
| Cell_Size: | 2.000 10.000 | Cell_Size: | 3.000 7.000 | Cell_Size: | 1.000 4.000 |
| Cell_Shape: | 2.000 10.000 | Cell_Shape: | 2.000 4.000 | Cell_Shape: | 1.000 5.000 |
| Marginal_Adh: | 1.000 10.000 | Marginal_Adh: | 1.000 4.000 | Marginal_Adh: | 1.000 6.000 |
| SE_Cell_Size: | 2.000 10.000 | SE_Cell_Size: | 3.000 6.000 | SE_Cell_Size: | 1.000 10.000 |
| Bare_Nuclei: | 8.000 10.000 | Bare_Nuclei: | 1.000 3.000 | Bare_Nuclei: | 1.000 3.000 |
| Bland_Chroma: | 4.000 10.000 | Bland_Chroma: | 3.000 7.000 | Bland_Chroma: | 1.000 7.000 |
| Norm_Nucl: | 1.000 10.000 | Norm_Nucl: | 2.000 10.000 | Norm_Nucl: | 1.000 8.000 |
| Mitoses: | 1.000 10.000 | Mitoses: | 3.000 10.000 | Mitoses: | 1.000 2.000 |
| **Rectangle: 10** | | **Rectangle: 84** | | **Rectangle: 188** | |
| Clump_Thick: | 2.000 10.000 | Clump_Thick: | 10.000 | Clump_Thick: | 1.000 4.000 |
| Cell_Size: | 5.000 10.000 | Cell_Size: | 3.000 4.000 | Cell_Size: | 1.000 4.000 |
| Cell_Shape: | 2.000 10.000 | Cell_Shape: | 3.000 6.000 | Cell_Shape: | 1.000 4.000 |
| Marginal_Adh: | 2.000 10.000 | Marginal_Adh: | 1.000 4.000 | Marginal_Adh: | 1.000 4.000 |
| SE_Cell_Size: | 2.000 10.000 | SE_Cell_Size: | 3.000 | SE_Cell_Size: | 1.000 6.000 |
| Bare_Nuclei: | 1.000 10.000 | Bare_Nuclei: | 3.000 5.000 | Bare_Nuclei: | 1.000 5.000 |
| Bland_Chroma: | 2.000 10.000 | Bland_Chroma: | 4.000 7.000 | Bland_Chroma: | 1.000 7.000 |
| Norm_Nucl: | 1.000 10.000 | Norm_Nucl: | 3.000 10.000 | Norm_Nucl: | 1.000 3.000 |
| Mitoses: | 1.000 10.000 | Mitoses: | 1.000 2.000 | Mitoses: | 1.000 7.000 |
| **Rectangle: 14** | | **Rectangle: 103** | | **Rectangle: 222** | |
| Clump_Thick: | 7.000 10.000 | Clump_Thick: | 5.000 | Clump_Thick: | 5.000 8.000 |
| Cell_Size: | 2.000 10.000 | Cell_Size: | 3.000 4.000 | Cell_Size: | 3.000 4.000 |
| Cell_Shape: | 2.000 10.000 | Cell_Shape: | 5.000 6.000 | Cell_Shape: | 4.000 6.000 |
| Marginal_Adh: | 1.000 10.000 | Marginal_Adh: | 5.000 8.000 | Marginal_Adh: | 3.000 5.000 |
| SE_Cell_Size: | 2.000 10.000 | SE_Cell_Size: | 3.000 4.000 | SE_Cell_Size: | 3.000 4.000 |
| Bare_Nuclei: | 4.000 10.000 | Bare_Nuclei: | 1.000 3.000 | Bare_Nuclei: | 1.000 7.000 |
| Bland_Chroma: | 1.000 3.000 | Bland_Chroma: | 4.000 8.000 | Bland_Chroma: | 4.000 7.000 |
| Norm_Nucl: | 1.000 9.000 | Norm_Nucl: | 10.000 | Norm_Nucl: | 3.000 8.000 |
| Mitoses: | 1.000 3.000 | Mitoses: | 1.000 | Mitoses: | 1.000 2.000 |
| **Rectangle: 39** | | **Rectangle: 125** | | **Rectangle: 231** | |
| Clump_Thick: | 7.000 10.000 | Clump_Thick: | 5.000 6.000 | Clump_Thick: | 5.000 6.000 |
| Cell_Size: | 4.000 10.000 | Cell_Size: | 3.000 | Cell_Size: | 7.000 8.000 |
| Cell_Shape: | 7.000 10.000 | Cell_Shape: | 2.000 3.000 | Cell_Shape: | 7.000 8.000 |
| Marginal_Adh: | 1.000 5.000 | Marginal_Adh: | 1.000 3.000 | Marginal_Adh: | 1.000 |
| SE_Cell_Size: | 2.000 6.000 | SE_Cell_Size: | 3.000 6.000 | SE_Cell_Size: | 3.000 5.000 |
| Bare_Nuclei: | 1.000 7.000 | Bare_Nuclei: | 4.000 10.000 | Bare_Nuclei: | 4.000 8.000 |
| Bland_Chroma: | 2.000 7.000 | Bland_Chroma: | 3.000 4.000 | Bland_Chroma: | 3.000 |
| Norm_Nucl: | 1.000 10.000 | Norm_Nucl: | 1.000 | Norm_Nucl: | 4.000 7.000 |
| Mitoses: | 1.000 5.000 | Mitoses: | 1.000 | Mitoses: | 1.000 |
| **Rectangle: 67** | | **Rectangle: 159** | | **Rectangle: 397** | |
| Clump_Thick: | 5.000 7.000 | Clump_Thick: | 2.000 | Clump_Thick: | 6.000 |
| Cell_Size: | 4.000 10.000 | Cell_Size: | 3.000 | Cell_Size: | 3.000 |
| Cell_Shape: | 6.000 | Cell_Shape: | 4.000 | Cell_Shape: | 3.000 |
| Marginal_Adh: | 1.000 4.000 | Marginal_Adh: | 4.000 | Marginal_Adh: | 5.000 |
| SE_Cell_Size: | 6.000 10.000 | SE_Cell_Size: | 2.000 | SE_Cell_Size: | 3.000 |
| Bare_Nuclei: | 1.000 4.000 | Bare_Nuclei: | 5.000 | Bare_Nuclei: | 10.000 |
| Bland_Chroma: | 4.000 | Bland_Chroma: | 2.000 | Bland_Chroma: | 3.000 |
| Norm_Nucl: | 3.000 10.000 | Norm_Nucl: | 5.000 | Norm_Nucl: | 5.000 |
| Mitoses: | 1.000 10.000 | Mitoses: | 1.000 | Mitoses: | 3.000 |

## Annexe C : Caractéristiques des bases de données utilisées pour les comparaisons d'algorithmes

Le tableau suivant présente les 15 bases de données [Murphy & Aha, 1993] utilisées dans la partie expérimentale de cette thèse.

| Abréviation | Domaine | Nb exemples | Nb classes | Nb attributs |
|---|---|---|---|---|
| *breast-w*[3] | cancer du poumon | 699 | 2 | 9 |
| *bupa* | problèmes médicaux chez les alcooliques | 345 | 2 | 6 |
| *diabetes* | diabètes chez les indiens Pima | 768 | 2 | 8 |
| *glass* | identification de type de verre par leur composition | 214 | 6 | 9 |
| *ionosphere* | évaluation de la pertinence d'information radar | 351 | 2 | 34 |
| *iris* | classification des iris (plantes) | 150 | 3 | 4 |
| *letter* | reconnaissance de caractères | 20000 | 26 | 16 |
| *new-thyroid* | classification du fonctionnement de la glande thyroïde | 215 | 3 | 5 |
| *segment* | segmentation d'images | 2310 | 7 | 19 |
| *sonar* | classification de données sonar | 208 | 2 | 60 |
| *vehicle* | reconnaissance de silhouette | 846 | 4 | 18 |
| *vowel* | reconnaissance de voyelles en anglais | 990 | 11 | 10 |
| *waveform*[4] | données synthétiques de caractérisation de formes (bruitées) | 5000 | 3 | 21 |
| *waveform40*[5] | idem + attributs aléatoires | 5000 | 3 | 40 |
| *wine* | classification des vins par leur composition chimique | 178 | 3 | 13 |

On peut remarquer la grande variété des domaines couverts :

- traitement d'images : 4 bases (*iris*, *letter*, *segment*, *vehicle*),
- médecine : 4 bases (*breast-w*, *bupa*, *diabetes*, *new-thyroid*),
- traitement du signal : 4 bases (*sonar*, *vowel*, *waveform*, *waveform40*),

---

[3] Cette base de données a été fournie au serveur de l'UCI par Dr. William H. Wolberg

[4] Nous avons utilisé pour ces deux expériences les bases de données directement disponibles sur le serveur de l'UCI (ce serveur fournit aussi des logiciels générateurs d'exemples).

- autres : 3 bases (*glass*, *ionosphere*, *wine*).

**Annexe D : Tailles (nombres de noeuds) des arbres de décision construits par C4.5 sur les bases de données numériques**

| Base | C4.5 Moy | ± | Base | C4.5 Moy | ± | Base | C4.5 Moy | ± |
|---|---|---|---|---|---|---|---|---|
| *breast-w* | 21.36 | 4.99 | *iris* | 7.84 | 1.22 | *vehicle* | 116.8 | 17.69 |
| *bupa* | 42.8 | 10.18 | *letter* | 2165.92 | 35.09 | *vowel* | 177.16 | 8.74 |
| *diabetes* | 38.84 | 15.4 | *new-thyroid* | 13.4 | 2.65 | *waveform* | 485.36 | 29.07 |
| *glass* | 42.84 | 4.16 | *segment* | 77.2 | 6.8 | *waveform40* | 534.32 | 20.31 |
| *ionosphere* | 22.76 | 4.36 | *sonar* | 25.12 | 2.87 | *wine* | 9.92 | 2.03 |

## Annexe E : Règles obtenues à l'aide de COPAC pour les syllabes appartenant aux mots grammaticaux

**SI** $F0(n) \leq 0.35$
**ET** $Energy(n+1) \geq 0.59$
**ET** $Duration(n-1) \geq 0.13$
**ET** $Duration(n) \leq 0.18$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.77$
**ET** $F0(n) \geq 0.45$
**ET** $Duration(n-1) \geq 0.37$
**ET** $Duration(n) \leq 0.35$
**ET** $Duration(n+1) \geq 0.23$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.48$
**ET** $Energy(n) \leq 0.8$
**ET** $Duration(n-1) \leq 0.22$
**ET** $Duration(n) \leq 0.2$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.39$
**ET** $F0(n) \leq 0.4$
**ET** $Energy(n) \leq 0.5$
**ET** $Energy(n+1) \leq 0.32$
**ET** $Duration(n+1) \geq 0.06$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.69$
**ET** $Energy(n) \leq 0.56$
**ET** $Duration(n+1) \leq 0.44$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.43$
**ET** $F0(n) \leq 0.52$
**ET** $F0(n+1) \geq 0.2$
**ET** $F0(n+1) \leq 0.45$
**ET** $Energy(n) \leq 0.75$
**ET** $Energy(n+1) \geq 0.56$
**ET** $Duration(n+1) \leq 0.63$
**ALORS GRAMATICAL**

**SI** $F0(n) \leq 0.39$
**ET** $Duration(n-1) \geq 0.61$
**ET** $Pause(n) \geq 0.01$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.54$
**ET** $F0(n) \leq 0.62$
**ET** $F0(n+1) \leq 0.37$
**ET** $Energy(n) \geq 0.5$
**ET** $Duration(n) \leq 0.8$
**ET** $Duration(n+1) \leq 0.44$
**ALORS GRAMATICAL**

**SI** $F0(n-1) \geq 0.66$
**ET** $F0(n) \geq 0.48$
**ET** $F0(n) \leq 0.76$
**ET** $F0(n+1) \leq 0.62$
**ET** $Energy(n) \geq 0.58$
**ET** $Duration(n-1) \geq 0.25$
**ET** $Duration(n) \leq 0.54$
**ALORS GRAMATICAL**

**Note:** Les seuils numériques s'appliquent aux paramètres prosodiques normalisés dans l'intervalle [0,1] en suivant les procédures mentionnées dans la section 4.1.2.

## Annexe F : Règles de classification des occlusives sourdes à partir du spectre du bruit d'occlusion codé en Barks

### Paquet de règles proposé par FFOIL

**SI** brk(1) > brk(9)
**ET** brk(6) > -6,19227
**ALORS p**

**SINON**
**SI** brk(5) > brk(8)
**ET** brk(9) > -0,002645
**ALORS k**

**SINON**
**SI** brk(8) > 7.9044
**ALORS t**

**SINON**
**SI** brk(8) ≤ brk(9)
**ET** brk(8) > -2,00143
**ET** brk(5) > -8,23726
**ALORS t**

**SINON**
**SI** brk(2) ≤ -9,14839
**ET** brk(1) > -11,6223
**ALORS k**

**SINON**
**SI** brk(7) > -0,006131
**ET** brk(8) > -3,4642
**ET** brk(7) ≤ 4,49489
**ALORS t**

**SINON**
**SI** brk(5) > -8,83826
**ALORS k**

**SINON**
**ALORS p**

### Paquet de règles révisé par l'expert

**SI** brk(1) > brk(9)
**ET** brk(2) > brk(4)
**ALORS p**

**SINON**
**SI** brk(4) > brk(6)
**ALORS k**

**SINON**
**SI** max(brk(5), brk(6))-min(brk(7), brk(8)) > 6,5
**ALORS k**

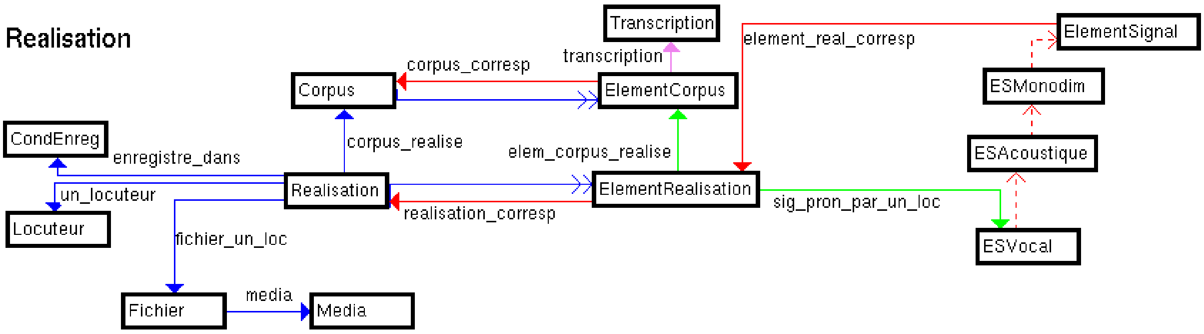
**SINON**
**SI** max(brk(5), brk(6)) > brk(8)
**ALORS t**

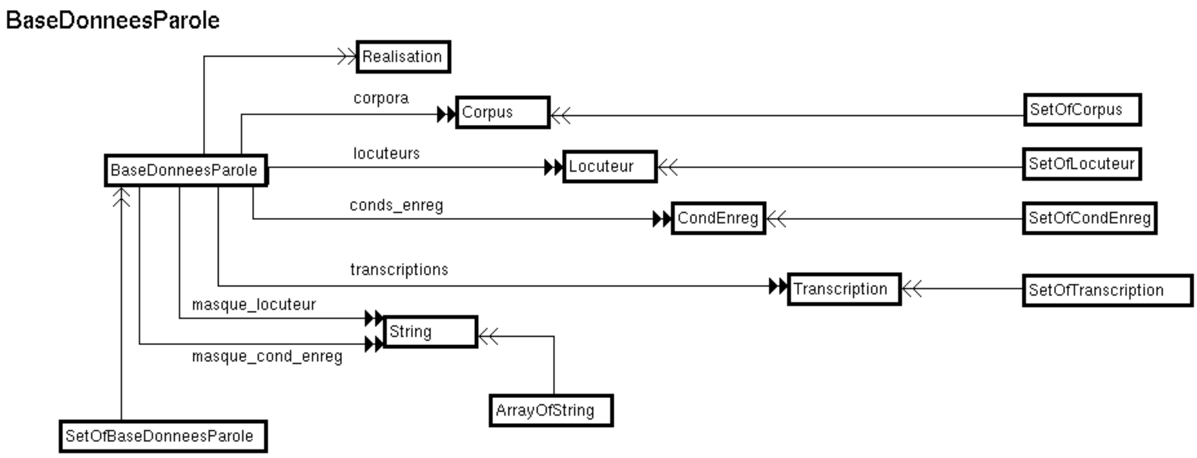
**SINON**
**SI** max(brk(9), brk(10))-brk(8) > 1
**ALORS t**

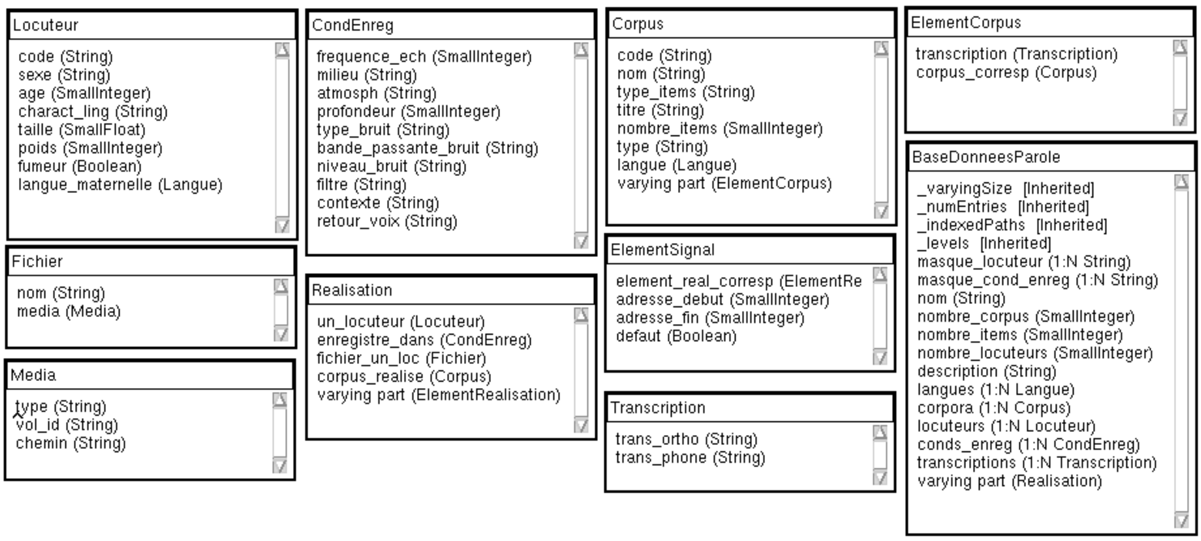
**SINON**
**SI** brk(8)-brk(10) > 7
**ALORS t**

**SINON**
**ALORS k**

# Annexe G : Schéma des classes de la base de données de SIDOC



## RESUME

L'objectif principal de cette thèse est d'apporter des arguments, d'ordre théorique et expérimental, pour soutenir l'intérêt des méthodes d'apprentissage symbolique dans la recherche en communication parlée proche des signaux physiques.
Dans un premier temps, nous présentons nos contributions en apprentissage symbolique qui visent à répondre à certaines particularités de ce domaine. PmBc est un algorithme original d'apprentissage par généralisations successives dans les espaces numériques, qui combine des heuristiques issues de l'apprentissage non-supervisé, de la classification statistique et de l'induction de règles. L'analyse théorique sur laquelle est fondé notre deuxième algorithme, IDF, fournit une explication rigoureuse et des perspectives d'amélioration des heuristiques généralement utilisées dans l'induction d'arbres de décision, notamment dans le traitement des attributs numériques. Les performances de ces deux algorithmes ont été validées sur un grand nombre de bases de données utilisées généralement pour la comparaison des résultats d'apprentissage. Notre troisième contribution, COPAC, est un environnement informatique d'induction interactive de règles de classification qui met l'accent sur la coopération entre les méthodes d'apprentissage automatique et les experts humains.
Nous présentons ensuite des résultats expérimentaux encourageants obtenus sur deux problèmes représentatifs des besoins en recherche dans le domaine de la parole : l'acquisition de connaissances prosodiques pour l'amélioration de l'accès lexical et la caractérisation des occlusives sourdes.
La dernière partie de cette thèse est consacrée à 'intégration des méthodes d'apprentissage symbolique dans l'environnement SIDOC de gestion de données et de connaissances sur la parole, fondé sur le modèle objet.



## MOTS CLES

apprentissage symbolique, induction d'hyperrectangles, induction d'arbres de décision, induction interactive, extraction de connaissances à partir des données (ECD), bases de données parole, modélisation objet, prosodie



## ABSTRACT

The main objective of this thesis is to provide theoretical and experimental support of the use of machine learning methods issued from Artificial Intelligence in signal analysis for speech research.
We first present some personal contributions in machine learning, motivated by the particularities of this domain. PmBc is an original learning algorithm for numerical spaces, based on successive generalizations and combining heuristics issued from unsupervised learning, from statistical classification techniques and from rule induction. The theoretical analysis which underlies our second algorithm, IDF, provides a rigorous explanation and suggests possible improvement of some heuristics largely used in decision trees induction, mainly for numerical features processing. The performances of these two algorithms have been validated on a great number of databases generally used for comparing learning results. Our third contribution, COPAC, is an interactive rule induction environment which emphasizes the collaboration between human experts and machine learning methods.
We then present some encouraging experimental results obtained on two representative problems issued from speech research : the acquisition of prosodic knowledge for lexical access improvement and the characterization of voiceless occlusive consonants.
The last part of this thesis is dedicated to the integration of machine learning methods in SIDOC object-oriented speech data and knowledge management environment.



## KEYWORDS

machine learning, induction of hyperrectangles, induction of decision trees, interactive induction, knowledge discovery in databases (KDD), speech databases, object modeling, prosody