Classification interactive

B. Crémilleux
GREYC, CNRS - UPRESA 6072
Université de Caen
14032 Caen Cédex France
cremilleux@info.unicaen.fr

Résumé. Après un bref panorama des principales méthodes de classification supervisée, nous montrons à partir de l'une d'entre entre elles (les arbres de décision) que l'utilisateur joue un rôle primordial lors de leur déroulement. Afin de l'associer explicitement, les grands axes d'un atelier d'aide à la classification interactive sont présentés. Cet atelier propose des outils pour la réalisation des tâches annexes - mais indispensables - à celle de classification à proprement parler car nous pensons que cette approche doit être prise dans sa globalité pour être réellement efficace.

Mots-clés: classification, interaction personne-système, arbre de décision, données incertaines, élgage, indice de qualité.

1. Introduction

Les méthodes de classification encore appelées discrimination ou classement ont pour but de déterminer la classe d'appartenance d'objets caractérisés par des descripteurs. Un objet - ou exemple - est une plante, une molécule chimique, un malade ou encore une situation (demande de prêt bancaire pour un client, déclenchement d'un processus d'alerte en fonction de signaux reçus par des capteurs,…). Généralement, l'un des descripteurs est la partie à expliquer - ou classe -, les autres forment la partie explicative (par exemple, l'âge, le poids, des données biologiques ou encore des résultats d'examens dans le cas d'un malade). La partie à expliquer se rapporte à une propriété particulière de l'exemple que l'on veut à la fois éclaircir et prédire sur de nouveaux exemples à partir de la partie explicative. Toujours pour le cas d'un malade, il peut s'agir d'un choix diagnostic, d'une aide à la détermination d'une stratégie thérapeutique. Ces méthodes s'appliquent à un grand nombre d'activités humaines et sont souvent associées à des mécanismes d'aide à la décision.
D'un point de vue opératoire, elles se décomposent en deux étapes. D'abord, une phase d'*apprentissage* où la classe des exemples est connue. Elles construisent alors un classifieur qui au vu de la partie explicative d'un nouvel exemple permet de prévoir sa classe. Pour certains domaines, cette prédiction ne peut être qu'incertaine. Il est parfois possible d'expliciter le fonctionnement du classifieur sous la forme d'un ensemble de règles de décision. Une deuxième étape est la phase de *test*. Un ensemble d'exemples non utilisés en apprentissage et dont la classe est volontairement omise est présenté au classifieur afin d'étudier ses performances. Si les exemples sont peu nombreux, des techniques comme le bootstrap [BRE 96] ou la cross-validation [BRE & al. 84; SCH 96b] permettent d'employer plusieurs fois un même exemple en apprentissage et/ou en test.
Ces méthodes sont souvent présentées en reléguant l'utilisateur au second plan, l'accent étant mis sur leur principe et leur adéquation avec les données et le but recherché. Pourtant, de par leur fonctionnement même qui nécessite l'existence de données pré-classifiées pour la phase d'apprentissage, ces méthodes requièrent la présence d'un expert pour classer des exemples typiques ou la possession de collection de cas. Les exemples de ces collections sont souvent

sous une forme brute et sont à travailler pour que ces méthodes soient capables de les traiter. Il est donc légitime de s'attendre déjà à ce que cette étape quasi-obligatoire de préparation des données influe sur le résultat.
Cet article poursuit un double objectif. D'une part, à partir de l'exemple d'une méthode de classification, il vise à montrer que l'utilisateur intervient implicitement tout au long du processus de production du résultat et pas seulement au niveau de la préparation des données. D'autre part, il propose les grands axes d'un atelier d'aide à la classification interactive comportant, outre des méthodes de classification, des outils aidant l'utilisateur à réaliser les tâches connexes. En effet, nous pensons que pour aller réellement vers une démarche de classification interactive associant l'utilisateur de façon fructueuse, il est important de lui offrir un cadre de travail englobant l'ensemble des tâches intervenant dans une approche de classification.
Le second paragraphe effectue un aperçu des méthodes les plus utilisées en classification. Nous nous plaçons ici uniquement dans le contexte de la classification supervisée et nous n'abordons pas la classification non supervisée couramment appelée "catégorisation" en intelligence artificielle [FIS 87; LEB 87]. Dans cette dernière approche, les classes ne sont pas pré-déterminées et les exemples utilisés pour l'apprentissage ne sont pas pré-classifiés. Ces méthodes construisent une hiérarchie de concepts en optimisant les différences et les similarités entre les classes émergeantes. La frontière entre classification supervisée et non supervisée n'est pas toujours clairement marquée par les systèmes informatiques. Par exemple, COBWEB [FIS 87], issu de l'apprentissage non supervisé réalise aussi de la classification supervisée. Fisher & al. [FIS & al. 93] effectuent un panorama des techniques d'apprentissage supervisé et non supervisé dans le contexte de l'analyse et de l'extraction de la connaissance à partir de bases de données et présentent ces approches dans un cadre unifié.
A partir de la stratégie de classification des arbres de décision, le troisième paragraphe met en évidence le rôle de l'utilisateur dans l'obtention du classifieur ainsi qu'un ensemble de tâches implicites effectuées par l'utilisateur. Forts de cette expérience, nous proposons au quatrième paragraphe l'ossature d'un atelier d'aide à la classification interactive laissant une place centrale à l'utilisateur et l'associant à l'ensemble des tâches liées à la question de la classification.

2. Classification supervisée

Ce paragraphe présente succinctement les principales méthodes de classification. Celles-ci sont regroupées en quelques familles : réseaux neuronaux, probabilités et statistiques, programmation linéaire, systèmes symboliques. Le projet StatLog [MIC & al. 94] présente les approches les plus classiques et en effectue une comparaison pour mieux mettre en évidence leurs forces et faiblesses. Ce projet teste 2 algorithmes basés sur les réseaux neuronaux, 6 systèmes symboliques et 8 provenant des statistiques sur 12 bases issues du monde réel et communément utilisées par la communauté de l'apprentissage. Les critères de comparaison sont des critères quantitatifs comme le temps d'apprentissage et le taux d'erreur et qualitatifs comme la compréhension du processus employé, la facilité d'utilisation, la robustesse face aux différents types de paramètres d'entrée. Les principales conclusions de cette étude sont : il n'y a pas un unique meilleur classifieur, on peut simplement établir des caractéristiques d'utilisation avec lesquelles certains classifieurs sont plus efficaces ; les algorithmes symboliques obtiennent de faibles taux d'erreur lorsque les descripteurs sont symboliques et/ou les données sont très dispersées et leurs processus décisionnels sont plus facilement compréhensibles, et enfin ce sont les réseaux neuronaux et les méthodes issues des statistiques qui obtiennent les résultats les plus proches. Remarquons que pour ce projet, afin de faciliter les comparaisons, certaines extensions de systèmes, non possédées par les autres, non pas été prises en compte. Nous présentons maintenant brièvement ces familles.

2.1 Réseaux neuronaux

Les réseaux de neurones se sont développés à l'origine par analogie avec le fonctionnement du cerveau humain, capable d'extraordinaires possibilités d'apprentissage et de plasticité. Un tel réseau est constitué de cellules reliées entre elles par des liens valués. Une cellule "simule" le

fonctionnement d'un neurone : elle reçoit un ensemble de données en entrée et produit une sortie par application d'une fonction (appelée fonction de transfert) souvent de type sigmoïde ou tangente hyperbolique. L'ensemble de ces paramètres marque le comportement du réseau auquel ils sont attachés et situe cette famille de classifieur comme méthode numérique. L'apprentissage consiste à déterminer, à partir d'exemples, les valeurs optimales (au sens d'un certain critère) de ces paramètres.
Il existe une multiplicité de réseaux de neurones qui diffèrent de par leurs structures (nombre de cellules, topologie des liens), les critères optimisés et les algorithmes d'apprentissage [HER & al. 91]. En classification, on utilise souvent le modèle du perceptron multi-couches [RUM & al. 86]. Dans ce modèle, les cellules sont rangées en couches successives avec des connexions d'une couche à la suivante. Le nombre de couches et de cellules par couches sont fixés par l'utilisateur. L'apprentissage s'effectue en modulant les poids des liaisons inter-cellules suivant l'écart mesuré entre les sorties désirées et les sorties calculées. Cet écart, indiquant l'adéquation entre le réseau et les données, est caractérisé par une mesure de risque. Le but de cette correction lors de l'apprentissage est de stabiliser le réseau dans un état optimal (au sens du critère utilisé) pour la classification.
L'adéquation des réseaux multicouches à rétropropagation à de nombreuses tâches de classification a été prouvée [VIE & al. 92], notamment dans des domaines comme la reconnaissance de caractères ou la reconnaissance d'objets dans une image.

2.2. Méthodes probabilistes et statistiques

Les méthodes probabilistes et statistiques de classement se distinguent de par leur champ d'action [CAR & al. 96].
Lorsque la répartition des descriptions pour chaque classe est connue, on est amené à mettre en œuvre des méthodes probabilistes. Cette connaissance est le plus souvent complète et donnée sous forme analytique. On considère qu'aucun exemple ne peut venir la remettre en cause aussi cette situation extrême est peu réaliste en pratique. Ces méthodes (dont les règles d'affectation de Bayes sont l'exemple typique) cherchent les règles de décision les mieux adaptées au modèle considéré.
Lorsqu'on ne fait qu'admettre l'existence des structures distributionnelles des descripteurs, on utilise une approche statistique [HAN 81]. Les règles de classement sont alors construites à partir d'échantillons supposés représentatifs. Il existe de nombreuses méthodes dont l'emploi varie suivant les hypothèses et les situations initiales [CAR & al. 96]. Citons par exemple, parmi les approches paramétriques, le modèle bayésien qui nécessite comme hypothèse la connaissance de la forme générale de la distribution de probabilité des exemples conditionnellement à leur classe d'appartenance. La discrimination logistique est une méthode semi-paramétrique basée sur l'estimation des probabilités a posteriori des classes qui est une hypothèse moins forte que celle utilisée dans le modèle bayésien. Les approches non paramétriques ne formulent pas d'hypothèse sur les distributions a priori des descripteurs. La seule information disponible est l'ensemble d'apprentissage. Certaines de ces approches (noyaux de Parzen, plus proches voisins) cherchent à interpoler localement la distribution des exemples dans l'espace des descripteurs, d'autres comme la segmentation par arbre (méthode que l'on retrouve en intelligence artificielle comme on le verra au paragraphe suivant) cherchent une interpolation locale de la fonction de classement. L'analyse factorielle discriminante (AFD), méthode d'analyse multivariée [ROB 89], est l'une des plus populaires pour la discrimination. L'AFD recherche les axes factoriels séparant au mieux les classes et l'équation de l'axe discriminant permet le calcul d'un score qui résume le mieux l'ensemble des descripteurs discriminants. Notons que l'équivalence formelle entre l'AFD et le perceptron multicouches linéaire a été établie [GAL & al. 88]. Dans le cas d'unités à fonction de transfert non linéaire (fonctions sigmoïde), l'équivalence évoquée ci-dessus n'est pas démontrée.

2.3. Programmation linéaire et classification

Les techniques de programmation linéaire ont été employées pour optimiser des critères de construction de classifieurs [MAN & al. 90; BEN 92] et entre autres pour l'obtention de

combinaisons linéaires de descripteurs formant des nœuds d'arbre de décision (cette dernière méthode est détaillée au paragraphe suivant).
Mais la programmation linéaire peut aussi être utilisée directement pour la classification. En effet, si on considère deux classes d'exemples, la programmation linéaire fournit les hyperplans séparant les enveloppes convexes qui rassemblent les individus d'une même classe. Si ces enveloppes sont disjointes, elles permettent de classer correctement les exemples de la base d'apprentissage. Si elles ne sont pas disjointes, cela signifie que tous les exemples ne sont pas séparables uniquement à l'aide des descripteurs utilisés. Dans ce cas, la méthode met en évidence les exemples posant problème. Cette stratégie a comme avantage d'indiquer la participation exacte des exemples : exemples déterminants (ceux dont le retrait modifierait les critères appris), exemples non-déterminants, exemples aberrants (ceux qui appartiennent à une classe sans en satisfaire les contraintes). Par contre, elle implique une levée explicite des ambiguïtés (pour séparer deux enveloppes, l'utilisateur doit enlever ou réaffecter un exemple "mal classé").
Michel [MIC 96] propose d'utiliser cette stratégie pour l'élaboration d'un classifieur effectuant une partition de l'espace des exemples par raffinements successifs. L'hyperplan séparant au mieux les exemples est d'abord déterminé, puis ce processus est appliqué récursivement sur chaque demi-espace obtenu. Chaque hyperplan est défini par une combinaison linéaire des descripteurs. Du point de vue de la structure obtenue du classifieur, on notera l'analogie entre cette méthode et celle de production d'arbres obliques citée au paragraphe suivant (le processus d'obtention des hyperplans étant cependant différent).

2.4. Systèmes symboliques

Par rapport aux approches précédentes, les systèmes symboliques [MIS & al. 86] se distinguent par l'importance accordée à la représentation des connaissances. Un objet peut être structuré et est donc plus complexe à manipuler qu'un vecteur numérique. La connaissance acquise, exprimée sous une forme symbolique, doit être directement compréhensible par un utilisateur humain. Ces systèmes s'illustrent aussi parfois par leur fonction d'affectation d'un objet. Celle-ci peut exploiter la notion de contraste (un concept se définit non seulement par ses caractéristiques propres mais également par contraste avec les autres concepts de la partition) et celle de typicité (un exemple typique d'un concept a beaucoup de points communs avec ceux appartenant au même concept et peu avec ceux des autres concepts).
Les classifieurs symboliques les plus classiques expriment la connaissance acquise sous la forme de clauses de Horn (induction logique), de règles ou encore d'arbres de décision.
L'induction logique [MUG 92] sépare les connaissances sur les classes de celles sur les descripteurs. L'apprentissage se fait par induction en recherchant la généralisation la plus petite possible pour les classes. Le mécanisme mis en jeu est similaire à celui de la recherche du plus grand unificateur par le mécanisme du langage de programmation Prolog. GOLEM est l'un des systèmes typiques de cette famille.
Des systèmes comme GREEDY3 [PAG & al. 90] ou CN2 [CLA & al. 91] produisent un ensemble de règles. Ils cherchent une combinaison des valeurs des descripteurs pour prédire une des classes. Pour cela, ils utilisent une mesure de l'information (comme le gain d'information qui est aussi couramment utilisé pour les arbres de décision) pour sélectionner la règle la plus pertinente. Les exemples vérifiés par cette règle sont alors écartés et le processus est réitéré sur les exemples restants. Il existe d'autres approches comme celle fondée sur la "rough sets theory" [PAW & al. 95]. Cette stratégie approxime le concept à l'aide d'un paramètre de similarité défini par l'utilisateur et produit des règles avec un degré d'appartenance [QUA & al. 97].
Bien qu'ayant des limitations conceptuelles bien connues, la construction d'arbres de décision est une des méthodes les plus utilisées en pratique, certainement parce qu'elle est aisée à mettre en œuvre et que l'architecture du classifieur obtenu a un fort pouvoir explicatif. Rappelons succinctement le principe de production de ces arbres [BRE & al. 84; QUI 86]. A partir des exemples de la base d'apprentissage, un critère de sélection détermine le descripteur qui sépare le mieux les exemples par rapport à la classe. Ce descripteur est alors choisi et les exemples sont partitionés en sous-ensembles suivant ses valeurs. Ce processus est appliqué récursivement sur chaque sous-ensemble jusqu'à ce que ceux-ci ne comportent plus que des

exemples d'une même classe ou qu'il ne reste plus de descripteurs. On aboutit ainsi à un arbre dont les nœuds sont les sous-ensembles successifs segmentés par un descripteur, les feuilles sont les sous-ensembles terminaux non divisés et les branches spécifient une valeur d'un descripteur. Un arbre de décision fournit une description efficace des exemples de la base, cette description étant orientée par un classement a priori des exemples. Il peut aussi être utilisé pour classer de nouveaux exemples, le classement d'un exemple se fait en parcourant un chemin qui part de la racine pour aboutir à une feuille. On attribue alors à l'exemple la classe la plus fréquente de la feuille. Lorsque les descripteurs sont numériques, une des limitations bien connue des arbres de décision est d'effectuer des séparations parallèles aux axes caractérisant les individus. Aussi, les techniques construisant des arbres de décision obliques (c'est-à-dire des arbres dont les axes de séparation sont des combinaisons linéaires des descripteurs) connaissent aujourd'hui un regain d'intérêt [MUR & al. 94; BRO & al. 95].

2.5. Conclusion

Les différentes méthodes que nous venons de développer sont souvent présentées indépendamment de l'utilisateur. Il est vrai que d'un point de vue conceptuel le résultat est directement obtenu à partir de la base d'apprentissage et elles méritent à ce titre le qualificatif "d'automatique". Cependant, leur utilisation concrète nécessite la réalisation d'un certain nombre de tâches. En ce qui concerne les arbres de décision qui vont être le support de nos propos pour la suite de cet article, on peut citer par exemple le recueil des données pour la conception de la base d'apprentissage, le recodage de certains descripteurs, le traitement spécifique d'individus, l'analyse de l'arbre obtenu afin de savoir si celui-ci peut être amélioré à la fois du point de vue de l'explicitation et de celui de la minimisation du nombre de mal classés sur un échantillon test. Tsatsarakis & Sleeman [TSA & al. 93] mettent en évidence certaines de ces tâches et proposent un environnement ergonomique nommé WILA (Workbench for Inductive Learning Algorithms) pour faciliter le déroulement de celles-ci. Le paragraphe suivant montre que non seulement ces tâches sont étroitement liées à l'utilisateur mais aussi qu'elles sont fondamentales car elles influent directement sur le résultat du travail d'induction.
On notera aussi que la plupart des méthodes cherchent à rassembler les objets similaires. Une autre approche est au contraire de travailler sur les différences qui singularisent les objets de classes distinctes. Le système ANADIA [BEU & al. 96] aide l'utilisateur à catégoriser les objets à partir de leurs différences. Le modèle d'ANADIA provient de théories systémiques sur la mémoire vue comme processus de catégorisation.

3. Le rôle de l'utilisateur dans la production d'arbres de décision

Ce paragraphe rappelle d'abord brièvement les deux étapes (construction de l'arbre et élagage de celui-ci) classiquement distinguées dans la littérature. Nous définissons ensuite un indice de qualité d'un arbre et nous montrons en quoi cet indice conduit à une méthode d'élagage plus particulièrement adaptée aux domaines incertains et permettant de mettre en évidence les parties les plus fiables de l'arbre. Cet indice nous permet d'illustrer graphiquement un des rôles de l'utilisateur lors de la construction d'arbres de décision.

3.1. Construction et élagage des arbres de décision

Le processus de production d'arbre de décision comporte deux étapes directement liées au système d'apprentissage : d'une part sa construction puis son élagage [BRE & al. 84; BUN & al. 92]. Pour la construction, il existe plusieurs critères de sélection d'un descripteur. Mais on sait que d'un point de vue théorique, les critères dérivés d'une mesure d'impureté possèdent de bonnes propriétés pour l'étude de problèmes réels et fournissent des résultats comparables [FAY & al. 92; BRE & al. 84; CRE & al. 97]. Nous appelons de tels critères des critères C.M. (concave-maximum) car une mesure d'impureté, entre autres caractéristiques, est définie à l'aide d'une fonction concave.
En médecine, comme dans d'autres domaines qualifiés "d'incertains", d'une part les mêmes causes ne produisent pas toujours les mêmes effets et d'autre part on ne possède qu'une vision

partielle des descripteurs expliquant la question étudiée. Il est alors impossible de construire un arbre dont les exemples de chaque feuille appartiennent à la même classe. Dans de telles situations, on constate que les arbres produits tendent à être complexes et surchargés et un phénomène de "sur-spécialisation", bien connu en apprentissage, apparaît. Certaines divisions, généralement au bas des arbres, sont dues au hasard. L'arbre s'est adapté aux particularités de la base d'apprentissage au lieu de refléter une véritable connaissance du domaine [BRE & al. 84; SCH 93a; CRE & al. 96]. Cette complexité injustifiée augmente le taux d'erreur de classement de nouveaux cas. Ces branches trop spécifiques ne doivent pas être construites ou doivent être élaguées. On sait qu'il vaut mieux générer l'arbre complet puis couper les branches non pertinentes plutôt qu'essayer de se définir des critères pragmatiques stoppant l'arbre au cours de sa construction [BRE & al. 84; GEL & al. 91].
Il existe de nombreuses méthodes d'élagages et pour plus de détails on se référera par exemple à [QUI 87; MIN 89; ESP & al. 93]. La plupart de ces méthodes sont basées sur la minimisation d'un taux de mal-classés estimé à partir d'un échantillon test ou avec des techniques statistiques comme la cross-validation ou le bootstrap. Ces méthodes sont justifiées si on veut transformer un arbre en classifieur mais elles sont inadéquates à certains usages des arbres de décision dans les domaines incertains. Le prochain paragraphe présente une méthode d'élagage appelée élagage C.M. car un critère C.M. doit être utilisé pour construire l'arbre et montre en quoi cette méthode est adaptée aux domaines incertains comme par exemple la médecine d'où sont tirés les exemples du paragraphe 3.4.

3.2. Indice de qualité et élagage C.M.

L'élagage C.M. repose sur le calcul de l'indice de qualité d'un arbre, cet indice est défini à partir du critère de sélection d'un descripteur. En effet, la valeur du critère de sélection sur un nœud indique avec quelle fiabilité le descripteur a été choisi. Mingers [MIN 87] propose une méthode d'élagage qui évalue la pertinence d'un nœud à partir de la valeur du critère de sélection. Malheureusement, cette méthode nécessite la définition de seuils dépendants de la base étudiée. Un autre inconvénient est qu'elle ne prend pas en compte le nombre d'exemples du sous-arbre à couper. La valeur du critère de sélection, en indiquant la qualité de la division d'un nœud, permet de comparer deux sous-arbres de profondeur 1. Mais cette valeur ne peut pas être retenue pour évaluer la qualité d'un sous-arbre de profondeur supérieure à 1 car elle correspond uniquement à la première division. On montre [CRE & al. 96], qu'il est possible de définir un indice de qualité d'un arbre qui généralise cette approche. Appelons T un arbre et I cet indice de qualité. I (T) est liée à chacune des feuilles de T de telle sorte que les différentes mesures de qualité de toutes les segmentations soient prises en compte.
On montre que [CRE & al. 96] :
- $0 \leq I(T) \leq 1$
- I (T) = 1 si et seulement si toutes les feuilles sont pures (i.e. : tous les exemples de chaque feuilles appartiennent à la même classe) et I (T) = 0 si et seulement si les lois de probabilités de la classe sur la racine et sur toutes les feuilles sont identiques.

I (T) mesure la différence entre l'impureté de la racine et l'impureté moyenne des feuilles, cette différence étant normalisée entre [0, 1]. En d'autres termes, I (T) peut être vue comme l'impureté expliquée ou éliminée par T.
L'élagage C.M. d'un arbre T produit une séquence d'arbres élagués imbriqués (à partir de T, chaque arbre élagué s'obtient en enlevant un sous-arbre de l'arbre précédent dans la séquence). A chaque étape, l'élagage C.M. coupe le sous-arbre qui entraîne la plus petite diminution de l'indice de qualité. Ce processus est, a priori, répété jusqu'à atteindre la racine. Nous verrons au paragraphe 3.4 que la courbe de l'indice de qualité en fonction du nombre de sous-arbres coupés définit une moyen pragmatique de stopper l'élagage.

Nous pensons que l'élagage C.M. est particulièrement approprié à l'induction incertaine car il permet de conserver des sous-arbres qui, même s'ils n'améliorent pas ou peu le taux de mal classés, possèdent des feuilles mettant en évidence des populations rares et intéressantes. Autrement dit, l'élagage C.M. n'élimine pas systématiquement un sous-arbre dont le taux de mal-classés sur les feuilles est égal à celui de la racine. Considérons par exemple le sous-arbre représenté à la figure 1. La classe a deux valeurs et pour chaque nœud, le premier (resp.

deuxième) chiffre indique le nombre d'exemples ayant la première (resp. deuxième) valeur de la classe. Ce sous-arbre ne fait pas diminuer le nombre d'erreurs, mais il met en évidence une population spécifique où la valeur de la classe est certaine (tout au moins sur l'échantillon d'apprentissage) et une autre population où il est impossible de prédire la valeur de la classe. Ce sous-arbre est, pour nous, plus riche que sa racine, surtout si les feuilles obtenues sont intéressantes au plan pratique, comme par exemple, si elles définissent une population de sujets à risque. L'indice de qualité de ce sous-arbre est 0,55 ce qui signifie qu'il explique 55 % de l'impureté initiale.

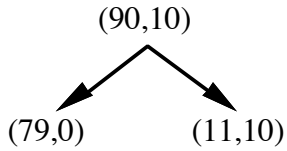

Figure 1. Les deux feuilles n'améliorent pas le classement des exemples (si on classe tous les exemples d'un nœud dans la classe la plus fréquente), mais la répartition des exemples suivant les valeurs de la classe est très différente sur les 3 nœuds.

Tel que nous l'avons défini, l'indice de qualité permet de comparer objectivement des arbres issus de fichiers différents et relatifs à un même domaine. Au paragraphe 3.4, nous présentons des exemples où l'évolution de cet indice illustre les liens personne-système au cours du développement d'arbres de décision.

3.3 Relations entre utilisateurs et système de développement d'arbres de décision

Nous indiquons dans ce paragraphe à quels moments et sous quelles formes les utilisateurs interviennent dans le processus de développement d'arbres de décision.

3.3.1. Les différentes phases

Dans la pratique, on distingue trois phases dans le travail de production d'arbres de décision : la phase de préparation des données, celle de construction d'arbres et celle d'élagage. On rajoute parfois une quatrième phase qui a pour but d'étudier le classement de nouveaux cas sur

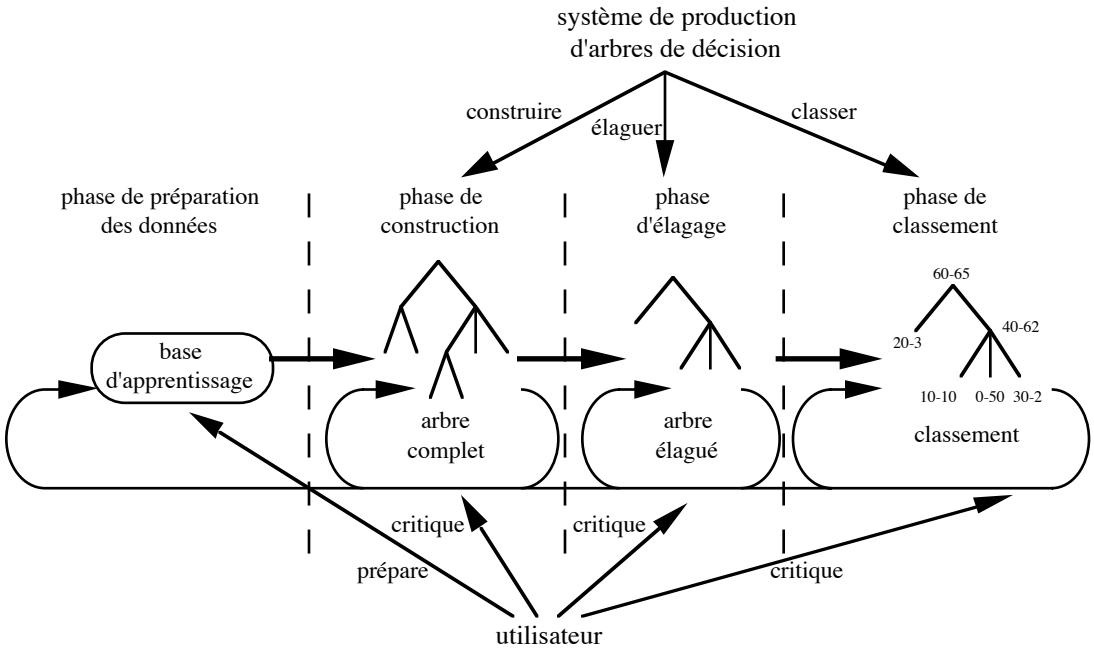

Figure 2. Les différentes phases lors de la production d'arbres de décision et leurs relations avec l'utilisateur.

un arbre induit et éventuellement élagué. L'utilisateur intervient de façon prépondérante durant la première phase, mais il a aussi un rôle de superviseur lors de l'ensemble des phases et plus particulièrement de critique après les deuxième et troisième phases (figure 2). Nous ne détaillons pas ici la quatrième phase qui est marginale du point de vue du rôle de l'utilisateur.

3.3.1.1. Phase de préparation des données

Le but de cette étape est de fournir, à partir de la base de données rassemblant les exemples sous leur forme brute, une base d'apprentissage la plus adaptée possible au développement d'arbres de décision. C'est la phase où l'utilisateur intervient le plus directement. Ses tâches sont nombreuses : élimination des individus jugés aberrants et/ou comportant trop de valeurs manquantes, élimination de descripteurs évalués non pertinents pour la question posée, recodage de valeurs de descripteurs (on sait en effet que si les descripteurs ont des nombres très différents de valeurs, ceux ayant le plus de valeurs tendent à être choisis en premier [WHI & al. 94; KON 95]), recodage de plusieurs descripteurs (par exemple, des fusions de descripteurs), segmentation éventuelle des descripteurs continus, traitement des données manquantes,…
Revenons sur certaines de ces tâches. A l'origine [HUN 66], les algorithmes de production d'arbres de décision n'acceptaient que des descripteurs qualitatifs, les descripteurs continus devant être au préalable discrétisés. Cette segmentation initiale peut être effectuée en demandant aux experts de fixer des seuils ou en utilisant stratégie reposant sur une fonction d'impureté [FAY & al. 93]. La segmentation peut aussi être réalisée au cours de la construction de l'arbre comme c'est le cas avec le logiciel C4.5 [QUI 96]. Une variable continue peut alors être segmentée plusieurs fois dans un même arbre. Il nous semble important que l'utilisateur puisse activement intervenir dans ce processus en indiquant par exemple une discrétisation a priori des descripteurs pour lesquels cela a un sens et en laissant agir le système pour les autres. Il est à noter que si on sait de façon raisonnable transformer un descripteur continu en binaire, la question est beaucoup plus délicate si on souhaite que le descripteur discrétisé ait au moins trois valeurs.
C'est également l'utilisateur qui généralement décide de supprimer, recoder et fusionner des descripteurs. Il possède des idées a priori permettant une première passe dans cette tâche. Mais nous verrons que la construction d'arbres, en explicitant le phénomène sous-jacent étudié, lui suggère de nouveaux recodages et/ou fusions de descripteurs entraînant souvent un niveau de description plus général.
Les algorithmes actuels de construction d'arbres de décision traitent la plupart du temps les valeurs manquantes par des méthodes où l'utilisateur n'intervient pas [BRE & al. 84; QUI 93]. Ragel [RAG 97] propose au contraire un traitement préalable de la base reposant sur la recherche de régularités. A partir de celles-ci, des règles incertaines déterminant les valeurs manquantes sont déduites. Une telle méthode laisse une place à l'utilisateur et à son savoir pour supprimer, ajouter ou modifier des règles.
Comme on le voit, cette phase dépend en fait étroitement du travail de l'utilisateur.

3.3.1.2. Phase de construction d'arbres

Le but de cette phase est d'induire un arbre à partir de la base d'apprentissage élaborée lors de l'étape précédente. Des paramètres du système sont souvent à spécifier. En induction incertaine, il est inutile de poursuivre la construction de l'arbre à partir d'un nœud ayant trop peu d'exemples, cette quantité étant relative au nombre initial d'exemples de la base. Aussi, un paramètre important à fixer est le nombre minimum d'exemples nécessaires pour segmenter un nœud. Devant un arbre particulièrement touffu, l'utilisateur va demander la construction d'un nouvel arbre en fixant une valeur plus élevée à ce paramètre, ce qui revient à élaguer l'arbre suivant une procédure pragmatique. Nous avons vu qu'en induction incertaine, on choisira un critère C.M. Mais si l'utilisateur sait que le phénomène étudié admet des causes déterministes dans des situations avec peu d'exemples, il sera peut-être préférable d'opter pour un critère de sélection d'un descripteur comme le critère ORT [FAY & al. 92] pour obtenir une description plus concise de ces situations.
La critique de l'arbre obtenu est la participation la plus importante de l'utilisateur au cours de cette phase. Celui-ci regarde si l'arbre s'interprète par rapport à sa connaissance du domaine, si sa structure générale est conforme à ses attentes. Face à un résultat surprenant, il s'interroge pour savoir si celui-ci est dû à un biais de la base d'apprentissage ou reflète un phénomène,

parfois soupçonné, mais pas encore explicitement énoncé. La plupart du temps, la vision de l'arbre donnant une nouvelle idée des descripteurs, l'utilisateur va décider de refaire un arbre en ayant retravaillé la base d'apprentissage et/ou en changeant un paramètre du système d'induction pour confirmer ou infirmer une conjecture.

3.3.1.3. Phase d'élagage

Nous nous plaçons ici dans le contexte de l'induction incertaine et nous utilisons l'élagage C.M. présenté au paragraphe 3.2 (les techniques d'élagage basées sur la minimisation d'un taux de mal classés produisent directement un arbre et seul ce dernier est soumis à la critique de l'utilisateur).
Avec l'élagage C.M., l'utilisateur dispose de plus d'informations pour réagir. D'abord, il connaît l'indice de qualité de l'arbre complet, ce qui lui permet d'évaluer la difficulté globale du problème. Si cet indice est faible, cela signifie que le problème est délicat ou pas suffisamment décrit, que la base d'apprentissage n'est pas représentative, ou encore que la méthode des arbres de décision s'adapte mal à celui-ci. Si l'utilisateur dispose de plusieurs arbres, l'indice de qualité permet de les comparer objectivement et éventuellement de suggérer de nouvelles expériences. En suivant l'évolution de cet indice au cours de l'élagage C.M, l'utilisateur distingue alors les parties de l'arbre dues au hasard de celles fiables. Ce travail fait ressortir les descripteurs les plus pertinents de ceux qu'il est peut-être nécessaire de redéfinir. Des populations où la classe est facile à déterminer se détachent d'ensembles d'individus où il est impossible de la prédire. De telles zones peuvent suggérer de nouvelles expériences sur des populations plus réduites ou encore interroger sur l'existence de nouveaux descripteurs pour aider à déterminer la classe pour les individus où cela n'est pas encore possible.
Enfin, tout comme avec les techniques basées sur la minimisation d'un taux de mal classés, l'utilisateur peut confronter un arbre élagué mis en évidence par l'élagage C.M. avec celui obtenu par un élagage basé sur ses connaissance, et évaluer a priori les capacités prédictives de l'arbre.

3.3.2. Conclusion

Au cours de ce paragraphe, nous avons vu que les interventions de l'utilisateur sont multiples et que l'étude des résultats suscite de nouvelles expériences. L'utilisateur recommence ainsi plusieurs fois le travail effectué au cours d'une phase en changeant des paramètres ou revient à des phases antérieures (les flèches tracées à la figure 2 indiquent l'ensemble des relations entre les différentes phases). Nous pensons qu'il est nécessaire que l'utilisateur soit partie prenante du système pour qu'un véritable cycle de développement se mette en place. Ce dernier nous semble fondamental pour aboutir à des arbres utiles et satisfaisants. L'utilisateur ne sait généralement pas à l'avance quel arbre est pertinent pour son problème et c'est d'ailleurs parce qu'il trouve enrichissante sa participation à cette recherche qu'il porte de l'intérêt au travail d'induction.

3.4. Etude expérimentale

A partir de l'étude de deux problèmes médicaux réels, ce paragraphe montre qu'il s'établit une relation entre le système et l'utilisateur et que cette relation se visualise avec les courbes de l'indice de qualité d'un arbre en fonction de ses sous-arbres élagués. Nous utilisons ici le logiciel ARBRE [CRE 91] parce que celui-ci pratique l'élagage C.M.
Nous présentons d'abord brièvement les bases étudiées et résumons les expériences effectuées, puis nous discutons de l'interprétation et de l'intérêt de l'ensemble des arbres successivement produits.

3.4.1. Bases étudiées et expériences

3.4.1.1. Maladie de Hodgkin

La maladie de Hodgkin est une forme de cancer touchant essentiellement une population jeune. Les progrès médicaux ont permis d'offrir des traitements atteignant actuellement un taux de survie de 80 % si la maladie est soignée dès ses premiers stades. Une des voies actuelles de

recherche est l'optimisation des traitements : tout en conservant les taux de survie obtenus, il s'agit de minimiser les complications et les effets secondaires. Le but de cette étude est d'affiner des profils pronostiques de patients afin d'obtenir une meilleure adéquation entre le profil pronostique et le traitement mis en route.
La base de données utilisée est celle de l'Organisation Européenne de Traitement et de Recherche sur le Cancer. Elle est gérée par le Centre François Baclesse de Caen et comporte environ 3500 malades (pour les stades I et II). Dans un premier temps, nous avons étudié les malades pris en charge avec le protocole H7 (soit 832 patients) et nous avons cherché à distinguer les malades de profils pronostiques favorable de ceux de profil pronostique défavorable. Les descripteurs sont des signes cliniques, des indications sur la localisation et la taille des aires ganglionnaires envahies et des données cytochimiques et biologiques. 8 individus aberrants ont été éliminés et la base finalement traitée comporte 824 exemples. Signalons que plusieurs descripteurs comportent un nombre important de valeurs manquantes.
La première expérience (nommée HODarb1) utilise 27 descripteurs (seuls quelques descripteurs non significatifs pour ce protocole ou marginaux n'ont pas été pris en compte). Le tableau 1 indique les principales caractéristiques des expériences.

nom du fichier	nombre d'exemples	nombre de descripteurs	valeurs / descripteurs	classe
HODarb1	824	27	2-3-4	369-455
HODarb2	824	19	2-3-4	369-455

Tableau 1. Caractéristiques des fichiers utilisés dans le cadre de la maladie Hodgkin. Le nombre de descripteurs s'entend y compris la classe, "valeurs / descripteurs" indique le nombre de valeurs des descripteurs. Le premier (resp. deuxième) nombre de "classe" est le nombre d'individus de profil pronostique favorable (resp. défavorable).

Comme les descripteurs relatifs à l'envahissement des aires ganglionnaires ont beaucoup de valeurs manquantes et qu'il est possible d'en approcher certaines par recoupements de descripteurs, les experts ont décidé d'effectuer des fusions parmi ces descripteurs. Ainsi, pour la deuxième expérience (HODarb2), ces douze descripteurs ont été simplifiés en cinq nouveaux. Un descripteur jugé non pertinent a également été éliminé. Cette dernière expérience comporte donc huit descripteurs de moins que la première. On peut aussi s'attendre à ce que ce recodage entraîne une certaine perte d'information.

3.4.1.2. Maladie thrombo-embolique

La maladie thrombo-embolique (MTE) recouvre deux entités : la thrombose veineuse profonde (TVP) et/ou sa conséquence, l'embolie pulmonaire (EP) qui met en danger la vie du patient. Le diagnostic d'EP repose sur des arguments paracliniques plus ou moins complexes et il est impossible actuellement de dépister efficacement les EP minimes chez tous les sujets présentant une TVP. Autrement dit, nous savons qu'il n'est pas possible de construire un arbre classant de façon robuste une grande part des individus. Parmi les patients atteints d'une TVP, le but principal de cette étude est d'identifier des populations à haut-risque à qui l'on proposera un protocole de dépistage complet.
La base de données initiale comporte 1280 patients hospitalisés au Département d'Angiologie du Centre Hospitalier Universitaire de Grenoble. Les descripteurs sont des signes cliniques, les aspects phlébographiques et les indices rhéographiques de la thrombose et divers autres examens et traitements. D'emblée, les experts ont éliminé de l'étude les individus pour lesquels on ne dispose pas de manière fiable les indices rhéographiques ou dont le diagnostic d'embolie pulmonaire reste entaché d'un doute. La base ainsi obtenue comporte alors 1073 patients.
La première expérience (appelée MTEarb1) emploie seize descripteurs dont onze décrivent les aspects phlébographiques de la thrombose et deux sont les indices rhéographiques (IV pour indice de vidange et IDV pour indice de débit de vidange). Le tableau 2 donne les principales caractéristiques des expériences.
Les neuf descripteurs indiquant la localisation de la thrombose se répartissent à tous les niveaux du premier arbre construit et les experts ne sont pas toujours parvenus à dégager un sens médical pour certains sous-arbres. Aussi ils ont proposé de recommencer ce travail avec une description simplifiée de la localisation de la thrombose. Pour la deuxième expérience (MTEarb2), nous avons remplacé ces neuf descripteurs par un nouveau qui indique uniquement

la hauteur de la tête du thrombus. Il est à noter que ce recodage entraîne une certaine perte de l'information par rapport à la première expérience. L'arbre obtenu avec la deuxième expérience a nettement plus satisfaisait les experts, mais ceux-ci ont pensé qu'il était encore possible d'aller plus loin. La position dans l'arbre des descripteurs IV et IDV leur paraissaient peu cohérente aussi ils ont décidé de les remplacer par un nouveau descripteur nommé "syndrome obstructif". Ce recodage a été réalisé par un expert qui a jugé du caractère pathologique des descripteurs IV et IDV et qui a évalué, au niveau médical, si ils définissaient un syndrome obstructif ou pas. Cet expert n'a pas pu déterminer la présence ou l'absence d'un syndrome obstructif pour dix cas, aussi cette troisième expérience (MTEarb3) porte sur 1063 exemples. Enfin, toujours pour cette troisième expérience, les experts ont effectué une nouvelle segmentation du descripteur "âge" et ont recodé le descripteur précisant la notion d'atteinte bilatérale.

nom du fichier	nombre d'exemples	nombre de descripteurs	valeurs / descripteurs	classe
MTEarb1	1073	16	2-3-4	535-538
MTEarb2	1073	8	2-4	535-538
MTEarb3	1063	7	2-3-4	528-535

Tableau 2. Caractéristiques des fichiers utilisés pour l'étude de la maladie thrombo-embolique. Le nombre de descripteurs s'entend y compris la classe, "valeurs / descripteurs" indique le nombre de valeurs des descripteurs. Le premier (resp. deuxième) nombre de "classe" est le nombre d'individus ayant donné lieu à une embolie (resp. n'ayant pas donné lieu à une embolie).

3.4.2. Résultats et commentaires

La figure 3 (resp. 4) schématise pour chaque arbre relatif à la maladie de Hodgkin (resp. à la maladie thrombo-embolique) son indice de qualité en fonction du nombre de sous-arbres élagués. Pour chaque domaine étudié, afin de faciliter la comparaison de cet indice entre les arbres, nous avons choisi une échelle et un intervalle identiques pour l'indice.
On remarque d'abord que la valeur de l'indice de qualité est nettement supérieure sur le domaine de la maladie de Hodgkin que sur celui de la MTE. Cette constatation n'est pas surprenante compte tenu de la difficulté de ce dernier problème. Les experts ont été cependant vivement intéressés par cette étude pour une double raison. D'une part, ces arbres ont dégagé des feuilles pertinentes repérant des populations à haut risque embolique. D'autre part, ils ont permis de reconsidérer certaines notions médicales (comme l'importance des caractéristiques hémodynamiques de la thrombose), pas totalement nouvelles, mais pas toujours assez mises en lumière. On note aussi que l'indice de qualité de l'arbre complet passe de 13 % pour MTEarb1 à 8 % pour les deux autres expériences. Bien que les experts aient une nette préférence pour le troisième arbre, le recodage a entraîné, de manière relative, une baisse importante de cet indice.

La figure 3 et surtout la figure 4 traduisent une évolution de l'ossature générale de la courbe de l'indice de qualité au fur et à mesure des expériences. Pour MTEarb1, cet indice décroît de manière assez régulière. Pour MTEarb2, on constate l'apparition de paliers comportant une série de sous-arbres qu'il est possible de couper sans diminuer la valeur de l'indice. Ces paliers sont de plus en plus réduits à mesure que le nombre de sous-arbres élagués augmente. Pour MTEarb3, on peut immédiatement couper un nombre important de sous-arbres tout en gardant la valeur initiale de l'indice correspondant à l'arbre complet. Cette stabilité de l'indice indique que même en choisissant un seuil plus petit pour le nombre de cas minimum pour segmenter un nœud, l'arbre n'expliquerait pas un pourcentage plus important de l'impureté de la base d'exemples. En tolérant une baisse de 1% de cet indice, on peut élaguer 14 sous-arbres sur les 23 possibles (soit 61 %), avec une baisse de 3 %, on peut couper 19 sous-arbres (soit 83 %). Passé ce seuil, l'indice de qualité décroît alors rapidement. Ce phénomène est typique des arbres finaux jugés intéressants par les experts et, bien que moins marqué, on l'observe aussi sur les arbres issus de la maladie de Hodgkin.
On obtient ainsi des courbes caractéristiques comportant trois parties distinctes : l'indice est d'abord stable, puis baisse un peu avec l'apparition de paliers et enfin s'effondre rapidement. Cela signifie qu'à mesure qu'on avance dans le choix et la définition des descripteurs, le haut des arbres est de plus en plus pertinent aux experts et fait ressortir les combinaisons

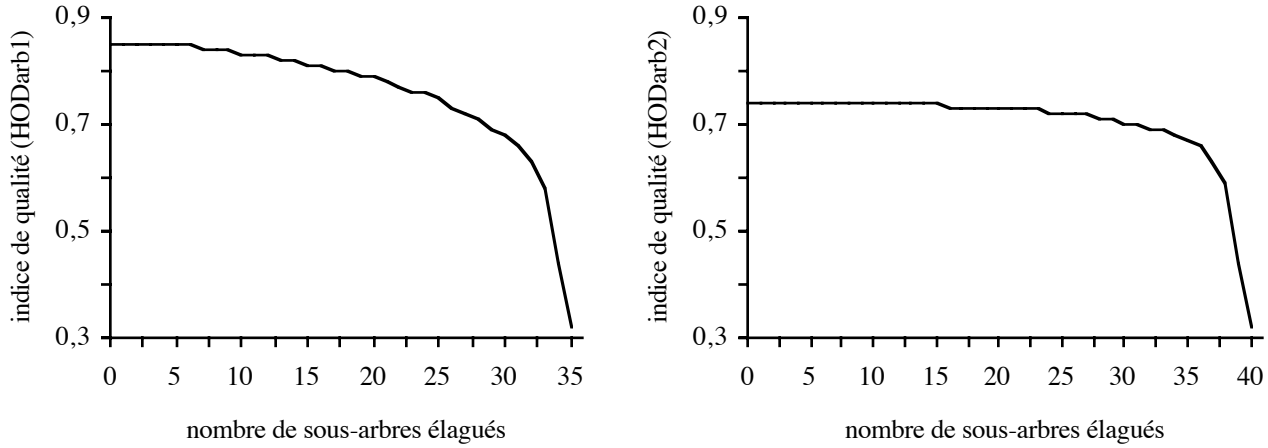

Figure 3. Pour chaque arbre dans le cas de la maladie de Hodgkin, évolution de l'indice de qualité en fonction du nombre de sous-arbres élagués.

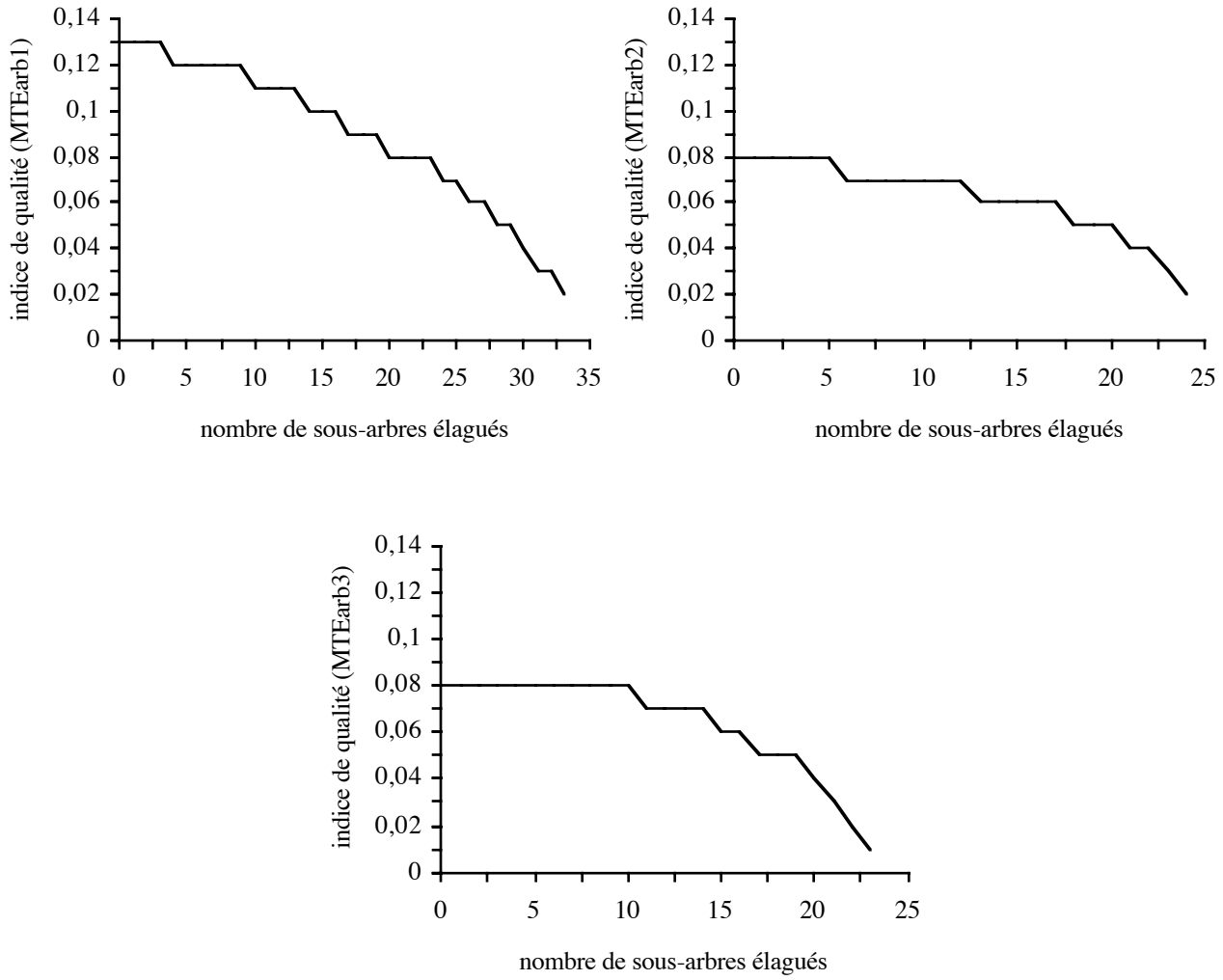

Figure 4. Pour chaque arbre dans le cas de la maladie thrombo-embolique, évolution de l'indice de qualité en fonction du nombre de sous-arbres élagués.

intéressantes des descripteurs. Parallèlement, on constate que le bas des arbres contient alors essentiellement des sous-arbres jugés non significatifs qui doivent être élagués. Il est clair que cette évolution est due aux échanges qui s'établissent entre l'utilisateur et le système d'apprentissage au cours du cycle de développement des arbres. Ces échanges permettent finalement d'aboutir à un arbre jugé plus satisfaisant par les experts.
Ces courbes fournissent aussi un moyen pragmatique pour obtenir un "bon" arbre élagué en indiquant le nombre de sous-arbres à couper pour obtenir le plus petit arbre ayant la valeur maximale (ou une valeur "élevée") de l'indice de qualité. Ces points sont particulièrement intéressants, car ils mettent en évidence de "bons" arbres élagués dans le sens où il s'agit d'arbres de taille minimum pour une part précise de l'impureté initiale expliquée.
Signalons enfin que les experts sont toujours impatients de voir les résultats proposés par l'élagage C.M. afin de les confronter avec les arbres élagués qu'ils ont eux-mêmes construits à partir de leurs connaissances du domaine.

4. Vers un atelier d'aide à la classification interactive

Le paragraphe précédent a montré que même pour une méthode qualifiée d'automatique comme les arbres de décision, l'utilisateur a un rôle prépondérant. On peut bien sûr s'interroger sur ce rôle et se demander quelle doit être sa place au cours du développement d'arbres de décision et plus généralement pour tout travail de classification.
En ce qui concerne les arbres de décision, rares sont les systèmes comme GID3* [FAY 94] qui vont vers une automatisation du processus, y compris au niveau du recodage de descripteurs. GID3* met en pratique un cas particulier de recodage au moment de la construction de l'arbre : à un nœud, pour chaque descripteur, GID3* détermine les valeurs du descripteur qu'il juge non pertinentes par rapport à ses autres valeurs. Toutes ces valeurs sont alors groupées en une seule nommée "valeur par défaut" qui sera utilisée pour segmenter le nœud si ce descripteur est choisi. La complexité de l'algorithme est linéaire en fonction du nombre de valeurs du descripteur.
La plupart des auteurs, au contraire, cherchent à définir des architectures logicielles intégrant explicitement l'utilisateur. Dans le contexte des graphes de décision (qui sont une généralisation des arbres de décision), le système SIPINA[1] permet à l'utilisateur de forcer le choix d'un descripteur, de regrouper temporairement des modalités, de stopper arbitrairement la construction à partir de certains nœuds,… Dabija & al. [DAB & al. 92] proposent une architecture (nommée KAISER, pour Knowledge Acquisition Inductive System driven by Explanatory Reasoning) pour une acquisition interactive de la connaissance à l'aide d'arbre de décision. Les experts peuvent incrémentalement ajouter de la connaissance relevant de la théorie du domaine. KAISER compare les arbres induits avec la théorie du domaine ce qui lui permet de détecter des incohérences (par exemple, la valeur du descripteur "œil" pour un chat doit être "ovale") [TSU & al. 96]. Keravhut & Potvin [KER & al. 96] ont défini un assistant pour collaborer avec l'utilisateur à la conception d'arbres de décision. Cet assistant, qui se présente sous la forme d'une interface graphique, aide l'utilisateur à tester les méthodes et leurs paramètres afin d'obtenir la combinaison la plus pertinente pour chaque problème étudié. CPLA (Conditional Probabilistic Learning Algorithm) de Hadjimichael & Wasilewska [HAD & al. 93] exploite un cycle où l'utilisateur examine à la fois les règles produites et des suggestions de modifications proposées par un module annexe nommé CSA (Condition Suggestion Algorithm). Ces suggestions ont pour premier but de généraliser les règles et de conduire à un plus petit ensemble de règles. Elles peuvent aussi introduire de nouveaux descripteurs ou modifier les coefficients de validité de règles. L'utilisateur peut aussi intervenir directement sur la définition de règles. Toute suggestion doit être validée par l'utilisateur et ce cycle se poursuit jusqu'à ce que l'utilisateur soit satisfait de l'ensemble final des règles générées. Remarquons que CPLA ne fournit pas un arbre de décision mais un ensemble de règles. Eklund & al. [EKL & al. 94] ont réalisé un environnement nommé DiagaiD pour l'acquisition de la connaissance entre autres à partir de bases de cas cliniques pour les systèmes experts médicaux. DiagaiD permet de générer des règles, de produire des interfaces graphiques et d'intégrer d'autres

---

1 SIPINA est téléchargeable à l'adresse : http://eric.univ-lyon2.fr/eric.html

applications. DiagaiD n'est pas spécifiquement destiné à être couplé avec un système de production d'arbres de décision.

Quant à nous, nous pensons que non seulement l'utilisateur a un rôle et doit intervenir lors de la classification mais aussi qu'on doit l'aider pour la réalisation des tâches annexes mais néanmoins indispensables. Certaines de ces tâches ont été mises en évidence à travers l'exemple de l'élaboration d'un arbre de décision mais elles ont un caractère générique à tout processus de classification. Citons par exemple :

- *pré-traitement des données en vue de leur amélioration pour leur exploitation par un classifieur standard* : traitement des données manquantes, segmentation si nécessaire de données continues, recodage de descripteurs, sélection de sous-ensembles d'exemples pertinents,…
- *évaluation de la qualité de l'apprentissage et du sur-apprentissage* : la meilleure réponse obtenue sur la base d'apprentissage tend à dépendre de particularités de celle-ci et ne se généralise pas toujours à d'autres bases du même domaine. Autrement dit, la phase de test ne conduit pas à de bonnes performances sur de nouveaux exemples. Ce phénomène nécessite l'étape d'élagage pour les arbres de décision. Plus généralement, l'utilisateur cherche à répondre à des questions du type [GAL & al. 96] : quel est le degré de généralité d'une règle inférée ? quelles garanties peut-on avoir sur ces règles ? quelle est la validité statistique de ces règles ?
- *choix et coopération de classifieurs* : comme nous l'avons vu au paragraphe 2, il existe de nombreux classifieurs et le choix de l'un d'entre eux, suivant le problème étudié, influe directement sur les résultats. Plutôt que de laisser uniquement l'utilisateur décider à partir de son savoir et de sa culture, on peut lui suggérer les classifieurs qui sont a priori les plus adaptés au domaine étudié. Cette recherche s'effectue à partir des caractéristiques propres au problème et aussi des objectifs de l'utilisateur (par exemple, attache t'il une grande importance au caractère explicatif de la connaissance produite). Bien évidemment, on peut aussi tirer profit des avantages spécifiques de différents classifieurs en les combinant. Par exemple, pour éviter la perte d'information due à la segmentation des données continues, celles-ci sont d'abord rassemblées sous la forme de composantes discriminantes avant d'être segmentées. Dans une approche où la classification est effectuée en parallèle par plusieurs systèmes [LU 96], l'étude par l'utilisateur des erreurs peut faire ressortir des profils d'exemples plus adaptés à tel classifieur et à aider à leur intégration.

Pour aller vers une démarche efficace de classification, nous pensons que celle-ci doit être interactive et doit prendre en compte l'ensemble des tâches énumérées ci-dessus. Nous travaillons sur la réalisation d'un atelier de classification interactive capable de traiter des connaissances et données incertaines et incomplètes qui sont des traits intrinsèques aux vrais problèmes réels. Nous demandons aussi à cet atelier d'offrir un environnement personnalisé à l'utilisateur. Cet environnement dépend du type de projet, de la nature des données, du profil spécifique de l'utilisateur qui lui-même peut être acquis et raffiné par apprentissage. Cet atelier comporte plusieurs sous-systèmes reprenant les tâches décrites ci-dessus :

- *sous-système de pré-traitement des données* dédié à leur amélioration en vue d'une certaine problématique.
- *sous-système d'exploration* composé de méthodes d'analyse de données ou d'intelligence artificielle au fort pouvoir explicatif. L'intérêt de ce sous-système est dans ses capacités à découvrir des connaissances implicitement contenues dans les données qui apportent un nouvel éclairage du phénomène étudié et contribuent donc à l'aide au choix d'un classifieur.
- *sous-système de classification* comportant les classifieurs les plus usuels avec leurs conditions d'applications mais aussi une aide au choix de leurs paramètres et de leurs particularités d'utilisation. Comme ont noté les acteurs du projet StatLog [MIC & al. 94], dans la pratique, le choix des paramètres pour lancer un classifieur nécessite souvent un "expert" de celui-ci.
- *sous-système d'évaluation de la qualité de l'apprentissage* et du risque de sur-apprentissage.
- *sous-système de validation de la connaissance acquise* permettant de confirmer ou infirmer une conjecture.

• *sous-système d'aide au choix et de combinaison de méthodes*. Celui-ci s'oriente autour de plusieurs axes :

- comme nous l'avons vu, l'étude des caractéristiques des exemples explorés fournit des pistes pour indiquer les classifieurs les plus appropriés. Par exemple, s'il existe une hiérarchie entre les valeurs de la classe, on peut suggérer d'employer un classifieur qui tient compte de cette spécificité.
- prise en compte des objectifs de l'utilisateur : attache t'il de l'importance à l'explication, au taux de bien classés, à la fiabilité des résultats,…
- intégration de la théorie du domaine du problème étudié [TAN & al. 95] et de toutes les connaissances que l'utilisateur souhaite apporter. Par exemple, la segmentation de descripteurs continus est mixée entre des seuils fixés par l'utilisateur et des méthodes automatiques.
- coopération de classifieurs. Outre la combinaison de méthodes afin d'associer leurs forces, on cherche aussi à tirer pleinement profit de différents paradigmes. A ce titre, nous nous intéressons aux propriétés de la classification reposant sur la similarité ou ressemblance qui est le paradigme le plus usuel, par rapport à la classification ontologique qui porte sur les propriétés caractéristiques des objets, à leur essence même et qui est plus particulièrement fondée sur leurs différences.

5. Conclusion

Les méthodes de classification sont souvent présentées comme "automatique" avec une participation marginale de l'utilisateur. A partir de l'exemple de la construction d'arbres de décision, nous nous sommes attachés à montrer que l'utilisateur a un rôle fondamental de critique et de superviseur et qu'il intervient de façon prépondérante. Cela conduit à ce qu'un véritable cycle de développement entre l'utilisateur et le système se mette en place. Celui-ci n'est possible que parce que la construction d'un arbre est quasi-immédiate. Au cours de ce cycle, l'ossature générale des arbres produits évolue et l'indice de qualité d'un arbre quantifie ce changement : au fur et à mesure que le processus avance, le haut des arbres est de plus en plus fiable et parlant aux experts alors que le bas est composé de sous-arbres jugés non pertinents à élaguer. Comme le montre l'évolution de l'architecture des arbres, cette relation conduit à de meilleurs arbres. L'expert trouve intellectuellement enrichissante sa participation à cette recherche d'un arbre pertinent pour son problème, et c'est pour cette raison qu'il porte de l'intérêt au travail d'induction.

Cette participation de l'utilisateur pour la préparation des données, le choix des paramètres, la critique des résultats est généralisable aux autres méthodes de classification. Nous pensons que pour aller vers une approche pertinente de la classification, il faut proposer à l'utilisateur un cadre qui l'associe explicitement et qui lui offre un ensemble d'outils non seulement de classification mais aussi des tâches associées afin qu'il puisse librement explorer les données, réagir, innover avec de nouvelles expériences. Pour cela, nous proposons un atelier de classification interactive prenant aussi bien en compte le pré-traitement des données, la qualité de l'apprentissage, l'aide au choix et la combinaison de classifieurs. Nous faisons ici l'hypothèse que la qualité du travail final ne dépend pas seulement des données mais aussi des interactions personne-système [ZRE 97]. Remarquons néanmoins que pour véritablement parler d'interactions il est nécessaire que l'atelier de classification interactive soit capable d'apprentissage.

Référence bibliographiques

[BEN 92] Bennet, K. P. (1992) Decision tree construction via linear programming. Computer Sciences Technical Report 1067. Madison. University of Wisconsin.

[BEU & al. 96] Beust, P., Delépine, L., Nicolle, A., & Coursil, J. (1996) ANADIA: a relevance examination tool for representations. *3rd Systems Science European Congress*. Rome.

[BRO & al. 95] Brodley, C. E., & Utgoff, P. E. (1995) Multivariate decision trees. *Machine Learning 19,* 45-77.

[BRE & al. 84] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984) *Classification and regression trees*. Wadsworth. Statistics probability series. Belmont.

[BRE 96] Breiman, L. (1996) Bagging predictors. *Machine Learning 24,* 123-140.

[BUN & al. 92] Buntine, W., & Niblett, T. (1992) A further comparison of splitting rules for decision-tree induction. *Machine Learning 8*, 75-85.

[CAR & al. 96] Caraux, G., & Lechevallier, Y. (1996) Règles de décision de Bayes et méthodes statistiques de discrimination. *Revue d'Intelligence Artificielle 10 (2/3),* 219-283.

[CLA & al. 91] Clark, P., & Boswell, R. (1991) Rule induction with CN2 : some recent improvements. *In proceedings of EWSL 91*. (pp. 151-163). Porto. Y. Kodratoff (Ed.). Lecture notes in artificial intelligence. N° 482. Springer-Verlag.

[CRE 91] Crémilleux, B. (1991) Induction automatique : aspects théoriques, le système ARBRE, applications en médecine. Thèse. Université Joseph Fourier. Grenoble (France).

[CRE & al. 96] Crémilleux, B., & Robert, C. (1996) A Pruning Method for Decision Trees in Uncertain Domains: Applications in Medicine. In proceedings of the workshop *Intelligent Data Analysis in Medicine and Pharmacology*, ECAI 96, Budapest, Hungary.

[CRE & al. 97] Crémilleux, B., & Robert, C. (1997) A Theoretical Framework for Decision Trees in Uncertain Domains: Application to Medical Data Sets. *6th Conference on Artificial Intelligence In Medicine Europe* (AIME 97).

[DAB & al. 92] Dabija, V.G., Tsujino, K., & Nishida, S. (1992) Theory formation in the decision trees domain. *Journal of Japanese Society for Artificial Intelligence, 7 (3),* 136-147

[EKL & al. 94] Eklund, P., Forsström, J., Holm, A., Nyström, M., & Selen, G. (1994) Rule generation as an alternative to knowledge acquisition: a systems architecture for medical informatics. *Fuzzy Sets and Systems, 66 (2),* 195-205.

[ESP & al. 93] Esposito, F., Malerba, D., & Semeraro, G. (1993) Decision tree pruning as search in the state space. *In proceedings of European Conference on Machine Learning ECML 93*. (pp 165-184). Vienna (Austria), P. B. Brazdil (Ed.). Lecture notes in artificial intelligence. N° 667. Springer-Verlag.

[FAY & al. 92] Fayyad, U. M., & Irani, K. B. (1992) The attribute selection problem in decision tree generation. *In proceedings of Tenth National Conference on Artificial Intelligence*. (pp 104-110). Cambridge, MA: AAAI Press/MIT Press.

[FAY & al. 93] Fayyad, U. M., & Irani, K. B. (1993) Multi-interval discretization of continuous-valued attributes for classification learning. *In proceedings of the Thirteenth International Joint Conference on Artificial Intelligence IJCAI 93*. (pp 1022-1027). Chambéry, France.

[FAY 94] Fayyad, U.M. (1994) Branching on attribute values in decision tree generation. *In proceedings of Twelfth National Conference on Artificial Intelligence*. (pp 601-606). AAAI Press/MIT Press.

[FIS 87] Fisher, D. (1987) Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2,* 139-172.

[FIS & al. 93] Fisher, D., & Hapanyengwi, G. (1993) Database management and analysis tools of machine induction. *Journal of Intelligent Information Systems 2,* 5-38.

[GAL & al. 88] Gallinari, P., Thiria, S., & S. Fogelman-Soulié F. (1988) Multilayer perceptron and data analysis. *In proceedings of IEEE ICNN'88* Vol. 1 (pp 391-401).

[GAL & al. 96] Gallinari, P., & Gascuel, O. (1996) Statistique, apprentissage et généralisation. Application aux réseaux de neurones. *Revue d'Intelligence Artificielle 10 (2/3),* 285-343.

[GEL & al. 91] Gelfand, S. B., Ravishankar, C. S., & Delp, E. J. (1991) An iterative growing and pruning algorithm for classification tree design. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13(2),* 163-174.

[HAD & al. 93] Hadjimichael, M., & Wasilewska, A. (1993) Interactive inductive learning. *International Journal of Man-Machine Studies 38,* 147-167.

[HAN 81] Hand, D. J. (1981) *Discrimination and classification*. John Wiley.

[HER & al. 91] Hertz, J., Krogh, A., & Palmer, R. G. (1991) *Introduction to the theory of neural computation*. Addison Welsey.

[HUN & al. 66] Hunt, E. B., Marin, J. & Stone, P. J. (1966) *Experiments in induction*. New York Academic Press.

[KER & al. 96] Kervahut, T., & Potvin, J. Y. (1996) An interactive-graphic environment for automatic generation of decision trees. *Decision Support Systems 18,* 117-134.

[KON 95] Kononenko, I. (1995) On biases in estimating mutli-valued attributes. *In proceedings of the Fourteenth International Joint Conference on Artificial Intelligence IJCAI 95*. (pp 1034-1040). Montréal, Québec, Canada.

[LEB 87] Lebowitz, M. (1987) Experiments with incremental concept formation: UNIMEM. *Machine Learning 2,* 103-138.

[LU 96] Lu, Y. (1996) Knowledge integration in a multiple classifier system. *Applied Intelligence 6,* 75-86.

[MAN & al. 90] Mangasarian, Setiono, & Wolberg, T. (1990) Pattern recognition via linear programming: theory and application to medical diagnosis. *SIAM Workshop on Optimization*.

[MIC & al. 86] Michalski, R. S., Carbonell, J., & Mitchell, T. (1986) *Machine learning: An artificial intelligence approach (Vol. 2)*. Los Altos, CA, Morgan Kaufmann.

[MIC & al. 94] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994) Machine learning, neural and statistical classification. Ellis Horwood Series in Artificial Intelligence.

[MIC 96] Michel, G. (1996) Classification et programmation linéaire. Application à la maladie de Hodgkin. Mémoire de DEA. Université de Caen. 1996.

[MIN 87] Mingers, J. (1987) Expert systems - rule induction with statistical data. *Journal of the Operational Research Society 38(1),* 39-47.

[MIN 89] Mingers, J. (1989) An empirical comparison of pruning methods for decision-tree induction. *Machine Learning 4,* 227-243.

[MUG 92] Muggleton, S. (1992) *Inductive logic programming*. Academic Press.

[MUR & al. 94] Murthy, S. K., Kasif, S., & Salzberg, S. (1994) A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research 2,* 1-32.

[PAG & al. 90] Pagallo, G., & Haussler, D. (1990) Boolean feature discovery in empirical learning. *Machine Learning 5(1),* 71-100.

[PAW & al. 95] Pawlak, Z., Gzymala-Busse, J., Slowinski, R., & Ziarko, W. (1995) Rough sets. *Communication of the ACM 38 (11),* 89-95.

[QUA 97] Quafafou, M., & Martienne, E. (1997) Classification et incrémentalité. Apprentissage par l'interaction. K. Zreik (Ed.). Europia.

[QUI 86] Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning 1,* 81-106.

[QUI 87] Quinlan, J. R. (1987) Simplifying decision trees. *International Journal of Man-Machine Studies 27,* 221-234.

[QUI 93] Quinlan, J. R. (1993) *C4.5 Programs for Machine Learning*. San Mateo, CA. Morgan Kaufmann.

[QUI 96] Quinlan, J. R. (1996) Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research 4,* 77-90.

[RAG 97] Ragel, A. (1997) Traitement des valeurs manquantes dans les arbres de décision. Rapport technique. Les cahiers du GREYC. Université de Caen.

[ROB 89] Robert, C. (1989) *Analyse descriptive multivariée. Application à l'intelligence artificielle*. Collection Statistiques en biologie et en médecine. Médecine-Sciences. Flammarion.

[RUM & al. 86] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986) Learning internal representations by error propagation. *Parallel Distributed Processing 1 (8)*.

[SCH 93a] Schaffer, C. (1993) Overfitting avoidance as bias. *Machine Learning 10,* 153-178.

[SCH 93b] Schaffer, C. (1993) Selecting a classification method by cross-validation. *Machine Learning 13,* 135-143.

[TAN & al. 95] Tanaka, M., Aoyama, N., Sugiura, A., & Koseki, Y. (1995) Integration of multiple knowledge representation for classification problems. *Artificial Intelligence in Engineering 9,* 243-251.

[TSA & al. 93] Tsatsarakis, C., & Sleeman, D. (1993) Supporting preprocessing and postprocessing for machine learning algorithms: a workbench for ID3. *Knowledge Acquisition 5 (4),* 367-384.

[TSU & al. 96] Tsujino, K., Dabija, V. G., & Nishida, S. (1996) Interactive improvement of decision trees through flaw analysis and interpretation. *Int. J. Human-Computer Studies 45,* 499-526.

[VIE & al. 92] Viennet, E., & Fogelmann-Soulié, F. (1992) Multi-resolution scene segmentation using MLPs. *In proceedings of IJCNN'92,* Artificial Neural Networks (pp 1599-1603). Brighton.

[WHI & al. 94] White, A. P., & Liu,W. Z. (1994) Bias in Information-Based Measures in Decision Tree Induction, *Machine. Learning,* 15 (3), 321-329.

[ZRE 97] Zreik, K. (1997) Apprentissage personne système. Apprentissage par l'interaction. K. Zreik (Ed.). Europia.