Arbres de Décision

Ricco RAKOTOMALALA
Laboratoire ERIC
Université Lumière Lyon 2
5, av. Mendés France
69676 BRON cedex
e-mail : rakotoma@univ-lyon2.fr

Résumé

Après avoir détaillé les points clés de la construction d'un arbre de décision à partir d'un petit exemple, nous présentons la méthode CHAID qui permet de répondre de manière cohérente à ces spécifications. Nous la mettons alors en œuvre en utilisant un logiciel gratuit téléchargeable sur Internet. Les opérations sont décrites à l'aide de plusieurs copies d'écrans. L'accent est mis sur la lecture et l'interprétation des résultats. Nous mettons en avant également l'aspect interactif, très séduisant, de la construction des arbres. De manière plus générale, nous essayons de mettre en perspective les nombreuses techniques d'induction d'arbres en faisant le bilan de l'état actuel de la recherche dans le domaine.

Mots-clés : Arbres de décision, segmentation, discrimination, apprentissage automatique

Abstract

In this paper, we show the key points of the induction of decision trees from a small dataset and we present the CHAID algorithm. Using a free software, the induction algorithm is detailed with several screenshots. We put emphasis on the interpretation of results and the interaction potentiality of the method. In a more general way, we try to give a comprehensive survey of the numerous variants which have been developed these last years.

Keywords: Decision Tree, Induction Tree, Supervised machine learning, Data mining

1 Introduction

La construction des arbres de décision à partir de données est une discipline déjà ancienne. Les statisticiens en attribuent la paternité à Morgan et Sonquist (1963) qui, les premiers, ont utilisé les arbres de régression dans un processus de prédiction et d'explication (AID – Automatic Interaction Detection). Il s'en est suivi toute une famille de méthodes, étendues jusqu'aux problèmes de discrimination et classement, qui s'appuyaient sur le même paradigme de la représentation par arbres (THAID -- Morgan et Messenger, 1973 ; CHAID -- Kass, 1980). On considère généralement que cette approche a connu son apogée avec la méthode CART (Classification and Regression Tree) de Breiman *et al*. (1984) décrite en détail dans une monographie qui fait encore référence aujourd'hui.

En apprentissage automatique, la plupart des travaux s'appuient sur la théorie de l'information. Il est d'usage de citer la méthode ID3 de Quinlan (Induction of Decision Tree – Quinlan 1979) qui, lui même, rattache ses travaux à ceux de Hunt (1962). Quinlan a été un acteur très actif dans la deuxième moitié des années 80 avec un grand nombre de publications où il propose un ensemble d'heuristiques pour améliorer le comportement de son système. Son approche a pris un tournant important dans les années 90 lorsqu'il présenta la méthode C4.5 qui est l'autre référence incontournable dès lors que l'on veut citer les arbres de décision (1993). Il existe bien une

autre évolution de cet algorithme, C5.0, mais étant implémentée dans un logiciel commercial, il n'est pas possible d'en avoir le détail.

En France, les travaux de Bourroche et Tenenhaus (1970) avec la méthode ELISEE est d'obédience statistique ; les travaux de Picard sur les pseudo-questionnaires (1972) sont à rapprocher de la théorie de l'information. On note surtout que de cette mouvance a émergé le concept de graphes latticiels (Terrenoire, 1970) qui a été popularisé par les graphes d'induction avec la méthode SIPINA (Zighed, 1992 ; Rakotomalala, 1997 ; Zighed et Rakotomalala, 2000).

Dans ce didacticiel, nous présentons les principes de construction des arbres de décision dans les problèmes de discrimination et classement : on veut expliquer et prédire la valeur (la classe, la modalité, l'étiquette) prise par une variable à prédire catégorielle, dite attribut classe ; à partir d'une série de variables, dites variables prédictives (descripteurs), discrètes ou continues. Selon la terminologie de l'apprentissage automatique, nous nous situons donc dans le cadre de l'apprentissage supervisé. Nous n'aborderons pas les autres types d'utilisation que sont les arbres de régression : il s'agit toujours d'un problème de prédiction mais la variable à prédire est continue (Torgo, 1999) ; et les arbres de classification, où l'objectif est de construire des groupes homogènes dans l'espace de descripteurs (Chavent *et al*., 1999).

Ce didacticiel est organisé comme suit. Dans la section suivante, à partir d'un tableau de 14 observations, nous décrivons les principales étapes de la construction d'un arbre de décision. La méthode CHAID est présentée dans la section 3, elle propose une réponse appropriée sur chacun des points mis en évidence précédemment. La section 4 est consacrée au traitement d'un ensemble de données « réalistes », le fichier IRIS de Fisher (1936), à l'aide du logiciel SIPINA, gratuit et accessible sur Internet. Chaque étape sera détaillée à l'aide de copies d'écran. Dans la section 5, nous faisons le point sur les avantages et inconvénients des arbres de décision. Nous tentons également d'élaborer une réflexion sur les avancées de la recherche dans le domaine. La section 6 correspond à la conclusion.

2 Un exemple introductif

2.1 Construire un arbre de décision

La popularité de la méthode repose en grande partie sur sa simplicité. Il s'agit de trouver un partitionnement des individus que l'on représente sous la forme d'un arbre de décision. L'objectif est de produire des groupes d'individus les plus homogènes possibles du point de vue de la variable à prédire. Il est d'usage de représenter la distribution empirique de l'attribut à prédire sur chaque sommet (nœud) de l'arbre.

Pour mieux appréhender la démarche, nous allons reprendre et dérouler un exemple qui est présenté dans l'ouvrage de Quinlan (1993). Le fichier est composé de 14 observations, il s'agit d'expliquer le comportement des individus par rapport à un jeu {jouer, ne pas jouer} à partir des prévisions météorologiques (Tableau 1).

Numéro	Ensoleillement	Température (°F)	Humidité (%)	Vent	Jouer
1	soleil	75	70	oui	oui
2	soleil	80	90	oui	non
3	soleil	85	85	non	non
4	soleil	72	95	non	non
5	soleil	69	70	non	oui
6	couvert	72	90	oui	oui
7	couvert	83	78	non	oui
8	couvert	64	65	oui	oui
9	couvert	81	75	non	oui
10	pluie	71	80	oui	non
11	pluie	65	70	oui	non
12	pluie	75	80	non	oui
13	pluie	68	80	non	oui
14	pluie	70	96	non	oui

Tableau 1 : Données "weather" (Quinlan, 1993)

L'arbre de décision correspondant est décrit ci-dessous (Figure 1).

- Le premier sommet est appelé la « racine » de l'arbre. Il est situé sur le premier niveau. Nous y observons la distribution de fréquence de la variable à prédire « Jouer ». Nous constatons qu'il y a bien 14 observations, dont 9 « oui » (ils vont jouer) et 5 « non ».
- La variable « ensoleillement » est la première variable utilisée ; on parle de variable de segmentation. Comme elle est composée de 3 modalités {soleil, couvert, pluie}, elle produit donc 3 sommets enfants.
- La première arête (la première branche), à gauche, sur le deuxième niveau, est produite à partir de la modalité « soleil » de la variable « ensoleillement ». Le sommet qui en résulte couvre 5 observations correspondant aux individus {1, 2, 3, 4, 5}, la distribution de fréquence nous indique qu'il y a 2 « jouer = oui » et 3 « jouer = non ».
- La seconde arête, au centre, correspond à la modalité « couvert » de la variable de segmentation « ensoleillement » ; le sommet correspondant couvre 4 observations, tous ont décidé de jouer (dans le tableau ce sont les individus n°6 à 9). Ce sommet n'ayant plus de sommets enfants, ce qui est normal puisqu'il est « pur » du point de vue de la variable à prédire, il n'y a pas de contre-exemples. On dit qu'il s'agit d'une feuille de l'arbre.

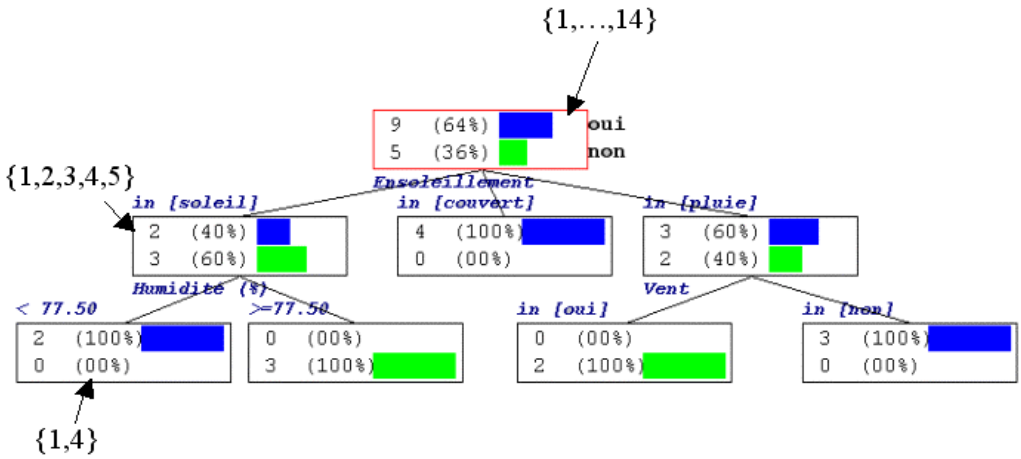

Figure 1 : Arbre de décision sur le fichier "weather"

- Reprenons le nœud le plus à gauche sur le deuxième niveau de l'arbre. Ce sommet, qui n'est pas pur, est segmenté à l'aide de la variable « humidité ». Comme le descripteur est continu, il a été nécessaire de définir un seuil dit de discrétisation qui permet de produire le meilleur partitionnement. Dans notre exemple, le seuil qui a été choisi est 77.5 %. Il a permis de produire deux feuilles complètement pures.
- Ce processus est réitéré sur chaque sommet de l'arbre jusqu'à l'obtention de feuilles pures. Il s'agit bien d'un arbre de partitionnement : un individu ne peut être situé dans deux feuilles différentes de l'arbre.
- Le modèle de prédiction peut être lu très facilement. On peut traduire un arbre en une base de règles sans altération de l'information. Le chemin menant d'un sommet vers la racine de l'arbre peut être traduit en une partie prémisse d'une règle de prédiction de type attribut-valeur « SI variable 1 = valeur 1 ET variable 2 = valeur 2 … ».
- Pour classer un nouvel individu, il suffit de l'injecter dans l'arbre, et de lui associer la conclusion attachée à la feuille dans laquelle il aboutit.

Cette simplicité apparente ne doit pas masquer des problèmes réels qui se posent lors de la construction de l'arbre. Nous allons les lister ci-dessous pour y apporter une réponse détaillée dans la section suivante.

1. La première question qui vient à l'esprit est le choix de la variable de segmentation sur un sommet. Pourquoi par exemple avons-nous choisi la variable « ensoleillement » à la racine de l'arbre ? Nous constatons également que le choix d'une variable de segmentation est relatif au sommet et non au niveau que nous sommes en train de traiter : les sommets à gauche et à droite du second niveau ont été segmentés avec des variables différentes. Il nous faut donc un indicateur (une mesure) qui permet d'évaluer objectivement la qualité d'une segmentation et ainsi de sélectionner le meilleur parmi les descripteurs candidats à la segmentation sur un sommet.
2. Pour mettre en œuvre la variable « humidité » au second niveau de l'arbre, nous avons été obligés de fixer un seuil (77.5%) pour segmenter le groupe d'observations. Comment a été fixé ce seuil ? Une fois que le seuil a été défini, comment sont mis en concurrence les variables continues et catégorielles pour la segmentation d'un sommet ?
3. L'objectif est de produire un partitionnement pur des observations de la base, ce qui est le cas de notre exemple. Que faire lorsque cela n'est pas possible ? De manière plus générale, est-ce qu'un partitionnement totalement pur est souhaitable sur le fichier de données ; est-ce qu'il est possible d'utiliser des règles plus efficaces pour définir la taille adéquate de l'arbre de décision ?
4. Enfin, si la prise de décision sur une feuille semble naturelle lorsqu'elle est pure, quelle est la règle de décision optimale lorsque qu'une feuille contient des représentants des différentes modalités de la variable à prédire ?

Répondre à ces questions permet de définir une méthode d'induction des arbres de décision à partir de données. La très grande majorité des méthodes recensées à ce jour respectent ce schéma, il est alors facile de les positionner les unes par rapport aux autres. On comprend également que le champ des stratégies possibles étant restreint, il paraît illusoire de trouver une avancée miraculeuse

sur un des 4 points ci-dessus qui permettrait de surclasser les techniques existantes. C'est pour cette raison que, si la communauté scientifique a été très prolixe dans les années 90 en explorant de manière quasi-exhaustive les variantes sur chacun de ces points, les comparaisons sur données réelles ont montré qu'elles produisaient des arbres avec des performances similaires. Des différences peuvent cependant apparaître dans des cas particuliers où telle ou telle caractéristique d'une variante que l'on a choisie s'avère mieux adaptée (voir par exemple Lerman et Da Costa pour les descripteurs à très grand nombre de catégories, 1996).

Il existe principalement trois méthodes référencées dans la communauté scientifique. Des didacticiels sur CART et C4.5 existant en très grand nombre par ailleurs (Nakache et Confais, 2003 ; Kohavi et Quinlan, 2002 ; Bardos, 2001; Zighed et Rakotomalala, 2000 ; Lebart et al., 2000 ; Gueguen, 1994 ; Celeux et Lechevallier, 1990), nous préférons dans cet article mettre l'accent sur une approche très largement inspirée de la méthode CHAID (CHi-squared Automatic Interaction Detection - Kass, 1980) qui a été une des premières à avoir été implémentée dans des logiciels commerciaux (SPSS Answer Tree et Knowledge Seeker). Elle a la particularité d'utiliser des formulations bien connues en statistique ; de plus elle est particulièrement efficace lorsque la taille de la base de données est importante.

3 Apprentissage d'un arbre de décision

3.1 Choix d'une variable de segmentation

Pour fixer les idées, nous nous plaçons sur la racine de l'arbre et nous mettons de côté le cas des variables continues « humidité » et « température ». Nous avons deux descripteurs candidats discrets. La quasi-totalité des méthodes d'induction d'arbres s'appuient sur le même procédé : pour chaque variable candidate, nous réalisons le partitionnement des observations et nous calculons un indicateur de qualité ; la variable retenue sera alors celle qui optimise cet indicateur. Les méthodes diffèrent selon la mesure utilisée.

Pour bien appréhender le procédé, il faut noter qu'une segmentation permet de définir un tableau de contingence croisant la variable à prédire et le descripteur candidat. Pour le cas de la variable « ensoleillement », on obtient le Tableau 2 à la racine de l'arbre.

NB Jouer	Ensoleillemer			
Jouer	couvert	pluie	soleil	Total
non	0	2	3	5
oui	4	3	2	9
Total	4	5	5	14

Tableau 2: Tri croisé à l'aide de la variable "ensoleillement" à la racine de l'arbre

Dans ce qui suit, nous adopterons les notations suivantes pour décrire les effectifs issus du croisement de l'attribut classe à K modalités et un descripteur à L modalités :

Y / X	x_1 $\quad x_l \quad$ x_L	Σ
y_1		
	$\vdots$	
y_k	$\cdots \quad n_{kl} \quad \cdots$	$n_{.l}$
	$\vdots$	
y_K		
Σ	$n_{k.}$	n

Tableau 3 : Tableau des effectifs lors du croisement de deux variables

Pour évaluer la pertinence de la variable dans la segmentation, CHAID propose d'utiliser le Khi-2 d'écart à l'indépendance, bien connu en statistique, dont la formule est la suivante :

$$\chi^2 = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{\left(n_{kl} - \frac{n_{k.} \times n_{.l}}{n} \right)^2}{\frac{n_{k.} \times n_{.l}}{n}}$$

Le critère du Khi-2 varie de 0 à +∞. Il n'est pas aisé de le manipuler car il avantage les descripteurs ayant un nombre élevé de modalités. Il est bien souvent préférable de le normaliser par le nombre de degrés de libertés, en prenant par exemple le t de Tschuprow dont le domaine de définition est [0 ; 1] ($t = \frac{\chi^2}{n\sqrt{(K-1)\times(L-1)}}$). Cette variante n'est pas proposée dans le descriptif originel de Kass (1980). Elle n'a aucun effet si les descripteurs comportent le même nombre de modalités, mais elle semble de bon sens dès lors que l'on traite des descripteurs très disparates.

	t de Tschuprow
Ensoleillement	0.3559
Vent	0.2582

Tableau 4 : Descripteurs discrets candidats sur la racine de l'arbre

Dans l'exemple, le calcul du t de Tschuprow sur les deux descripteurs candidats a produit les résultats repris dans le Tableau 4. Nous notons que la meilleure variable est bien « ensoleillement » avec un t de Tschuprow de 0.3559.

Ce processus est réitéré sur chaque sommet que l'on veut segmenter. S'il semble assez lourd au premier abord, il est facile à implémenter sur les ordinateurs, et surtout son temps d'exécution est raisonnable dans la pratique, même lorsque la base contient un nombre élevé de descripteurs. Cela n'est guère étonnant dans la mesure où la complexité de l'opération est linéaire par rapport au nombre d'individus et de variables. Ceci reste vrai tant qu'il est possible de faire tenir la totalité de la base de données en mémoire. Si ce n'est pas le cas, il s'avère nécessaire de parcourir toute la base sur le disque pour évaluer la pertinence de chaque descripteur. L'opération peut se révéler très lente. Des stratégies ont été proposées pour améliorer la rapidité du système face à de grandes bases de données, sans dégrader les performances (Catlett, 1991 ; Chauchat et Rakotomalala, 2000).

Il existe une quantité très grande de publications relatives à la définition d'une mesure pertinente d'évaluation d'un partitionnement dans les arbres de décision. Certains essaient de les classer selon un ou plusieurs critères (Shih, 1999) ; d'autres essaient de trouver une formulation générique permettant de retrouver l'ensemble des mesures sous forme de cas particuliers (Wehenkel, 1996). Un très grand nombre de travaux ont comparé leurs performances en utilisant un algorithme standard tel que ID3 dans lequel la mesure à tester est substituée à l'indicateur originel (le gain d'entropie de Shannon dans ce cas). La quasi-totalité de ces expérimentations ont montré que, dès lors que les mesures utilisées possèdent de bonnes propriétés de spécialisation, c'est-à-dire tendent à mettre en avant les partitions avec des feuilles pures, elles ne jouent pas un rôle majeur dans la qualité de la prédiction (Mingers, 1989 ; Buntine et Niblett, 1992), conclusion à laquelle étaient déjà arrivés les promoteurs de la méthode CART plusieurs années auparavant (Breiman *et al.*, 1984).

Enfin, un point important : on voit se dessiner ici un des principaux reproches que l'on peut adresser aux arbres de décision : leur instabilité. En effet, lorsque des descripteurs possèdent un pouvoir prédictif équivalent, la détection de la variable correspondant au maximum est fortement dépendant de l'échantillon d'apprentissage, les choix effectués sur les parties hautes de l'arbre

n'étant pas sans conséquence sur les choix réalisés sur les parties basses. Il est tout à fait possible d'obtenir un arbre visuellement très différent en modifiant quelques observations de l'échantillon. Cette instabilité est très gênante pour les praticiens, qui la comparent à des méthodes linéaires, comme l'analyse discriminante, où des modifications mineures dans l'échantillon se traduisent par une variation faible des coefficients calculés. Il faut cependant rappeler que, si le modèle de prédiction – l'arbre de décision – semble très différent, la variabilité de la prédiction sur un individu pris au hasard dans la population n'est pas aussi forte et, généralement, on lui attribuera la même étiquette.

3.2 Traitement des variables continues

Plaçons nous maintenant sur le sommet le plus à gauche sur le 2ème niveau de l'arbre. Il couvre 5 individus et a été segmenté à l'aide de la variable « humidité », le seuil de coupure utilisé étant « 77.5 % ». Ce résultat est la conséquence de deux tâches élémentaires :

1. Sélectionner la meilleure valeur de coupure pour chaque variable continue ;
2. Sélectionner globalement la meilleure segmentation en comparant la pertinence de tous les descripteurs : les descripteurs discrets et les descripteurs continus qui ont été découpés en 2 intervalles.

Choix du point de coupure

La première opération consiste à déterminer le meilleur point de coupure pour les variables continues. Dans ce didacticiel, nous considérons le cas du découpage binaire. Ceci n'est pas limitatif dans la mesure où il est possible de reconsidérer la même variable sur un sommet situé plus bas dans l'arbre et initier une autre discrétisation avec une valeur seuil différente. Les études évaluant l'opportunité d'une discrétisation n-aire ont par ailleurs montré qu'il n'y avait pas d'avantage à réaliser ce type de découpage, mis à part que l'on réduit visuellement le nombre de niveaux de l'arbre, sans en réduire le nombre de feuilles.

Le choix du seuil de discrétisation doit être cohérent avec la procédure de sélection des variables de segmentation ; il paraît donc naturel de faire intervenir dans le choix de la borne le t de Tschuprow qui sert à évaluer les partitions. Le procédé est alors relativement simple pour un descripteur continu X : il s'agit dans un premier temps de trier les données selon les valeurs de X, puis tester chaque borne de coupure possible entre deux valeurs de la variable en calculant le Tschuprow du tableau de contingence que l'on forme temporairement. Pour illustrer notre propos, considérons le cas de la variable « humidité » pour le sommet que nous voulons segmenter (Figure 2).

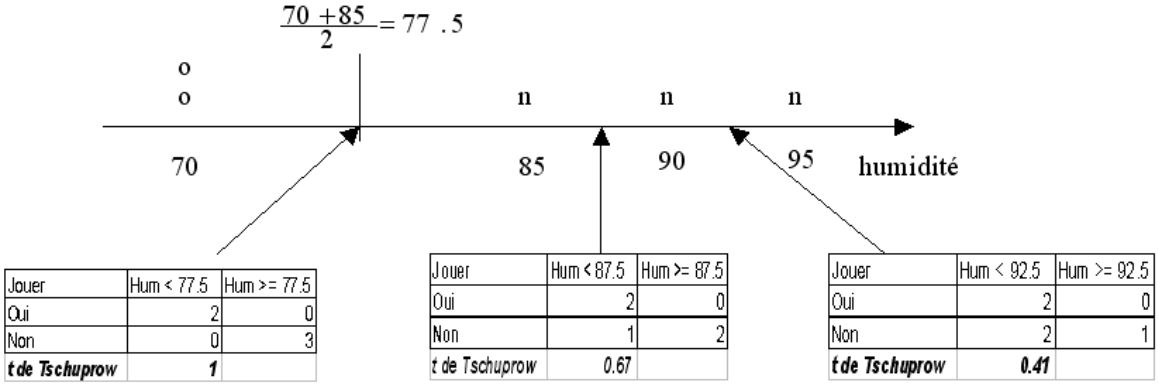

Figure 2 : Sélection de la borne de discrétisation

Détaillons les calculs et commentons-les.

- Il y a 5 observations sur le sommet, avec 4 valeurs distinctes de la variable « humidité ». Nous pouvons tester 3 points de coupures candidats.
- Généralement, le point de coupure est pris à mi-chemin entre 2 points successifs ; en réalité toute valeur située dans l'intervalle pourrait être utilisée.
- Il est inutile d'essayer de trouver un point de coupure entre deux valeurs ex-aequo. Cette remarque, qui semble tout à fait superflue (elle est visuellement évidente) n'est pas sans conséquences parfois désastreuses si l'on n'en tient pas compte lors de l'implémentation sur ordinateur.
- Pour chaque point de coupure à tester, nous formons le tableau de contingence et nous calculons l'indicateur associé ; le t de Tschuprow ici. Nous constatons que le meilleur découpage produit une partition pure, avec un Tschuprow égal à 1. La borne de découpage optimale est 77.5 %.

La discrétisation s'opère donc en deux étapes : (1) trier les données, (2) tester chaque point de coupure candidat et retenir celui qui optimise l'indicateur de qualité du partitionnement. Le temps de calcul n'est pas rédhibitoire tant que la base de données est de taille raisonnable, surtout sur les ordinateurs actuels. En revanche, dès lors que la base atteint une taille critique, de l'ordre de plusieurs centaines de milliers d'individus, avec un grand nombre de descripteurs continus, la majeure partie du temps de construction de l'arbre est utilisée à trier les données et à tester les points de coupures.

Il existe plusieurs stratégies pour remédier à ce goulot d'étranglement. Au lieu de trier localement les données sur le sommet que l'on traite, on les ordonne une fois pour toutes avant l'exécution de l'apprentissage et l'on conserve un index des variables triées (Witten et Franck, 2000). La borne de discrétisation étant de toute manière un estimateur, il est possible de le calculer sur un échantillon réduit des observations présentes sur le sommet (de l'ordre de 500 individuspar exemple) sans dégrader la qualité de l'apprentissage (Chauchat et Rakotomalala, 2000). Enfin, il ne paraît pas nécessaire de tester les points de coupures situés entre deux observations portant la même étiquette. Dans l'exemple, les deux bornes (87.5 et 92.5) ne devraient pas être évaluées. Des travaux ont montré qu'avec certaines mesures, il était impossible d'améliorer l'indicateur de qualité de partition avec un point de coupure situé entre deux individus de même étiquette (Fayyad et Irani, 1993 ; Muhlenbach et Rakotomalala, 2005).

Sélectionner la variable de segmentation

Après avoir déterminé le point de coupure optimal pour chaque variable continue, l'étape suivante consiste à déterminer la variable de segmentation pour le sommet traité.

La procédure est encore relativement simple. Il s'agit de sélectionner parmi l'ensemble des variables, discrètes ou continues discrétisées, celle qui maximise la mesure de référence sur le sommet que nous sommes en train de traiter. Dans notre cas, nous calculons donc le t de Tschuprow pour l'ensemble des variables (Tableau 5). Il apparaît que la variable « humidité » est optimale, ce qui n'est pas étonnant dans la mesure où elle a permis de mettre en avant des feuilles pures.

Descripteur	Point de coupure	T de Tschuprow
Humidité	77.5	1.00
Température	77.5	0.67
Vent	-	0.17
Soleil	-	0.00

Tableau 5: Segmentation candidates et bornes de discrétisation associées pour les descripteurs continus

La borne de discrétisation calculée localement lors de la segmentation peut être très instable car le résultat est fortement dépendant de l'échantillon d'apprentissage. De plus la valeur obtenue

peut ne pas être interprétable pour l'expert du domaine. Plusieurs solutions ont été mises en avant pour y remédier. La première possibilité est la faculté d'intervenir dans le processus d'élaboration de l'arbre. Dans le cas de la discrétisation, à la vue d'un résultat proposé par un algorithme, l'expert peut lui substituer une valeur de coupure plus appropriée pour un sommet ; le reste de l'arbre peut alors être construit automatiquement. La seconde possibilité est la définition d'un point de coupure « flou » : nous définissons sur un sommet non plus une estimation ponctuelle mais une distribution de points de coupures. Ceci permet de réduire considérablement la variabilité des arbres de décision mais peut nuire à leur lecture. En effet, un individu présenté sur le sommet sera redirigé sur plusieurs feuilles avec des poids différents ; ce processus de décision est moins immédiat (Suarez et Lutsko, 1999). Enfin, des chercheurs ont comparé les performances de la discrétisation locale, lors de la construction de l'arbre, avec une discrétisation globale des variables, dans une phase de pré-traitement, suivie d'une construction de l'arbre sur les données pré-discrétisées. Assez curieusement, il n'y a pas de différence notable de performance entre ces deux approches alors que le biais d'apprentissage est manifestement différent (Dougherty et al., 1995).

Dans l'exemple, si on ré-effectue les calculs, on constate que la première variable de segmentation à la racine n'est pas « ensoleillement » mais la variable « humidité » avec un seuil de 82.5. Nous avons volontairement exclu les variables continues lors de la segmentation de ce premier sommet pour les besoins de l'explication.

3.3 Définir la bonne taille de l'arbre

Dans leur monographie, Breiman *et al*. (1984) affirmaient que les performances d'un arbre de décision reposaient principalement sur la détermination de sa taille. Les arbres ont tendance à produire un « classifieur » trop complexe, collant exagérément aux données ; c'est le phénomène de sur-apprentissage. Les feuilles, mêmes si elles sont pures, sont composées de trop peu d'individus pour être fiables lors de la prédiction. Il a été démontré également que la taille des arbres a tendance à croître avec le nombre d'observations dans la base d'apprentissage (Oates et Jensen, 1997). Le graphique mettant en relation les taux d'erreur (calculés sur l'échantillon servant à l'élaboration du modèle et sur un échantillon à part) avec le nombre de feuilles de l'arbre a servi à montrer justement la nécessité de déterminer une règle suffisamment efficace pour assurer les meilleures performances à l'arbre de décision (Figure 3). Dans cet exemple, nous voyons effectivement qu'à mesure que le nombre de feuilles – la taille de l'arbre – augmente, le taux d'erreur calculé sur les données d'apprentissage diminue constamment. En revanche, le taux d'erreur calculé sur l'échantillon test montre d'abord une décroissance rapide, jusqu'à un arbre avec une quinzaine de feuilles, puis nous observons que le taux d'erreur reste sur un plateau avant de se dégrader lorsque l'arbre est manifestement surdimensionné.

L'enjeu de la recherche de la taille optimale consiste à stopper - *pré-élagage* - ou à réduire - *post-élagage* - l'arbre de manière à obtenir un classifieur correspondant au « coude » de la courbe sur échantillon test, lorsque le taux d'erreur commence à stagner. Dans ce qui suit, nous détaillerons tout d'abord la méthode implémentée par CHAID (pré-élagage) ; vue l'importance du sujet, nous étudierons le post-élagage dans la section suivante.

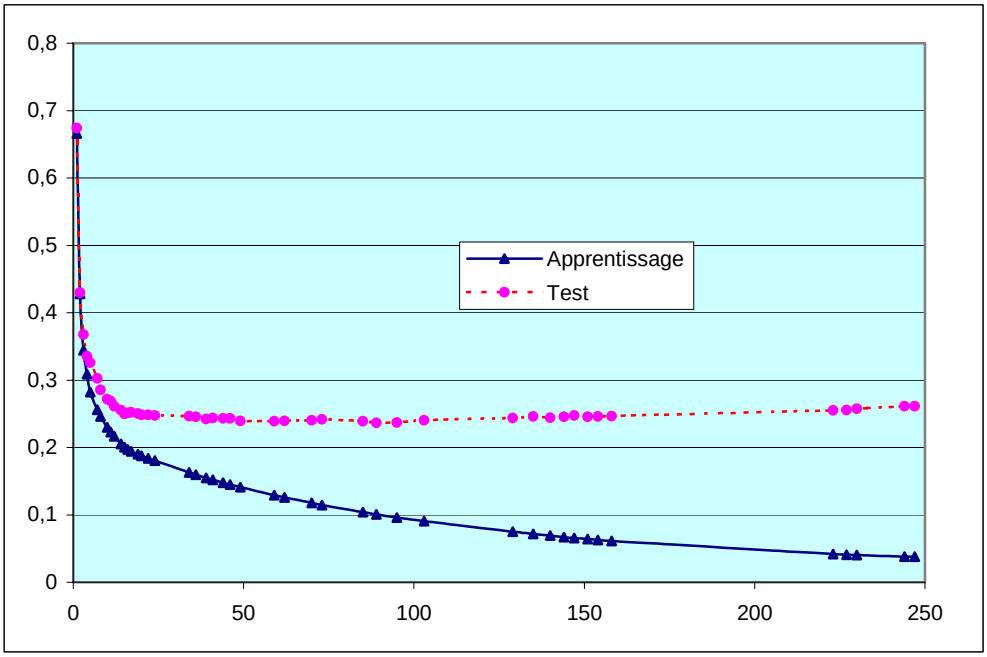

Figure 3 : Evolution taux d'erreur en apprentissage et en test

Pré-élagage

Le pré-élagage consiste à fixer une règle d'arrêt qui permet de stopper la construction de l'arbre lors de la phase de construction. Une approche très simple consiste à fixer un critère d'arrêt local, relatif au sommet que l'on est en train de traiter, qui permet d'évaluer *l'apport informationnel* de la segmentation que l'on va initier.

En ce sens, CHAID a le mérite de la cohérence : on accepte la segmentation si le Khi-2 calculé (ou le t de Tschuprow) sur un sommet est significativement supérieur à un seuil que l'on se fixe. La formalisation passe par un test d'hypothèse statistique : l'hypothèse nulle est l'indépendance de la variable de segmentation avec l'attribut classe. Si le Khi-2 calculé est supérieur au seuil théorique correspondant au risque de première espèce que l'on s'est fixé, on accepte la segmentation (ou ce qui revient au même, si la *p-value* calculée est inférieure au risque de première espèce).

Dans l'exemple ci-dessus, pour segmenter le sommet le plus à gauche du second niveau, nous avons utilisé la variable « humidité » qui donne un t de Tschuprow égal à 1.0. En réalisant le test d'indépendance du Khi-2, la p-value calculée est de 0.025 ; si nous fixons un risque de première espèce de 5% la segmentation sera acceptée ; si nous fixons un risque de 1%, elle sera refusée.

Nous sommes en présence, dans cet exemple, de la principale difficulté de cette approche : comment choisir le risque qui sera utilisé dans le test ? En effet, l'arbre résultant sera fortement dépendant du paramètre que l'on aura choisi. Il est très difficile de choisir correctement le seuil dans la pratique : s'il est trop restrictif, l'arbre sera sous-dimensionné (dans l'exemple, avec un seuil à 1%, l'arbre aurait été stoppé dès la racine); s'il est trop permissif, l'arbre sera sur-dimensionné. Ce problème est théoriquement insoluble parce que la règle d'arrêt n'a aucun lien direct avec l'objectif de construire un arbre de décision le plus précis possible dans la phase de prévision. Le test correspond à un test d'indépendance statistique, utilisant le Khi-2 qui est une mesure symétrique, donc nous ne nous situons pas dans une situation de prévision. De plus, lorsque les effectifs sont élevés, nous sommes souvent obligés de fixer un risque de première espèce très bas, à la limite de la précision des calculateurs, pour espérer contrôler la taille de l'arbre. Enfin l'évaluation est locale à un sommet : on ne tient pas compte du comportement global de l'arbre. Malgré tout, à l'usage, cette approche donne cependant de bons résultats. On en devine l'explication en regardant le graphique d'évolution de l'erreur ci-dessus (Figure 3) : la plage dans laquelle l'erreur en généralisation est

faible est relativement large ; il suffit donc de proposer une règle assez frustre pour obtenir un arbre convenable (en prenant garde à ne pas produire un arbre sous-dimensionné).

Plus ennuyeux aux yeux des puristes est l'utilisation même du test ci-dessus. En effet, nous ne sommes pas en présence d'un test d'indépendance classique car la variable que nous testons a été produite aux termes de plusieurs étapes d'optimisation : recherche du point de discrétisation optimal pour les variables continues ; recherche ensuite de la variable de segmentation qui maximise la mesure utilisée. Nous nous trouvons en situation de comparaisons multiples et la loi statistique n'est plus la même : nous accepterons *trop* souvent les segmentations (Jensen et Cohen, 2000). On peut songer corriger le test en introduisant certaines procédures connues comme la correction de Bonferroni (présentée d'ailleurs dans le descriptif originel de CHAID). En réalité le risque critique joue avant tout le rôle d'un paramètre de contrôle de la taille de l'arbre. Dans la pratique, ce type de correction n'amène pas d'amélioration en termes de performances de classement.

D'autres critères plus empiriques relatifs à la taille des feuilles peuvent être mis en place. L'objectif est d'éviter l'apparition de sommets d'effectifs trop faibles pour espérer obtenir une prédiction fiable. Ils reposent en grande partie sur l'intuition du praticien. Ils peuvent également être fixés en procédant à des essais : la première stratégie consiste à fixer une taille de sommet à partir de la laquelle nous ne réalisons plus de tentative de segmentation ; la seconde revient à fixer un effectif d'admissibilité : si une des feuilles produites par la segmentation est inférieure à un seuil que l'on s'est fixé, l'opération est refusée. De nature plutôt empirique, ces règles d'arrêt se révèlent pratiques lors de la mise en oeuvre des arbres de décision dans des études réelles.

Post-élagage

Cette approche est apparue avec la méthode CART (Breiman *et al*, 1984). Elle a été très largement reprise sous différentes formes par la suite. Le principe est de construire l'arbre en deux temps : une première phase d'expansion, où l'on essaie de produire des arbres les plus purs possibles et dans laquelle nous acceptons toutes les segmentations même si elles ne sont pas pertinentes – c'est le principe de construction « hurdling » ; dans un second temps, nous essayons de réduire l'arbre en utilisant un autre critère pour comparer des arbres de tailles différentes. Le temps de construction de l'arbre est bien sûr plus élevé ; il peut être pénalisant lorsque la base de données est de très grande taille ; en contrepartie, l'objectif est d'obtenir un arbre plus performant en classement.

Deux approches s'opposent dans la littérature. La première, en s'appuyant sur des formulations bayésiennes (ou des dérivées telles que la théorie de la description minimale des messages) transforment le problème d'apprentissage en un problème d'optimisation. Le critère traduit le compromis entre la complexité de l'arbre et son aptitude à coller aux données. Dans la théorie de la longueur minimale des messages, le critère établit un comprmis entre la quantité d'informations nécessaire pour décrire l'arbre, et les données qui font exception à l'arbre (Wallace et Patrick, 1993). Malgré l'élégance des formulations utilisées, il faut reconnaître que ces méthodes sont peu connues ; elles ne sont d'ailleurs implémentées que dans quelques programmes distribués sous forme de code source (Buntine, 1991 ; Kohavi et Sommerfield, 2002).

Plus répandues sont les méthodes s'appuyant sur une estimation non-biaisée du taux d'erreur en classement lors de la phase d'élagage. Certaines utilisent une estimation calculée sur le même échantillon d'apprentissage mais pénalisée par la taille de l'effectif du sommet à traiter (C4.5 - Quinlan, 1993) ; d'autres utilisent une évaluation du taux d'erreur avec un second échantillon, dit de validation (le terme anglais « pruning set » est moins ambigu) (cas de CART - Breiman *et al*., 1994). Le parallèle entre ces deux méthodes a été réalisé dans un article publié par deux auteurs importants dans le domaine des arbres (Kohavi et Quinlan, 2002). La première est plus connue dans le monde de l'apprentissage automatique ; la seconde est plus cotée chez les statisticiens. Nous nous bornerons à dire que la méthode CART se révèle plus robuste dans la pratique. Elle intègre tous les « bons » ingrédients d'un apprentissage efficace : évaluation non biaisée de l'erreur pour déterminer

le bon arbre ; réduction de l'espace des hypothèses avec le principe des séquences d'arbres rangés à coût-complexité décroissant, limitant ainsi le risque de sur-apprentissage sur l'échantillon de validation ; préférence donnée à la simplicité avec la règle de « 1'écart-type » avant l'erreur minimale : l'idée est de se rapprocher du coude dans l'évolution de l'erreur en fonction du nombre de feuilles de l'arbre (Figure 3). Lorsque la taille du fichier d'apprentissage est réduite, un système de validation croisée est proposé pour réaliser le post-élagage.

3.4 Décision

Dernière étape de la construction de l'arbre : affecter une conclusion à chaque feuille de l'arbre. Le chemin reliant une feuille à la racine de l'arbre peut être lu comme une règle de prédiction du type attribut-valeur « Si prémisse… alors Conclusion… » ; comment conclure, c'est-à-dire attribuer une étiquette à une feuille ?

Lorsque la feuille est pure, lui attribuer la conclusion correspond à la seule modalité présente semble naturel. Dans l'exemple, toutes les feuilles étant pures, nous pouvons très facilement déduire 5 règles (Tableau 6).

N°	Prémisse	Conclusion
1	Ensoleillement = « Soleil » ET Humidité < 77.5	Jouer = « oui »
2	Ensoleillement = « Soleil » ET Humidité >= 77.5	Jouer = « non »
3	Ensoleillement = « Couvert »	Jouer = « oui »
4	Ensoleillement = « pluie » ET Vent = « oui »	Jouer = « non »
5	Ensoleillement = « pluie » ET Vent = « non »	Jouer = « oui »

Tableau 6: Règles extraites de l'arbre de la Figure 1

En revanche, lorsque plusieurs modalités sont présentes dans la feuille, il faut utiliser une règle d'attribution efficace. La règle la plus souvent utilisée est la règle de la majorité : on affecte à la feuille la modalité de la variable à prédire qui présente l'effectif le plus grand. Cette règle, qui semble de bon sens, repose sur des fondements théoriques bien établis. En effet, la distribution de fréquences visible sur la feuille est une estimation de la probabilité conditionnelle d'appartenance à chaque étiquette de la variable à prédire ; affecter à la feuille l'étiquette la mieux représentée minimise donc la probabilité de mauvaise affectation sous deux conditions : les données constituent un échantillon représentatif de la population ; les coûts de mauvaise affectation sont unitaires (les bonnes affectations coûtent 0, et les mauvaises affectations coûtent 1).

Lorsque nous nous écartons de ce cadre, notamment lorsque les coûts de mauvaise affectation ne sont pas symétriques, ce qui est souvent le cas dans les études réelles, il faut se méfier de la règle de la majorité : la conclusion devrait être celle qui minimise le coût moyen de mauvaise affectation. Lle détail des calculs sur un exemple est décrit dans l'ouvrage de Bardos (2001).

3.5 Fusion des sommets lors de la segmentation

CHAID intègre une variante assez intéressante par rapport aux quatre éléments standards évoqués dans ce didacticiel : la possibilité de fusionner les sommets enfants issus d'une segmentation (Kass, 1980). Initialement, chaque modalité du descripteur induit un sommet enfant (ID3 et C4.5 par exemple) ; certaines méthodes comme CART imposent l'induction d'un arbre binaire et donc les modalités sont regroupées en deux sous-ensembles de manière à optimiser l'indicateur de qualité de la partition. Dans CART (Breiman *et al.*, 1984), le regroupement n'est pas justifié, l'objectif étant plutôt de proposer des solutions pour réduire le nombre de calculs nécessaires pour produire le regroupement binaire optimal. Quinlan (1993) a exploré de manière

empirique l'influence des regroupements, il montre que cet artifice permet de réduire la « largeur » de l'arbre sans vraiment en améliorer les performances en classement.

Pourtant, les avantages du regroupement ne sont pas négligeables (Rakotomalala, 1997). Il permet de lutter contre la fragmentation, surtout préjudiciable lorsque l'on travaille sur des petits effectifs ; il améliore la lisibilité de l'arbre en isolant les modalités non-informatives des descripteurs ; il permet aussi de réduire la taille de l'arbre en évitant que des séquences de segmentations se répètent dans différentes zones de l'arbre (la « réplication des sous-arbres »).

CHAID propose un procédé original, toujours en adéquation avec son approche statistique. Il vérifie la proximité des profils des sommets enfants issus de la segmentation et fusionne itérativement les sommets produisant des feuilles avec des distributions similaires. Il utilise pour ce faire un test d'équivalence distributionnelle du Khi-2 pour lequel un paramètre - le risque de première espèce du test - est fixé par l'utilisateur. L'algorithme est très simple : on fusionne d'abord les deux feuilles présentant le profil le plus proche, au sens du test ; on réitère l'opération sur les feuilles restantes jusqu'à ce qu'aucune fusion ne soit plus possible. Il se peut qu'il n'y ait aucune fusion de réalisée pour une segmentation donnée ; il se peut aussi que tous les sommets enfants soient fusionnés dans un seul groupe, rejetant d'office la possibilité de segmenter avec le descripteur.

Pour deux sommets à fusionner, la statistique du Khi-2 est la suivante, elle suit une loi du χ^2 à (K-1) degrés de liberté sous l'hypothèse d'égalité des distributions :

$$\chi^2 = \sum_{k=1}^{K} \frac{\left(\frac{n_{k1}}{n_{.1}} - \frac{n_{k2}}{n_{.2}} \right)^2}{\frac{n_{k1} + n_{k2}}{n_{.1} \times n_{.2}}}$$

Reprenons l'exemple et introduisons maintenant la possibilité de réaliser des fusions (Figure 1). La segmentation de la racine de l'arbre à l'aide de la variable « ensoleillement » est maintenant précédée d'une phase de fusion des sommets enfants générés. Pour faciliter la lecture, nous numérotons les sommets enfants du second niveau de droite à gauche (a, b et c). Fixons le risque de première espèce du test d'équivalence distributionnelle à 10%. Le premier passage essaie de fusionner les sommets deux à deux ; les résultats des calculs sont repris dans le Tableau 7.

Sommets	Distribution	CHI-2	p-value	Sortie
a & b	(2 ; 3) et (4 ; 0)	3.60	0.058	
a & c	(2 ; 3) et (3 ; 2)	0.40	0.527	***fusion***
b & c	(4 ; 0) et (3 ; 2)	2.06	0.151	

Tableau 7 : 1ère passe, fusion des sommets pour la segmentation de la racine

Nous constatons qu'au risque de 10%, les sommets (a & c) et (b & c) peuvent être fusionnés (p-value supérieure au seuil). Les distributions qui sont les plus proches sont celles des sommets (a & c), nous décidons donc de les fusionner. Au deuxième passage, nous re-numérotons les sommets en A (a & c) et B (b). Puis relançons les calculs (Tableau 8). Nous constatons dans ce cas qu'aucune fusion n'est possible, les deux sommets présentant des distributions différentes au sens de notre test.

Sommets	Distribution	CHI-2	p-value	Sortie
A & B	(5 ; 5) et (4 ; 0)	3.11	0.078	

Tableau 8 : 2ème passe, fusion des sommets pour la segmentation de la racine

En activant cette option, la segmentation de la racine de l'arbre à l'aide de la variable « ensoleillement » aurait donc produit le partitionnement suivant (Figure 4).

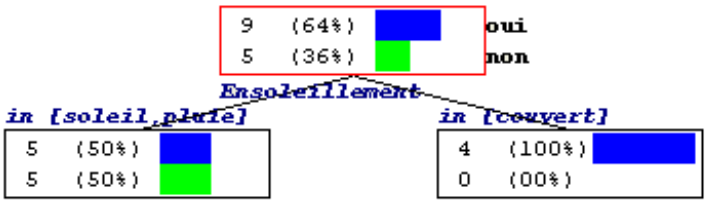

Figure 4 : Segmentation de la racine de l'arbre avec fusion des sommets enfants

Notons que cette technique ne joue aucun rôle lorsque nous traitons les descripteurs continus car la segmentation est forcément binaire dans ce cas. Notons également que si la variable de segmentation est ordinale, il est possible d'intégrer cette contrainte dans la recherche des fusions (dans notre exemple, on peut tester les fusions a&b et b&c, mais pas a&c).

4 Un exemple détaillé

4.1 Données et logiciels

Pour illustrer l'induction des arbres de décision sur des données réelles – ou tout du moins réalistes – nous avons choisi le fichier IRIS de Fisher (1936) : il décrit 150 observations correspondant à trois variétés d'iris {« setosa », « versicolor », « virginica »} à partir de leurs caractéristiques morphologiques (longueur des sépales, largeur des sépales, longueur des pétales, largeur des pétales). Il est accessible sur plusieurs serveurs de données, sur le site UCI Irvine par exemple (Hettich et Bay, 1999). L'intérêt de ce fichier est essentiellement pédagogique, il présente des particularités très intéressantes qui rendent la compréhension et l'interprétation des résultats faciles. Pour d'autres types d'études, données financières ou médicales, il existe des exemples détaillés dans plusieurs publications en français (Gueguen, 1994 ; Zighed et Rakotomalala, 2000 ; Bardos, 2001).

Pour ce didacticiel, nous avons utilisé le logiciel SIPINA (Système Interactif pour l'Induction Non-Arborescente), initié par Zighed (1992) ; la version que nous utilisons a été développée par nos soins de 1998 à 2000 (http://eric.univ-lyon2.fr/~ricco/sipina). Ce logiciel est dédié à l'apprentissage supervisé ; il possède une large bibliothèque d'algorithmes d'induction d'arbres. Par rapport aux logiciels libres disponibles par ailleurs, il intègre également un module interactif qui permet à l'utilisateur d'intervenir manuellement lors de la construction de l'arbre. Cette spécificité, très rarement disponible sur les logiciels de recherche, est en revanche systématiquement présente dans les logiciels commerciaux telles que COGNOS, SAS, SPAD, SPSS, STATISTICA ; elle a très largement contribué à populariser les arbres de décision auprès des praticiens.

SIPINA présente des lacunes qui nous ont poussé à développer le logiciel TANAGRA (Rakotomalala, 2005) : les choix architecturaux nous ont empêché d'intégrer des méthodes autres que le supervisé ; il n'est pas possible de sauvegarder les traitements et leur enchaînement, imposant à l'utilisateur de refaire toute la séquence de manipulations s'il veut reprendre une analyse interrompue. Concernant les arbres, la possibilité d'interagir en explorant les sommets de l'arbre oblige à sauvegarder une quantité importante d'informations qui limite les performances du logiciel sur de très grandes bases de données. Relativisons néanmoins cette dernière limitation qui était rédhibitoire il y a quelques années : notre machine de développement disposait de 64 Mo de RAM. Elle est moins contraignante de nos jours : notre machine actuelle dispose de 1 Go de RAM. Il est

vrai que, dans le même temps, la taille moyenne des bases de données a également augmenté, mais dans des proportions moindres.

4.2 Analyse automatique avec CHAID

Le logiciel est téléchargeable sur le site indiqué ; la procédure d'installation est standardisée. La première étape consiste à charger les données via le menu *FILE / OPEN* , choisir l'extension *TXT* pour accéder au format texte. Dans SIPINA, le point décimal est toujours le « . » quelle que soit la version de Windows. Il faut également spécifier le type de séparateur (tabulation) et indiquer que la première ligne contient le nom des variables. Les données sont alors affichées dans la grille principale du logiciel. Il est possible d'éditer les données bien que les possibilités en ce sens soient assez réduites.

Par défaut, « IMPROVED CHAID » est la méthode d'apprentissage sélectionnée. Elle correspond au descriptif de ce didacticiel. Elle diffère de CHAID essentiellement par l'utilisation du t de Tschuprow comme mesure d'évaluation du partitionnement. Nous devons maintenant définir le problème à traiter en sélectionnant les descripteurs (les 4 caractéristiques) et l'attribut à prédire (la variable TYPE) avec le menu *ANALYSIS / DEFINE CLASS ATTRIBUTE* ; puis, subdiviser le fichier en partie apprentissage 67% (100 individus) et test 33% (50 individus) à l'aide du menu *ANALYSIS / SELECTIVE ACTIVE EXAMPLES*, puis en choisissant l'option *RANDOM SAMPLING*. La subdivision étant aléatoire, il est possible que vous n'obteniez pas un partage identique à ce que nous décrivons dans ce document. La fenêtre principale du logiciel doit correspondre à la figure ci-dessous. On note à gauche la description du traitement que nous allons exécuter avec plus particulièrement les paramètres par défaut de l'induction : risque de 1ère espèce pour la fusion – 5% - et la règle d'arrêt – 1% ; taille minimale du sommet à segmenter – 10 ; taille minimal des sommets enfants générés – 5 (Figure 5).

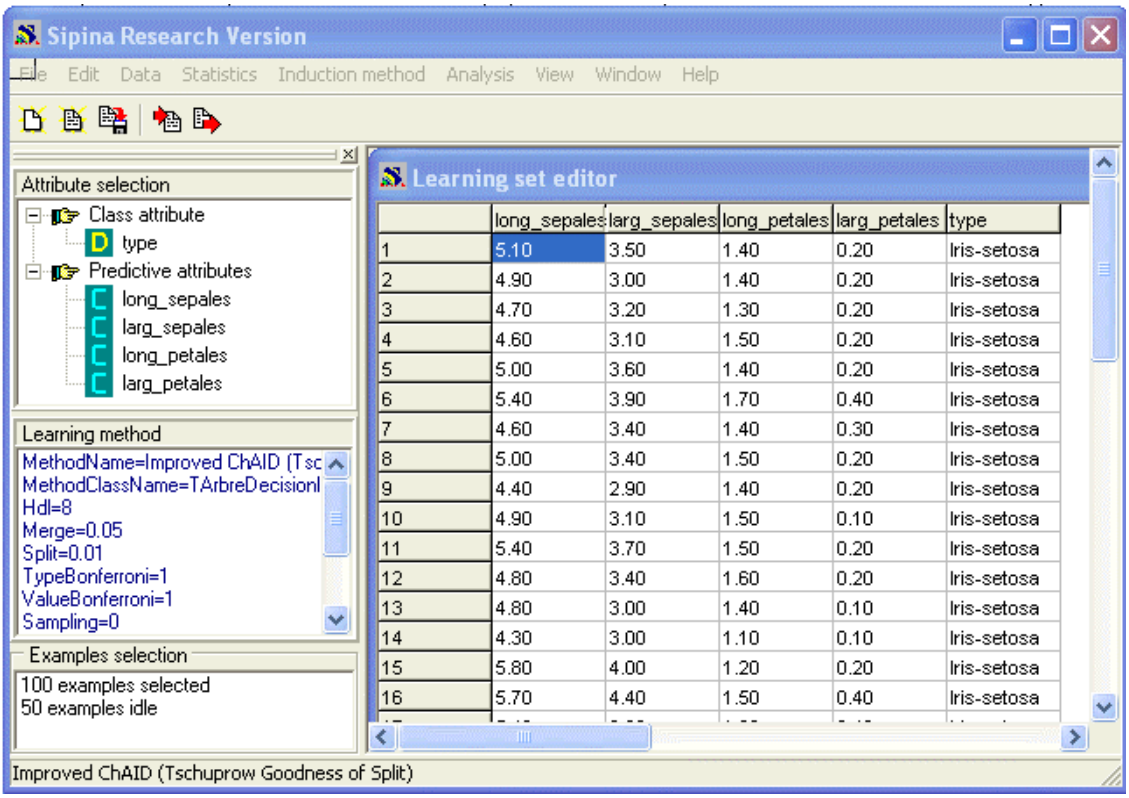

Figure 5 : Fenêtre principale du logiciel SIPINA

A ce stade, il est possible de lancer l'analyse en activant le menu ANALYSIS / LEARNING, l'arbre de décision obtenu est décrit dans la figure suivante (Figure 6).

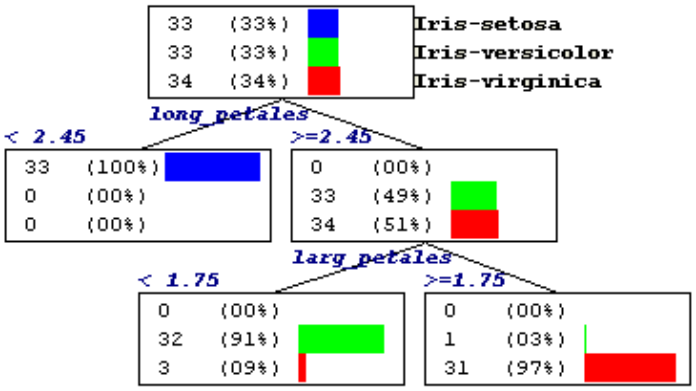

Figure 6 : Arbre de décision sur 100 individus pris au hasard dans le fichier IRIS

La lecture des règles de décision est immédiate : trois règles ont été produites ; nous constatons également que 2 variables seulement parmi les 4 ont été réellement utilisées (Tableau 9).

N°	Prémisse	Conclusion
1	Longueur des pétales < 2.45	Type = « setosa »
2	Longueur des pétales >= 2.45 ET Largeur des pétales < 1.75	Type = « versicolor »
3	Longueur des pétales >= 2.45 ET Largeur des pétales >= 1.75	Type = « virginica »

Tableau 9: Règles extraites de l'arbre traitant le fichier IRIS (Figure 6)

4.3 Evaluation du modèle de prédiction

Une manière classique d'évaluer la qualité de l'apprentissage est de confronter la prédiction du modèle avec les valeurs observées sur un échantillon de la population. Cette confrontation est résumée dans un tableau croisé appelé matrice de confusion. Il est possible d'en extraire des indicateurs synthétiques, le plus connu étant le taux d'erreur ou taux de mauvais classement. Il est possible de l'interpréter comme un coût moyen de mauvais classement lorsque la matrice de coût de mauvaise affectation est unitaire ; il est également possible de l'interpréter comme un estimateur de la probabilité d'effectuer une mauvaise prédiction à l'aide de l'arbre de décision.

Le principal intérêt du taux d'erreur est qu'il est objectif ; il sert généralement à comparer les méthodes d'apprentissage sur un problème donné. Pour obtenir un indicateur non biaisé, il est impératif de ne pas le mesurer sur l'échantillon qui a servi à élaborer le modèle. A cet effet, le praticien met souvent de côté un échantillon, dit de test, qui servira à évaluer et à comparer les modèles. Dans notre exemple, nous allons utiliser les 50 individus qui n'ont pas servi à l'apprentissage. Après les avoir étiquetés avec l'arbre de décision, nous obtenons la matrice de confusion (Tableau 10). Dans SIPINA, nous devons activer le menu ANALYSIS / TEST et choisir l'option INACTIVE EXAMPLES (choisir les exemples actifs reviendrait à évaluer le modèle sur les données ayant servi à sa construction, le taux d'erreur obtenu est dit de resubstitution dans ce cas).

		Prédite			
		Setosa	Versicolor	Virginica	Somme
Observé	Setosa	17	0	0	17
	Versicolor	0	17	0	17
	Virginica	0	2	14	16
	Somme	17	19	14	50

Tableau 10: Matrice de confusion sur l'échantillon test

Le taux d'erreur en test est égal à $\varepsilon_{test} = \frac{2}{50} = 4\%$, nous pouvons donc dire qu'en classant un individu pris au hasard dans la population, nous avons 4 chances sur 100 de réaliser une mauvaise affectation. Attention, si le taux d'erreur en test est non biaisé, il ne donne aucune idée sur la variance de l'indicateur ; il est plus approprié dans ce cas d'utiliser les méthodes de ré-échantillonnage telles que la validation croisée ou le bootstrap (Efron et Tibshirani, 1997).

Quoiqu'il en soit, ce taux d'erreur est intéressant. En effet, sans modèle, si nous attribuons au hasard une étiquette à un individu de la population, compte tenu de la répartition initiale, nous avons 67% de chances de réaliser une mauvaise affectation.

4.4 Interprétation géométrique

Un des principaux intérêts du fichier IRIS, outre sa paternité prestigieuse et sa large diffusion dans notre communauté, est la possibilité de donner une interprétation simple et visuelle à la règle d'affectation. L'arbre a sélectionné 2 variables pertinentes, nous allons donc projeter les points dans le plan en mettant en évidence l' étiquette des individus (Figure 7).

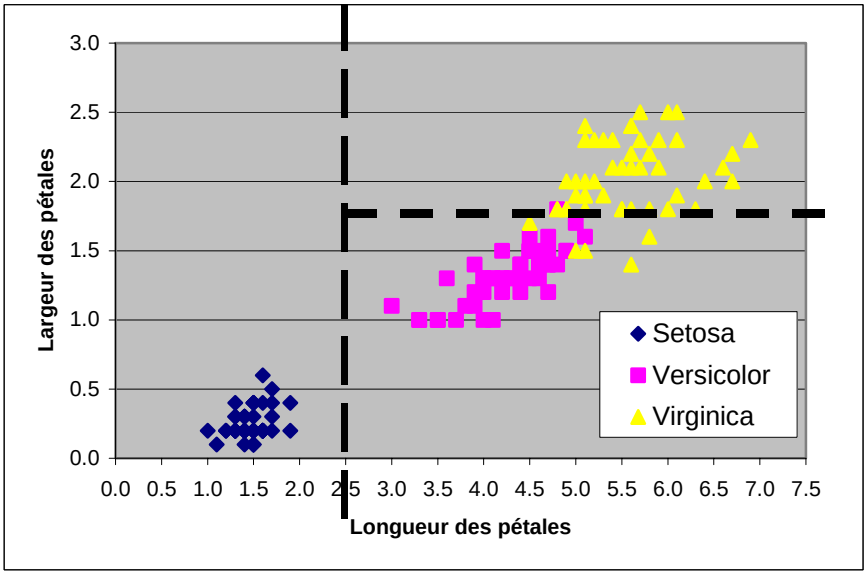

Figure 7 : Frontières induites par l'arbre de décision dans l'espace de représentation

Le principe d'induction par arbre de décision est très bien traduit par ce graphique : la méthode vise à produire des sous-groupes aussi homogènes que possibles en traçant des droites de séparation dans l'espace de représentation. Par rapport aux méthodes linéaires classiques telles que l'analyse discriminante, ces droites ont la particularité d'être « parallèles aux axes », elles peuvent également s'imbriquer. Au final, le modèle de prédiction global est non-linéaire.

De fait, il est possible avec un arbre de décision de trouver une représentation qui apporte une solution à tout problème de discrimination pour peu que la classe ne soit pas « bruitée », cas où

plusieurs individus décrits de la même manière ont des étiquettes différentes. Cela ne veut pas dire pour autant qu'un algorithme d'induction d'arbre est capable de trouver la solution. En effet le principe de construction pas à pas, local sur chaque sommet, qualifié de « myope » (Kononenko et al., 1997), empêche de trouver la solution globalement optimale ; le partitionnement successif entraîne la fragmentation des données et il devient rapidement difficile de trouver les bonnes frontières dans certaines zones de l'espace de représentation car nous ne disposons plus d'observations suffisantes.

4.5 Manipulation interactive de l'arbre

A la lumière du graphique ci-dessus (Figure 7), nous constatons qu'il existe plusieurs solutions au problème de discrimination des IRIS, en utilisant les mêmes deux variables de notre ensemble de données. Il aurait été possible par exemple de segmenter la racine avec la variable « Largeur des pétales » avec un seuil (estimé visuellement) égal à 0.8 ; cela aurait permis également d'isoler complètement les observations portant l'étiquette « setosa ». De la même manière, au second niveau, dans la partie droite de l'arbre, nous pouvons également initier une coupure avec la variable « Longueur des pétales », avec un seuil estimé visuellement à 5.0. Un des avantages décisifs des arbres de décision est justement la possibilité pour le praticien de tenter des segmentations différentes, inspirées par les connaissances du domaine (la variable est plus fiable, plus représentative d'un phénomène connu, moins coûteuse à mesurer, etc.). Cela permet ainsi de mieux décider lorsque deux descripteurs sont en compétition pour la segmentation d'un nœud. La popularité des logiciels qui implémentent les arbres de décision repose en grande partie sur cette fonctionnalité ; il est impensable à l'heure actuelle de diffuser un logiciel commercial qui ne l'intègre pas.

Dans l'exemple d'arbre de décision, nous allons évaluer les segmentations concurrentes sur le sommet à droite au second niveau de l'arbre. Pour ce faire, dans SIPINA, nous allons tout d'abord supprimer les feuilles qui lui sont consécutifs en activant le sommet, effectuer un clic avec le bouton droit de la souris et sélectionner l'option de menu CUT. Pour visualiser les segmentations candidates, il faut ré-activer de nouveau le menu contextuel et cliquer sur l'option SPLIT NODE. Une boîte de dialogue affichant tous les descripteurs triés par ordre décroissant de l'indicateur de qualité de la segmentation apparaît alors (Figure 8).

Nous constatons que si la variable « Largeur des pétales » tient la première place avec un t de Tschuprow de 0.56, la variable « Longueur des pétales » en est très proche avec un t de 0.48 ; le seuil de discrétisation est dans ce cas égal à 4.75 (dans la partie basse de la fenêtre), ce qui semble justifié au regard du graphique représentant les observations dans le plan. La segmentation associée est également acceptée, symbolisée par l'histogramme de couleur verte. En ce qui concerne les deux autres variables, nous constatons que « Longueur de sépales » se situe assez loin avec un t de 0.08 ; même si la solution proposée est acceptée, la segmentation proposée par la variable « Largeur des sépales » est en revanche refusée.

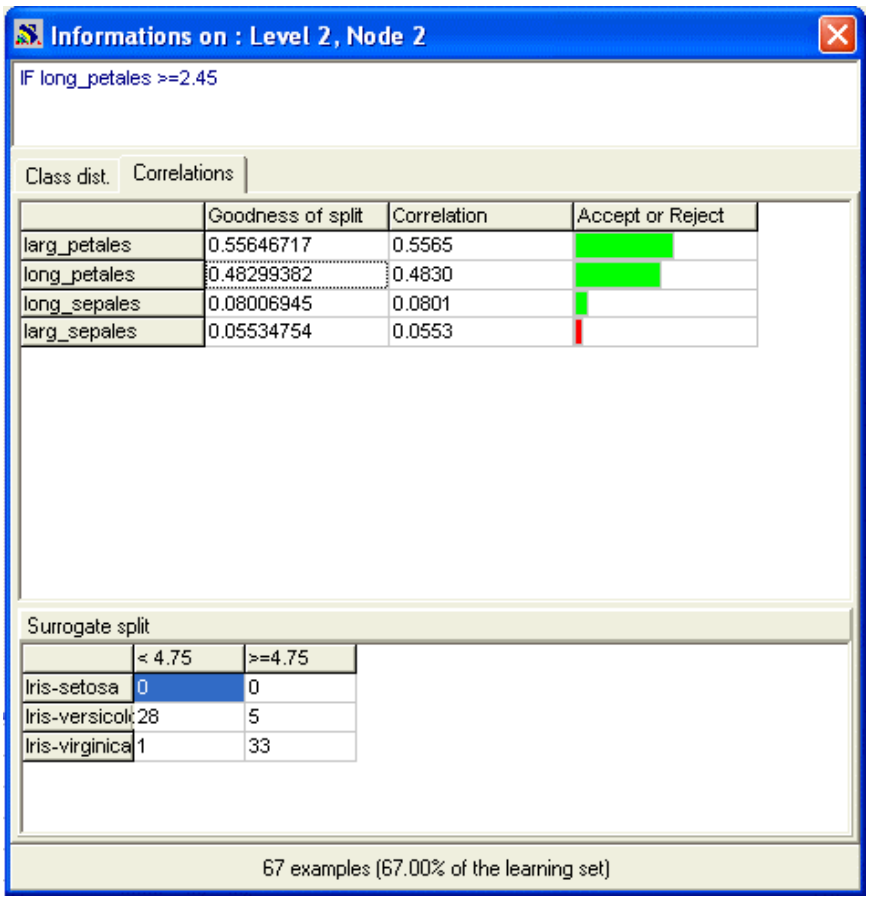

Figure 8 : Segmentations alternatives sur le second sommet du second niveau

Pour appliquer la nouvelle segmentation utilisant la variable « Longueur des pétales », il faut double-cliquer sur la case correspondante. L'arbre de décision prend alors l'aspect suivant (Figure 9).

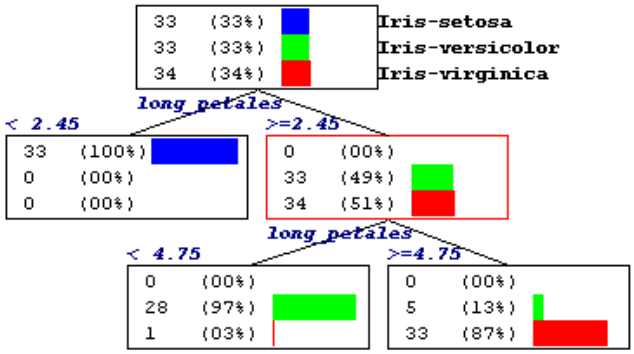

Figure 9 : Arbre de décision après modification manuelle de la variable de segmentation

Dans les logiciels commerciaux, il est même possible de modifier à la main les seuils de discrétisation, ce qui n'est pas le cas de SIPINA.

Il est bien entendu possible de ré-évaluer l'arbre ainsi construit sur l'échantillon test pour déterminer la meilleure solution. Mais il faut être prudent face à cette pratique ; en effet l'échantillon devient partie prenante dans la construction du modèle, il joue le rôle d'échantillon de

réglage (« tuning set » en anglais). En réalité il agit comme un second fichier d'apprentissage dans ce cas.

En offrant à l'expert du domaine la faculté de comprendre et d'intervenir au cœur de l'induction, les arbres de décision élargissent considérablement leur champ d'action. Il est ainsi possible d'intégrer directement dans le processus d'apprentissage les connaissances et les contraintes du domaine. Cette caractéristique est souvent mise en avant lors de leur mise en œuvre dans les problèmes réels, en médecine par exemple (Crémilleux, 1997).

5 Quelques éléments de discussion

5.1 Avantages et inconvénients

L'induction par arbres de décision est une technique arrivée à maturité ; ses caractéristiques, ses points forts et ses points faibles sont maintenant bien connus ; il est possible de la situer précisément sur l'échiquier des très nombreuses méthodes d'apprentissage (Hastie *et al*., 2001).

Les arbres présentent des performances comparables aux autres méthodes supervisées ; les nombreuses comparaisons empiriques l'ont suffisamment montré (Zighed et Rakotomalala, 2000 ; Lim *et al*., 2000). La méthode est non paramétrique ; elle ne postule aucune hypothèse a priori sur la distribution des données ; elle est résistante aux données atypiques ; le modèle de prédiction est non linéaire. Lorsque la base d'apprentissage est de taille importante, elle présente des propriétés similaires aux algorithmes des plus proches voisins (Breiman *et al*., 1984).

Il faut néanmoins tempérer ce constat. Le premier reproche qu'on peut lui adresser est son incapacité, avec les algorithmes classiques (C4.5, CART, CHAID, etc.), à détecter les combinaisons de variables ; ceci est dû au principe de construction pas à pas de l'arbre, entraînant une certaine « myopie ». Le second reproche est dans la nécessité de disposer d'un échantillon d'apprentissage de grande taille. L'arbre certes peut reproduire approximativement toutes formes de frontières, mais au prix d'une fragmentation rapide des données, avec le danger de produire des feuilles avec très peu d'individus. Corollaire à cela, les arbres sont en général instables ; les bornes de discrétisation notamment dans les parties basses de l'arbre sont entachées d'une forte variabilité. Ainsi, certains chercheurs préconisent de procéder à la discrétisation préalable des variables avant la construction de l'arbre (Dougherty *et al*., 1995).

L'induction par arbre de décision est capable de traiter de manière indifférenciée les données continues et discrètes. Elle dispose de plus d'un mécanisme naturel de sélection de variables. Elle doit être privilégiée lorsque l'on travaille dans des domaines où le nombre de descripteurs est élevé, dont certains, en grand nombre, sont non-pertinents. Nous devons également relativiser cette affirmation. En effet, non sans surprise, des travaux dans le domaine de la sélection de variables ont montré que la réduction préalable des descripteurs dans des domaines fortement bruités améliorait considérablement les performances des arbres de décision (Yu et Liu, 2003). Il y a principalement deux causes à cela : à force de multiplier les tests, l'algorithme multiplie également le risque d'introduire des variables non-significatives dans l'arbre. Ce risque est d'autant plus élevé que les méthodes comme C4.5, très utilisées dans la communauté de l'apprentissage automatique, adoptent la construction « hurdling » (introduire une variable même si elle induit un gain nul) en misant, parfois à tort, sur le post-élagage pour éliminer les branches non-pertinentes de l'arbre.

Enfin, dernier point de différenciation, qui assure en grande partie la popularité des arbres auprès des praticiens : leur capacité à produire une connaissance simple et directement utilisable, à la portée des non-initiés. Un arbre de décision peut être lu et interprété directement ; il est possible de le traduire en base de règles sans perte d'information. A la fin des années 80, on considérait que cette méthode assurait le renouveau des systèmes experts en éliminant le goulot d'étranglement que constitue le recueil des règles (Kononenko, 1993). Cette qualité est renforcée par la possibilité qu'a l'expert d'intervenir directement dans le processus de création du modèle de prédiction.

L'appropriation de l'outil par les experts du domaine assure dans le même temps une meilleure interprétation et compréhensibilité des résultats.

5.2 Variantes

Si les arbres de décision ont connu une période faste dans les années 90 avec un très grand nombre de publications visant à en améliorer les performances, force est de constater qu'aucune avancée décisive n'a été produite en matière de taux de reconnaissance par rapport aux algorithmes de référence que constituent ID3, CHAID, CART et C4.5 (Rakotomalala, 1997 ; Lim *et al*., 2000). Il paraît illusoire aujourd'hui de prétendre produire une nouvelle technique surclassant les autres dans un schéma d'apprentissage simple sur un échantillon de données.

Le point positif est que ces nombreuses études ont permis de mieux maîtriser les propriétés des arbres. Il est possible de caractériser les variantes et le cadre dans lequel elles fonctionnent le mieux. Si elles se révèlent bien souvent performantes sur les données artificielles construites à partir de fonctions logiques, elles sont peu décisives sur des bases réelles ; elles permettent surtout d'obtenir des classifieurs de taille réduite sans dégrader les performances (Breslow et Aha, 1997).

La première catégorie de variantes vise à améliorer l'algorithme de recherche dans l'espace des solutions, soit en améliorant la méthode d'élagage, soit en procédant à une optimisation globale plus puissante (le récuit simulé, par exemple), soit en procédant à des recherches en avant lors de la segmentation (lookahead search, Ragavan et Rendell, 1993). Ces techniques permettent d'obtenir généralement un arbre plus concis au prix d'un temps de calcul plus élevé ; elles ne sont pas exemptes de tout reproche (Murthy et Salzberg, 1995). En effet à force d'optimiser sur le fichier d'apprentissage, elles peuvent ingérer des informations qui ne sont pas pertinentes ; on peut se demander à ce sujet si la recherche *gloutonne* n'est pas une bonne manière de se prémunir contre le sur-apprentissage.

La seconde catégorie de variantes cherche à modifier itérativement l'espace de recherche en produisant au fur et à mesure de nouveaux descripteurs. Connu sous le terme d'induction constructive, l'objectif est de trouver un espace de représentation plus approprié en élaborant des combinaison de variables (Pagallo et Haussler, 1990).

Enfin, dernière possibilité, modifier la forme du concept lui-même en sortant du cadre de l'arbre de décision classique. Deux types de représentation sont généralement rencontrés : les arbres obliques utilisent une combinaison linéaire des variables lors de la segmentation des sommets de l'arbre, cette variante permet de lever la contrainte « parallèle aux axes » lors du partitionnement dans l'espace de représentation ; généralement l'arbre produit est plus concis ; en revanche la lecture des règles de décision est un peu plus compliquée (Murthy et al., 1994 ; Brodley et Utgoff, 1995 ; Cantu-Paz et Kamath, 2003). Les graphes d'induction introduisent un nouvel opérateur « fusion » dans l'algorithme d'apprentissage ; le modèle de prédiction n'est donc plus un arbre mais un graphe latticiel ; l'objectif est de permettre le regroupement d'individus de mêmes caractéristiques et d'assurer ainsi une meilleure résistance à la fragmentation des données (Zighed *et al*., 1992 ; Oliver, 1993 ; Rakotomalala, 1997).

En réalité, c'est plutôt du côté de la mise en oeuvre des arbres de décision dans un cadre plus large qu'il faut trouver les principales innovations de ces dernières années. On peut citer par exemple leur utilisation dans les méthodes d'agrégation de classifieurs. Les travaux de Breiman (1996) sur le « *bagging* », de Freund et Schapiro (1997 ; 2002) sur le « *boosting* » et leur utilisation dans les arbres (Quinlan, 1996) ont montré qu'il était possible d'améliorer considérablement les performances du modèle de prédiction, au prix certes d'une moindre lisibilité de la règle de décision puisque plusieurs arbres sont en concurrence lors du classement d'un nouvel individu. Autre innovation intéressante de ces dernières années, l'extension des algorithmes d'induction aux données non-tabulaires avec les données floues et symboliques (Olaru et Wehenkel, 2003 ; Périnel, 1996). Outre le fait que ces approches ont permis d'étendre le domaine d'application des arbres de

décision, elles ont aussi permis de traiter de manière élégante le problème des données manquantes, et plus généralement, le problème des données imprécises. Enfin, s'il est possible de traiter des bases de tailles conséquentes en chargeant toutes les données en mémoire (un fichier de 500000 observations avec quelque 60 variables occupe approximativement 128 Mo en mémoire avec un codage efficace ; Rakotomalala, 2005), cela n'est plus possible dès que l'on veut accéder à des bases de données constituées de millions d'observations. Des techniques spécifiques ont alors été mises au point pour permettre le traitement de telles bases à l'aide d'un algorithme d'induction d'arbre de décision (Shafer et al., 1996).

6 Conclusion

Les arbres de décision répondent simplement à un problème de discrimination, c'est une des rares méthodes que l'on peut présenter assez rapidement à un public non spécialiste du traitement des données sans se perdre dans des formulations mathématiques délicates à appréhender. Dans ce didacticiel, nous avons voulu mettre l'accent sur les éléments clés de leur construction à partir d'un ensemble de données, puis nous avons présenté une approche – la méthode CHAID – qui permet de répondre à ces spécifications.

L'induction des arbres de décision a été la coqueluche des chercheurs dans les années 90 : les références citées dans ce didacticiel sont assez édifiantes. Ses propriétés sont maintenant bien connues et, si les tentatives pour faire évoluer la méthode sont moins nombreuses aujourd'hui, elle se positionne surtout comme une méthode de référence. Les articles proposant de nouvelles techniques d'apprentissage l'utilisent souvent dans leurs comparatifs pour situer leurs travaux. La méthode préférée en apprentissage automatique est certainement C4.5 ; la disponibilité du code source sur Internet n'est pas étrangère à ce succès.

En ce qui concerne la documentation en français, Zighed et Rakotomalala (2000) réalisent un large tour d'horizon des méthodes, qu'elles soient d'origine statistique ou de l'apprentissage automatique ; Rakotomalala (1997) produit une description plus technique mettant l'accent sur les points essentiels de la construction d'un arbre ; il existe également de nombreux didacticiels qui explicitent la construction d'un arbre. On remarquera la préférence des chercheurs pour CART en France (Nakache et Confais, 2003 ; Bardos, 2001 ; Lebart *et al*., 2000 ; Gueguen, 1994 ; Celeux et Lechevallier, 1990). En langue anglaise, les « surveys » sont également nombreux (Kohavi et Quinlan, 2002 ; Breslow et Aha, 1997 ; Murthy, 1997 ; Safavian et Landgrebe, 1991). Malgré sa relative ancienneté, la monographie CART (Breiman *et al*., 1984) reste une référence incontournable, par sa précision, son exhaustivité et le recul dont les auteurs font preuve dans les solutions qu'ils préconisent.

Références

Bardos M, *Analyse Discrimininante : Application au risque et scoring financier*, Dunod, 2001.

Bouroche J., Tenenhaus M., *Quelques méthodes de segmentation*, RAIRO, 42, 29-42, 1970.

Breiman L, Friedman J., Olshen R., Stone C., *Classification and Regression Tree*, California: Wadsworth International, 1984.

Breiman L., *Bagging Predictors*, Machine Learning, 24, 123-140, 1996.

Breslow L., Aha D., *Simplifying Decision Trees: A survey*, The Knowledge Engineering Review, 12, 1, 1-40, 1997.

Brodley C., Utgoff P., *Mutlivariate Decision Trees*, Machine Learning, 19, 1, 45-77, 1995.

Buntine W., *About the IND tree package*, Technical Report, NASA Ames Research Center, Moffet Field, California, September 1991.

Buntine W., Niblett T., *A further comparison of splitting rules for decision tree induction*, Machine Learning, 8, 75-85, 1992.

Cantu-Paz E., Kamath C., Inducing Oblique Decision Trees with Evolutionary Algrothms, IEEE Transactions on Evolutionary Computation, 7, 1, 54-69, 2003.

Catlett J., *Megainduction : machine learning on very large databases*, PhD Thesis, University of Sidney, 1991.

Celeux G., Lechevallier Y., *Méthodes de segmentation*, in Analyse Discriminante sur Variables Continues, Celeux G. éditeur, INRIA, 7, 127-147, 1990.

Chavent M., Guinot C., Lechevallier Y., Tenenhaus M., *Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine*, Revue de Statistiques Appliquées, XLVII (4), 87—99, 1999.

Chauchat J.H., Rakotomalala R., *Sampling Strategy for Building Decision Trees from Very Large Databases Comprising Many Continuous Attributes*, in Instance Selection and Construction for Data Mining, Liu H. and Motoda H. Editors, Kluwer Academic Press, 171-188, 2000.

Crémilleux B., *Classification Interactive*, Apprentissage par l'interaction, Edition Europa, 207-239, 1997.

Efron B., Tibshirani R., *Improvements on cross-validation : the .632+ bootstrap method*, Journal of the American Statistical Association, 92, 548-560, 1997.

Jensen D., Cohen P., Multiple Comparisons in Induction Algorithms, Machine Learning, 38(3), 309-338, 2000.

Dougherty J, Kohavi R., Sahami M., Supervised and unsupervised discretization of continuous attributes, in Proceedings of 12th International Conference on Machine Learning, 194-2002, 1995.

Fayyad U, Irani K., Multi-interval discretization of continuous attributes for classification learning, in Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1022-1027, 1993.

Fisher R., *The use of multiple measurements in taxonomic problems*, Annals of Eugenics, 7, 179-188, 1936.

Freund Y., Schapire R., A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 55, 1, 119-139, 1997.

Gueguen A., *Arbres de décision binaires*, in Analyse Discriminante sur Variables Qualitatives, G. Celeux et J.P. Nakache Editeurs, chapitre 7, Polytechnica, 1994.

Hand D., Manilla H., Smyth P., *Principles of data mining*, Bardford Books, 2001.

Hastie T., Tibshirani R., Friedman J., *The elements of statistical learning - Data Mining, inference and prediction*, Springer, 2001.

Hettich S., Bay S., *The UCI KDD Archive* [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science, 1999.

Hunt E.B., *Concept Learning: An Information Processing Problem*, Wiley, 1962.

Kass G., *An exploratory technique for investigating large quantities of categorical data*, Applied Statistics, 29(2), 119-127, 1980.

Kohavi R., Quinlan J., Decision-tree Discovery, in Handbook of Data Mining and Knowledge Discovery, Klösgen and Zytkow Editors, 267-276, 2002.

Kohavi R., Sommerfield D., MLC++. In Will Klosgen and Jan M. Zytkow, editors, *Handbook of Data Mining and Knowledge Discovery*, chapter 24.1.2, pages 548-553. Oxford University Press, 2002.

Kononenko I., *Inductive and bayesian learning in medical diagnosis*, Applied Artificial Intelligence, 7, 317-337, 1993.

Kononenko I., Simec E., Robnik-Sikonja M., Overcoming the myopia of inductive learning algorithm with RELIEFF, Applied Intelligence, 7(1), 39-55, 1997.

Lebart L., Morineau A., Piron M., *Statistique exploratoire multidimensionnelle*, Dunod, 2000.

Lerman I., Da Costa F., *Coefficients d'association et variables à très grand nombre de catégories dans les arbres de décision, application à l'identification de la structure secondaire de protéïnes*, Rapport INRIA, n°2803, Février1996.

Lim T., Loh W., Shih Y., A comparison of prediction accuracy, complexity and training of thirty-three old and new classification algorithms, Machine Learning Journal, 40, 203-228, 2000.

Mingers J., *An empirical comparison of selection measures for decision tree induction*, Machine Learning, 3, 319-342, 1989.

Morgan J., Sonquist J.A., *Problems in the Analysis of Survey Data, and a Proposal*, Journal of the American Statistical Association, 58:415-435, 1963.

Morgan J., Messenger R., *THAID-a sequential analysis program for the analysis of nominal scale dependent variables*, Survey Research Center, U of Michigan, 1973.

Muhlenbach F., Rakotomalala R., *Discretization of Continuous Attributes*, Encyclopedia of Data Warehousing and Mining, Wang J. editor, Idea Group Reference, 2005.

Murthy S.K., Kasif S., Salzberg S., *A System for Induction of Oblique Decision Trees*, Journal of Artificial Intelligence Research, 2, 1-32, 1994.

Murthy S., Salzberg S., *Lookahead and pathology in decision tree induction*, in Proceddings of the Fourteenth International Joint Conference on Artificial Intelligence, 1025-1031, 1995.

Murthy S., *On Growing Better Decision Trees from Data*, PhD Thesis, University of Maryland, 1997.

Nakache J-P., Confais J., *Statistique Explicative Appliquée*, Edition Tecnip, Paris, 2003.

Oates T., Jensen D., The effects of Training Set Size on Decision Tree Complexity, in Proceedings of 14th International Conference on Machine Learning, 254-262, 1997.

Olaru C., Wehenkel L., *A complete fuzzy decision tree technique*, Fuzzy Sets and Systems, 138, 2, 2003.

Oliver J., *Decision Graphs – An extension of Decision Trees*, in Fourth International Workshop on Artificial Intelligence and Statistics, 343-350, 1993.

Pagallo G., Haussler D., *Boolean feature discovery in empirical learning*, Machine Learning, 5, 71-100, 1990.

Perinel E., *Segmentation et analyse de données symboliques : application à des données probabilistes imprécises*, INRIA, 1996.

Picard C., *Graphes et questionnaires*, Gauthier-Villars, 1972.

Quinlan R., *Discovering rules by induction from large collections of examples*, D. Michie ed., Expert Systems in the Microelectronic age, pp. 168-201, 1979.

Quinlan R., *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.

Quinlan R., *Bagging, Boosting and C4.5*, in Proceedings of the Thirteenth National Conference on Artificial Intelligence, 725-730, 1996.

Ragavan H., Rendell L., *Lookahead feature construction for learning hard concepts*, in Proceedings of the Tenth International Conference on Machine Learning, 252-259, 1993.

Rakotomalala R., *Graphes d'Induction*, PhD Thesis, Université Claude Bernard Lyon 1, 1997.

Rakotomalala R., TANAGRA : Une Plate-Forme d'Expérimentation pour la Fouille de Données", Revue MODULAD, 32, 70-85, 2005.

Safavian R., Landgrebe D., A Survey of Decision Tree Classifier Mathodology, IEEE Transactions on Systems Man and Cybernetics, 21, 3, 660-674, 1991.

Schapire R., *The boosting approach to machine learning: An overview*, in MSRI Workshop on Nonlinear Estimation and Classification, 2002.

Shafer J., Agrawal R., Mehta M., SPRINT: A Scalable Parallel Classifier for Data Mining, in Proceedings of the 22nd Conference on Very Large Databases, 544-555, 1996.

Shih Y., Families of Splitting Criteria for Classification Trees, Statistics and Computing, 9(4), 309-315, 1999.

Suarez A., Lutsko J., Globally Optimal Fuzzy Decision Trees for Classification and Regression, IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(12), 1297-1311, 1999.

Terrenoire M., *Un modèle mathématique de processus d'interrogation: les pseudo-questionnaires*, PhD Thesis, Université de Grenoble, 1970.

Torgo L., *Inductive Learning of Tree-Based Regression Models*, PhD Thesis, Department of Computer Science, University of Porto, 1999.

Wallace C., Patrick J., Coding Decision Trees, Machine Learning, 11, 7-22, 1993.

Wehenkel L., On Uncertainty Measures Used for Decision Tree Induction, in Proceedings of IPMU, 413-418, 1996.

Witten I., Frank E., *Data Mining: Practical machine learning tools with Java implementations*, Morgan Kaufmann, San Francisco, 2000.

Yu L., Liu H., *Efficiently Handling Feature Redundancy in High-Dimensional Data*, in Proceedings of International Conference on Knowledge Discovery and Data Mining, 685-690, 2003.

Zighed D., Auray J., Duru G., *SIPINA : Méthode et Logiciel*, Lacassagne, 1992.

Zighed D., Rakotomalala R., *Graphes d'Induction : Apprentissage et Data Mining*, Hermès, 2000.