Classement d'objets incomplets dans un arbre de décision probabiliste

Lamis Hawarah, Ana Simonet , Michel Simonet*
*TIMC-IMAG
Institut d'Ingénierie et de l'information de Santé
Faculté de Médecine – IN3S
38700 LA TRONCHE
{Lamis.Hawarah, Ana.Simonet, Michel.Simonet}@imag.fr
http://www-timc.imag.fr

Résumé. Nous présentons une approche probabiliste pour déterminer les valeurs manquantes des objets incomplets pendant leur classement dans les arbres de décision. Cette approche est dérivée de la méthode d'apprentissage supervisé appelée Arbres d'Attributs Ordonnés proposée par Lobo et Numao en 2000, qui construit un arbre de décision pour chacun des attributs, selon un ordre croissant en fonction de l'information mutuelle entre chaque attribut et la classe. Notre approche étend la méthode de Lobo et Numao d'une part en prenant en compte les dépendances entre les attributs pour la construction des arbres d'attributs, et d'autre part en fournissant un résultat de classement d'un objet incomplet sous la forme d'une distribution de probabilités (au lieu de la classe la plus probable).

1 Introduction

Le problème des valeurs manquantes est un problème connu dans le domaine de la fouille de données, où, dans la base d'apprentissage, on rencontre des objets ayant des valeurs manquantes pour certains attributs. Nous étudions ce problème dans le cadre des arbres de décision. Un arbre de décision est construit à partir d'un ensemble d'apprentissage selon l'approche divide-and-conquer (Quinlan 1993). Une fois l'arbre construit, il est utilisé pour classer de nouveaux objets. Pour cela on parcourt l'arbre en commençant par la racine et en suivant les branches correspondant aux valeurs de l'objet, jusqu'à une feuille. La classe associée à cette feuille est la classe de cet objet. Les arbres de décision sont confrontés au problème des données manquantes, à la fois lors de leur construction et lors du classement d'objets. Lors de la construction, l'existence de valeurs manquantes pose problème pour le calcul de gain d'information, nécessaire au choix de l'attribut test, ainsi que pour la partition de l'ensemble d'apprentissage selon l'attribut test choisi. Le classement d'un objet avec des valeurs manquantes soulève également des problèmes lorsqu'un nœud correspondant à un attribut manquant est rencontré dans le parcours de l'arbre. Dans ce travail nous nous intéressons exclusivement au second problème, c'est-à-dire le classement d'objets incomplets.

Les méthodes qui traitent les valeurs manquantes dans les arbres de décision, remplacent un attribut manquant par une seule valeur, qui peut être la valeur la plus probable (Kononenko et al. 1984) ou la plus similaire (Breiman et al. 1984), etc. Ce type d'approche présente l'inconvénient d'oublier les autres valeurs possibles. Notre approche vise à une détermination probabiliste des valeurs manquantes, en prenant en compte les dépendances entre l'attribut manquant et les autres attributs de l'objet, ce qui permet d'utiliser le maximum de l'information contenue dans l'objet pour le calcul des valeurs manquantes. De plus, nous voulons un résultat sous la forme d'une distribution de probabilités plutôt que la valeur la plus probable, ce qui donne une information plus fine. Parce que les arbres de décision sont

capables de déterminer la classe d'une instance à partir des valeurs de ses attributs, on peut les utiliser pour déterminer les valeurs d'un attribut inconnu (qui joue alors le rôle de la classe) à partir des attributs dont il dépend. Dans notre travail, nous nous sommes intéressés aux méthodes qui utilisent les arbres de décision pour trouver l'attribut manquant, et en particulier à la méthode des Arbres d'Attributs Ordonnés (Lobo 1999) et (Lobo 2000).

Dans cet article, nous rappelons les principales méthodes qui traitent le problème des valeurs manquantes dans un arbre de décision, et nous détaillons la méthode des Arbres d'Attributs Ordonnés, qui sert de base à notre approche. Nous présentons ensuite notre extension à cette approche (Hawarah et al. 2004). Enfin, nous montrons comment, pour chaque attribut manquant, nous calculons une distribution de probabilités afin d'obtenir un résultat de classement probabiliste.

2 Etat de l'art

Plusieurs méthodes ont été proposées pour traiter le problème des valeurs manquantes dans un arbre de décision (White et al. 1997) et (Quinlan 1989) lors de la phase de construction de l'arbre, comme la méthode de majorité (Kononenoko et al. 1984) et la méthode de Shapiro, décrite par (Quinlan 1986). L'utilisation de l'arbre pour classer un objet avec des valeurs manquantes a aussi fait l'objet de quelques études, comme l'approche probabiliste de C4.5 (Quinlan 1993), la méthode *Lazy decision tree* (Friedman et al. 1996), et la méthode *surrogate splits* proposée par (Breiman et al. 1984), qui consiste à utiliser un autre attribut, appelé l'attribut de substitution, pour décider quelle branche (gauche ou droite) choisir pour continuer le classement. En général, lorsqu'un attribut manquant est envoyé dans un sous-arbre d'un nœud en suivant une branche déterminée par l'attribut de substitution, cela revient à compléter cet attribut manquant par la modalité[1] qui étiquette la branche choisie. Nous nous sommes intéressés à une méthode particulière, les *Arbres d'Attributs Ordonnés* (Lobo1999) et (Lobo 2000), que nous expliquons en détail dans la section suivante.

2.1 Les Arbres d'Attributs Ordonnés

Les Arbres d'Attributs Ordonnés (AAO) sont une méthode d'apprentissage supervisé proposée par Lobo et Numao pour traiter le problème des valeurs manquantes, à la fois dans les phases de construction et de classement (Lobo 1999) et (Lobo 2000). L'idée générale de cette méthode est de construire un arbre de décision, appelé arbre d'attribut, pour chaque attribut dans la base en utilisant un sous-ensemble d'apprentissage contenant les instances ayant des valeurs connues pour cet attribut. Pour un attribut donné, son arbre d'attribut est un arbre de décision dont les feuilles représentent les valeurs de cet attribut. Ces arbres sont construits selon un ordre de construction croissant en fonction de l'Information Mutuelle (IM)[2] entre chaque attribut et la classe (Shannon 1949). L'arbre d'attribut est utilisé pour

[1] Les arbres produits par la méthode CART sont des arbres binaires, où tous les tests étiquetant les nœuds de décision sont binaires. Le nombre de tests à explorer va dépendre de la nature des attributs. A un attribut binaire correspond un test binaire. A un attribut qualitatif ayant n modalités, on peut associer autant de tests qu'il y a de partitions en deux classes, soit 2^{n-1}-1 tests binaires possibles. Enfin, dans le cas d'attributs continus, il y a une infinité de tests envisageables. Dans ce cas, on découpe l'ensemble des valeurs possibles en segments.

[2] L'Information Mutuelle mesure la force de la relation entre deux attributs ou entre un attribut et la classe. L'IM entre deux attributs catégoriels X et Y est définie comme suit:

$IM(X,Y) = - \sum_{x \in Dx} P(x) \log_2 P(x) + \sum_{y \in Dy} P(y) [\sum_{x \in Dx} P(x\backslash y) \log_2 P(x\backslash y)]$

déterminer la valeur de l'attribut pour des instances où elle est inconnue. Il est utilisé dans deux cas distincts : 1) lors de la construction de l'arbre de décision, pour déterminer la valeur de l'attribut pour les instances de la base d'apprentissage où cet attribut est inconnu; 2) lors du classement d'instances incomplètes, pour déterminer la valeur de l'attribut lorsque celle-ci est manquante.

Après avoir calculé l'IM entre chaque attribut et la classe, les attributs sont ordonnés par ordre croissant d'IM. Le premier arbre d'attribut construit est celui qui correspond à l'attribut ayant l'IM minimale. Il est représenté par un seul nœud-feuille avec sa valeur la plus probable dans la base d'apprentissage. Pour les autres attributs, on fournit, à partir de l'ensemble d'apprentissage initial, le sous-ensemble d'apprentissage qui contient les instances ayant des valeurs connues pour cet attribut. Ces instances sont décrites seulement par les attributs qui ont déjà été traités (c'est à dire les attributs pour lesquels on a déjà construit les arbres d'attributs et déterminé leurs valeurs manquantes dans la base d'apprentissage). L'algorithme utilisé pour la construction est un algorithme standard de construction d'un arbre de décision. Lors d'un classement, les valeurs des attributs inconnus de l'objet sont calculées successivement, par ordre d'IM croissante. Nous présentons dans la Fig.1 la méthode AAO en utilisant un exemple pris de (Quinlan 1993); les attributs sont ordonnés par ordre croissant en fonction de l'IM : *Température, Vent, Humidité, Temps*. Les arbres sont construits en utilisant l'algorithme ID3 (Quinlan 1986) et le logiciel Weka[3]. Le nombre de cas sur chaque nœud est indiqué entre parenthèses.

Dans cet exemple, on suppose qu'il n'y a pas de valeurs manquantes dans la base d'apprentissage initial, mais les arbres sont construits à partir d'une base d'apprentissage complète et ils sont utilisés seulement pendant le classement d'objet ayant des valeurs manquantes.

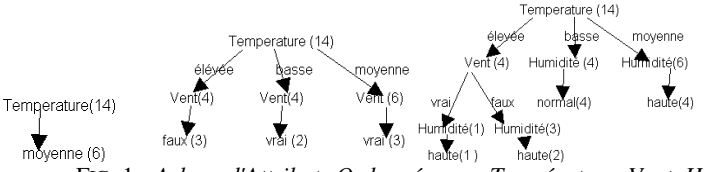

FIG. 1 - *Arbres d'Attributs Ordonnés pour Température, Vent, Humidité.*

Le premier arbre construit selon de la méthode AAO est donc l'arbre de *Température* ; il est composé d'un seul nœud, ayant pour valeur *moyenne*, qui est la valeur la plus probable. Selon l'ordre de construction imposé, l'arbre de *Vent* est construit en utilisant seulement l'attribut *Température*. Les arbres pour les attributs *Humidité* et *Temps* sont ensuite construits dans cet ordre[4]. Nous rappelons que le fait d'associer la valeur la plus probable à une feuille élimine les autres valeurs possibles. Par exemple, cette méthode remplace l'attribut *Température* par la valeur *moyenne* dans tout objet dont l'attribut *Température* est inconnu. D'autre part, pour l'arbre de *Vent* et pour la valeur *basse* de l'attribut *Température*, la base d'apprentissage dispose de deux cas où *Vent* est *faux* et deux cas où *Vent* est *vrai*. Dans cette

---

D_x et D_y sont les domaines des attributs catégoriels X et Y. P(x) et P(y) sont les probabilités de $x \in D_x$ et $y \in D_y$, respectivement. P(x\y) est la probabilité conditionnelle que X prenne la valeur x sachant que Y est connu et prend la valeur y.

[3] www.cs.waikato.ac.nz/ml/weka/index.html

[4] Dans cet exemple simple destiné à illustrer le fonctionnement de la méthode AAO et les extensions proposées dans la suite de l'article, on n'a pas présenté tous les arbres construits. Par exemple, l'arbre de Temps est construit en utilisant les attributs Température, Vent et Humidité, mais il n'est pas présenté ici.

situation, ID3 a choisi arbitrairement la valeur *vrai*. Enfin, nous rappelons que l'ordre imposé par cette méthode ne garantit pas que l'arbre d'un attribut soit construit à partir des attributs dont il dépend. L'étude qui a été faite par Lobo et Numao (Lobo 2001) a montré que les relations entre attributs d'une base d'apprentissage doivent vérifier certaines conditions[5] pour que la méthode AAO soit applicable.

Pour traiter le problème de valeurs inconnues de manière probabiliste en prenant en compte les dépendances entre les attributs, nous faisons deux propositions : les Arbres d'Attributs Ordonnés Probabilistes (AAOP), qui étendent les AAO par la prise en compte de la distribution des fréquences des classes dans les feuilles, et les Arbres d'Attributs Probabilistes (AAP), qui sont construits en utilisant les dépendances entre attributs.

2.2 Vers une approche probabiliste

Un arbre de décision idéal est un arbre où toutes les instances arrivant à une feuille appartiennent à la même classe. Cette catégorie d'arbres, rarement rencontrée dans le monde réel, est représentée par un arbre semblable à celui de la Fig.2. Plus généralement, un arbre de décision se présente sous une forme où les instances d'une feuille appartiennent à plusieurs classes. Dans ce cas, classiquement on associe à chaque feuille la classe la plus probable. (Breiman et al. 1984) ont proposé de construire des *Class Probability Trees* où on associe à chaque feuille F la probabilité de chaque classe J sur F; P(J\F) avec J=1,..,n. Ainsi, dans le cas du diagnostic en médecine, pour un patient qui pourrait avoir trois maladies m_1, m_2, m_3, il serait préférable d'estimer les probabilités relatives d'avoir m_1, m_2, m_3 au lieu de lui affecter une seule maladie, et ceci même s'il n'y a pas d'information manquante.

Quinlan a également proposé d'utiliser les probabilités pour traiter le problème des valeurs manquantes dans les phases de construction et de classement (Quinlan 1986), (Quinlan 1990) et (Quinlan 1993). Son approche consiste à associer un poids à chaque valeur d'un attribut. Pour un attribut connu, le poids est 1 pour la valeur de l'attribut et 0 pour toutes ses autres valeurs. Pour un attribut inconnu, le poids associé à chacune de ses valeurs est sa fréquence dans le sous-ensemble d'apprentissage correspondant au nœud de cet attribut. Dans ce cas, lors de la construction de l'arbre de décision, Quinlan associe à chaque feuille F la classe la plus probable, notée C_j. Cependant, il conserve le nombre total de cas arrivant à F ainsi que des couples (C_i, nb_i) où, nb_i est le nombre de cas appartenant à la classe C_i ($C_j \neq C_i$) qui arrivent à F. Ces informations lui permettent, lors d'un classement d'un objet avec des valeurs manquantes, de calculer la probabilité de chaque classe.

Selon (Quinlan 1990), pour un seul chemin (une seule règle de classement) de la racine de l'arbre jusqu'à une feuille F passant par les branches B_1, B_2, .., B_L, où chaque branche correspond à une valeur d'attribut test (le résultat d'un nœud), la probabilité qu'un objet E arrive à une feuille F (c'est à dire qu'il passe par les branches B_1, B_2, .., B_L) est :

$P_E(F) = P_E(B_1, B_2, .., B_L) = P_E(B_1) * P_E(B_2\B_1) * P_E(B_3\B_1,B_2) *\ldots* P_E(B_L\ B_1, B_2, ..,B_{L-1})$

Si la valeur de chaque attribut est connue, chacune des probabilités précédentes est 0 ou 1. Si la valeur de l'attribut correspondant à la branche B_i n'est pas connue, sa probabilité est calculée par $P(B_i\ B_1, B_2,\ldots B_{i-1})$ à partir de l'ensemble d'apprentissage initial, i.e., la proportion des cas (instances) arrivés au $i^{ième}$ test qui prennent la branche B_i.

Parce qu'un cas E avec des valeurs manquantes peut appartenir à plusieurs feuilles, la probabilité que le cas E appartienne à une classe C est : $\sum_F P_E(F)\ P(C\F)$

[5] Les attributs qui ont des relations faibles avec la classe devraient avoir des relations fortes avec le reste des attributs dans la base. En général, une base d'apprentissage dont les corrélations entre les attributs sont fortes est plus favorable pour l'application de cette méthode.

Par exemple, si on veut classer une instance où *Temps* est *ensoleillé* et *Humidité* est inconnue, la probabilité que cette instance appartienne à la classe A est (Fig.2) :
P(A) = P(A\ensoleillé, haute) * P(ensoleillé) * P(haute\ensoleillé)
+ P(A\ Pluvieux, vrai) * P(Pluvieux, vrai) =1*1*3/5= 0.6.
La partie P(A\ Pluvieux, vrai) * P(Pluvieux, vrai) est égale à 0 car le *Temps* est *ensoleillé* alors P(Pluvieux) = 0.

Remarquons que la probabilité de *haute* est calculée sachant seulement *ensoleillé*, car le *Temps* est le nœud-père de *Humidité*. Ainsi, le fait que *Humidité* dépend de *Température* n'est pas pris en compte. En prenant en compte la corrélation qui existe entre *Humidité* et *Température*, le résultat sera vraisemblablement meilleur.

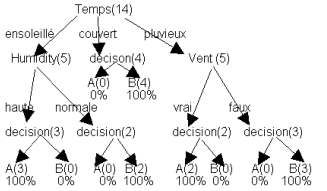

FIG.2 - *Arbre de décision probabiliste.*

3 Approche probabiliste

Nous étudions le problème des valeurs manquantes pendant le classement. Nous étendons la méthode des Arbres d'Attributs Ordonnés (Lobo 1999) et (Lobo 2000) en construisant pour chaque attribut un arbre de décision probabiliste au lieu d'un arbre de décision classique (§3.1). Pour cela, nous nous inspirons de (Breiman et al. 1984) (Quinaln 1990) et (Quinlan1993) pour la construction des arbres d'attributs probabilistes. Une deuxième extension, que nous appelons Arbres d'Attributs Probabilistes, est proposée afin de prendre en compte les dépendances entre les attributs au lieu de l'ordre d'IM croissante, lors de la construction des arbres d'attributs (Hawarah et al. 2004).

3.1 Arbres d'Attributs Ordonnés Probabilistes

Cette première proposition est une extension de la méthode Arbres d'Attributs Ordonnés (Lobo 1999) et (Lobo 2000). Elle consiste à construire pour chaque attribut un arbre d'attribut selon la méthode de Lobo. Cependant, contrairement à Lobo, qui, en suivant la méthodologie classique, associe à chaque feuille la valeur la plus probable, nous proposons de conserver dans chaque feuille d'un arbre d'attribut la distribution des fréquences des valeurs de l'attribut en question. Cette distribution de probabilités permet de déterminer le classement probabiliste des valeurs d'un attribut manquant. En conséquence, elle permet le classement probabiliste d'un objet avec des attributs manquants. On appelle cette proposition Arbres d'Attributs Ordonnés Probabilistes (AAOP). Le résultat du classement permettant de déterminer une valeur manquante est une distribution de probabilités des valeurs de l'attribut. En conséquence, le classement d'un objet incomplet en utilisant les AAOPs est une distribution probabiliste de classe au lieu d'une seule valeur de classe. La Fig.3 montre les AAOPs de *Température*, *Vent*, *Humidité* :

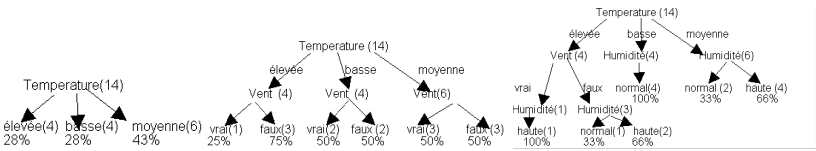

FIG.3 - *Arbres d'Attributs Ordonnés Probabilistes pour Température, Vent, Humidité.*

Ces arbres sont utilisés pendant le classement d'objets avec des valeurs d'attributs manquantes. Ainsi, si on classe un objet où *Temps* est *ensoleillé*, *Vent* est *faux*, *Température* est *élevée* mais *Humidité* est inconnue, les probabilités des valeurs de l'attribut *Humidité* sont calculées à partir de son arbre d'attribut dans la Fig.3 (à droite). Dans notre exemple, la distribution de probabilités des valeurs de l'attribut *Humidité* sont : *normal* avec la probabilité 0.33 et *haute* avec la probabilité 0.66. Contrairement aux arbres d'attributs ordonnés de Lobo, les AAOPs ont l'avantage de permettre d'aboutir à des résultats probabilistes en utilisant un calcul identique à celui proposé par (Quinlan 1990). Cependant, de notre point de vue, ces AAOPs continuent de poser problème car les attributs pris en compte pour la construction d'un arbre d'attribut sont choisis en fonction de leur IM par rapport à la classe. En particulier, seuls les attributs ayant, par rapport à la classe, un IM inférieur à celui de l'attribut courant, pour lequel on construit l'arbre d'attribut, sont pris en compte. En conséquence, il n'y a aucune garantie que ces attributs dépendent de l'attribut courant. Or, ce sont ces attributs qui détiennent l'information la plus pertinente pour le calcul de la distribution de probabilités de cet attribut. Les Arbres d'Attributs Probabilistes (AAP), présentés ci-dessous, constituent notre deuxième proposition d'extension des arbres d'attributs de Lobo. Contrairement aux AAOP, les AAP prennent en compte les dépendances entre attributs.

3.2 Arbres d'Attributs Probabilistes

La méthodologie des arbres d'attributs probabilistes (AAP) que nous proposons ici est une méthodologie qui, pour chaque attribut, construit un arbre d'attribut probabiliste en utilisant les attributs dont il dépend. Afin de déterminer les dépendances entre les attributs, nous calculons l'IM entre chaque couple d'attributs de la base. En effet, l'IM entre deux attributs est la réduction de l'incertitude sur un attribut sachant l'autre. Ainsi, pour un attribut A_i, les attributs dont il dépend sont calculés par l'expression :

$Dep(A_i)= \{A_j \mid IM(A_i, A_j)>0.01$[6]$\}$

Dans une deuxième étape, un arbre de décision probabiliste est construit pour chaque attribut en prenant en compte les attributs dont il dépend. L'application de cette méthodologie à la base extraite de (Quinlan 1993) détermine que :

IM(Humidité, Température) = 0.37465, IM(Humidité, Temps) = 0.02074
IM(Temps, Température) = 0.2377, IM(Température, Vent) = 0.039
IM(Humidité, Vent) = 0, IM(Vent, Temps) =0.005

En conséquence, les dépendances prises en compte sont :

Dep(Humidité) = {Température, Temps}
Dep(Temps) = {Température, Humidité}
Dep(Température) = {Humidité, Temps, Vent}
Dep(Vent) = { Température }

[6] On a choisi le degré de dépendance 0.01 arbitrairement.

L'arbre d'attribut probabiliste pour l'attribut *Humidité* est donné dans la Fig.4 et l'arbre de *Temps* est donné dans la Fig.5 :

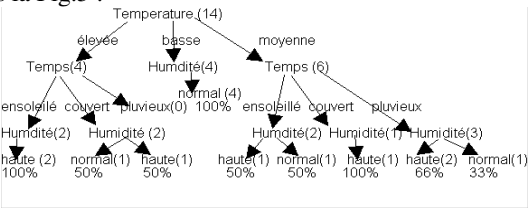

FIG. 4 - *Arbre d'Attribut Probabiliste de Humidité.*

Si *Température* est *moyenne* et *Temps* est *ensoleillé*, alors la probabilité que *Humidité* soit *haute* est 0.5 et la probabilité qu'elle soit *normale* est 0.5. On remarque ici que ces probabilités sont calculées en prenant en compte les valeurs de *Température* et *Temps*. Cette approche est meilleure que AAOP car elle prend en compte les dépendances qui existent entre les attributs connus dans l'objet à classer et l'attribut dont la valeur est manquante. Les contraintes (Lobo 2001) que les attributs d'une base d'apprentissage devraient vérifier pour que la méthode AAO soit applicable n'ont pas de raison d'être dans les AAPs car il n'y a pas d'ordre de construction imposé.

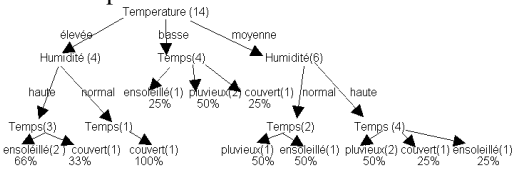

FIG. 5 - *Arbre d'Attribut Probabiliste de Temps.*

3.3 Problèmes rencontrés avec l'approche AAP

L'utilisation des AAP pose un certain nombre de problèmes :
1) *Cycle*. Lorsque deux attributs mutuellement dépendants sont manquants, le problème du calcul se pose. Pour casser ce cycle, nous proposons la solution suivante : on calcule d'abord la probabilité de l'attribut le moins dépendant de la classe à partir de son arbre d'attribut ordonné probabiliste AAOP, puis les distributions de probabilités de l'autre attribut à partir de son arbre d'attribut probabiliste AAP.
2) *Feuilles indéterminées*. On rencontre ce problème quand aucune instance n'est associée à une feuille. Dans ce cas, on ne sait pas quelle classe on doit associer à cette feuille. Par exemple, prenons l'AAP de *l'Humidité* donné dans la Fig.4. Aucune instance ayant pour l'attribut *Température* la valeur *élevée* et le *Temps* la valeur *pluvieux* n'a été retrouvée dans la base d'apprentissage. Dans ce cas, nous utilisons l'AAOP de l'attribut *Humidité*, ce qui fournit les probabilités suivantes : P(Humidité =haute) = 0.66, P(Humidité = normale) = 0.33

4 Le classement probabiliste

Pour classer un nouvel objet, on parcourt l'arbre de décision final de la racine jusqu'à une feuille en suivant les branches correspondant aux valeurs de l'objet à classer. Si on rencontre un attribut inconnu, on appelle son arbre d'attribut probabiliste pour récupérer la distribution

de probabilité de ses valeurs. Ce résultat est utilisé dans l'arbre de décision final pour trouver la distribution de probabilité de la classe.

Nous présentons le processus de classement à l'aide d'un exemple : étant donné un objet où *Température* a pour valeur *élevée* et *Vent* a pour valeur *faux*, la probabilité que l'objet appartient à la classe A est calculée à partir de l'arbre de décision donné dans la Fig.2, ce qui correspond à la probabilité totale de A :

P(A)=P(A\ensoleillé,haute)*P(ensoleillé,haute)+P(A\pluvieux,vrai)*P(pluvieux,vrai)

Comme *Vent* a pour valeur *faux*:

P(Vent =vrai) = 0, P(A\pluvieux, vrai) * P(pluvieux, vrai)=0

P(A\ensoleillé, haute)=1 C'est la probabilité de la classe A sur cette feuille, tous les cas arrivant à cette feuille appartiennent à la classe A (voir Fig.2). On a alors :

P(A) = P(A\ensoleillé, haute)* P(ensoleillé, haute)= P(ensoleillé, haute)

Pour calculer P(ensoleillé, haute) on va distinguer trois cas :

1) Si *Temps* est *ensoleillé* et *Humidité* est manquante, on a :

P(ensoleillé, haute) = P(haute) * P(ensoleillé \haute)

P(ensoleillé \haute) =1 car P(ensoleillé) = 1

P(haute) est calculé à partir de son AAP donné dans la Fig.4 ; La racine de l'arbre est *Température*, et sa valeur dans l'objet à classer est *élevée*. En descendant dans l'arbre et en suivant la branche correspondant à la valeur *ensoleillé* pour l'attribut *Temps*, on arrive à une feuille où P(Humidité = haute)= 1, P(Humidité= normal)=0

P(A) = P(ensoleillé, haute)= P(haute) * P(ensoleillé \haute)=1*1 = 1

2) Si *Temps* est inconnu et *Humidité* est *haute* :

P(ensoleillé, haute) = P(ensoleillé)* P(haute\ ensoleillé)

P(haute\ ensoleillé) = 1 car P(haute) = 1

P(ensoleillé) est calculé à partir de son AAP construit en utilisant *Température*, *Humidité*. L'arbre est donné dans la Fig.5 :

P(ensoleillé) = P(ensoleillé\haute, élevé) * P(haute, élevé) = 0.66

P(haute, élevé) = 1 car P(haute)=1 et P(élevé)=1

P(A) = P(ensoleillé, haute)= P(ensoleillé)* P(haute\ ensoleillé) =0.66*1 = 0.66

3) Si *Temps* et *Humidité* sont inconnus, on a un cycle:

P(ensoleillé, haute) = P(haute) * P(ensoleillé\ haute)

Humidité étant l'attribut le moins dépendant de la classe, on calcule la probabilité que *Humidité* soit *haute* à partir de son AAOP donné dans la Fig.3 :

P(haute) = P(haute\faux, élevé) * P(faux, élevé) = 0.66

P(faux, élevé) =1 car on sait que *Vent* est *faux*, et *Température* est *élevée*.

La probabilité que *Temps* soit *ensoleillé* sachant que *Humidité* est *haute* est calculée à partir de son AAP donné dans la Fig.5 :

$P(\text{ensoleillé}\backslash\text{haute}) = \sum P(\text{ensoleillé}\backslash\text{haute}, A_i) * P(A_i\backslash\text{haute})$ (1)[7]

= P(ensoleillé\haute, élevée) * P(élevée\haute)

+ P(ensoleillé\ haute, moyenne) * P(moyenne\ haute)

= P(ensoleillé\haute, élevée) * P(élevée\haute) = 0.66*1= 0.66

Car *Température* est *élevée* alors P(élevée\haute)=P(élevée) =1, P(moyenne\ haute) = 0

P(A) = P(ensoleillé, haute)= P(haute) * P(ensoleillé \haute)= 0.66 * 0.66 = 0.4356

Nous calculons la probabilité que l'objet appartienne à la classe B de la même manière que précédemment. Nous pouvons également calculer cette probabilité comme suit :

[7] Pour prouver que $P(B\backslash C) = \sum P(B\backslash A_i, C) * P(A_i \backslash C)$ donné dans la relation (1) :
$P(B) = \sum P(B\backslash A_i) * P(A_i) = \sum P(B, A_i) \rightarrow P(B\backslash C) = \sum P(B, A_i \backslash C) = \sum P(B, A_i, C) / P(C)$
$= \sum P(C) * P(A_i \backslash C) * P(B\backslash A_i, C) / P(C) = \sum P(B\backslash A_i, C) P(A_i \backslash C)$

$P(B) = 1-P(A)$

5 Conclusion et Perspectives

Dans le monde réel, un objet incomplet peut potentiellement appartenir à plusieurs classes et devrait donc être associé à plusieurs feuilles dans l'arbre de décision. Dans un domaine critique comme la médecine, prendre une seule décision quand il y a manque d'information peut être dangereux. Notre approche consiste à utiliser la notion de probabilité pour résoudre le problème des valeurs manquantes dans les données. Nous avons proposé de remplacer une valeur manquante par une distribution de probabilités et un objet incomplet par une distribution de probabilités de classe.

La première expérimentation concernant la construction des arbres d'attributs probabilistes a été faite sur deux étapes : nous avons étendu l'algorithme ID3 (Quinlan 1986) qui construit des arbres de décision sans élagage pour avoir un algorithme qui construit des arbres de décision probabilistes (ID3-Probabiliste) et nous avons développé un programme en Java qui utilise ID3-Probabiliste pour construire à partir d'une base d'apprentissage complète un arbre de décision probabiliste (AAP) et un arbre de décision ordonné probabiliste (AAOP) pour chaque attribut dans la base, ainsi que l'arbre de décision probabiliste final qui correspond à la base entière.

A partir de ces arbres, on peut déduire les relations entre les attributs. Par exemple, si les attributs sont indépendants comme dans la base contact-lenses (Blake 1998), chacun de ses arbres est un seul nœud-feuille avec sa distribution de probabilité. Par contre, chaque arbre construit à partir de cette base selon AAO est un seul nœud-feuille avec sa valeur la plus probable. Dans le cas, où tous les valeurs sont équiprobables, l'algorithme utilisé par AAO (comme ID3 ou C4.5) choisit une valeur aléatoirement. D'autre part, nous remarquons que quelques feuilles contiennent une valeur de classe avec la probabilité 1, mais le nombre d'instances arrivant à cette feuille est faible (inférieur à 5). Contrairement à d'autres tests statistiques, il n'existe pas pour les arbres de décision de seuil reconnu sur le nombre d'individus nécessaires pour que le résultat soit significatif. Selon (Labarere et al. 2003), il est souvent considéré en informatique que 5 individus par feuille sont suffisants pour la valider alors qu'en médecine il serait nécessaire d'avoir au moins 20 individus par feuille, faute de quoi le résultat risque de ne pas être significatif. Dans notre cas, nous trouvons que l'augmentation du degré de dépendance peut éliminer quelques attributs non significatifs ce qui conduit également à augmenter le nombre de cas par feuille. Une étude est en cours pour valider, avec des experts du domaine, les résultats de notre approche appliquée à une base de données médicale sur l'apnée du sommeil.

Références

Blake C.L. and Merz C.J. (1998): UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Breiman L., Friedman J.H., Olshen R.A. and Stone C.J. (1984): Classification and Regression Trees, Wadsworth and Brooks.

Kononenko I., Bratko I. and Roskar E. (1984): Experiments in Automatic Learning of Medical Diagnostic Rules, Technical Report, Jozef Stefan Institute, Ljubljana, Yugoslavia.

Friedman J.H., Kohavi R. and Yun Y. (1996): Lazy Decision Trees, AAAI.

Hawarah L., Simonet A. and Simonet M. (2004): Une approche probabiliste pour le classement d'objets incomplets dans un arbre de décision, EGC 2004, poster.

Hawarah L., Simonet A. and Simonet M. (2004): A probabilistic approach to classify incomplete objects using decision trees, Spain, DEXA. Lecture Notes in Computer Science 3180 pp. 549-558.

Labarere J., Bosson J-L. and Robert C. (2003): Utilisation des arbres d'induction en épidémiologie : Principes et exemple d'application à l'analyse d'une enquête de pratiques de prévention de la maladie thrombo-embolique veineuse. Congrès épidémiologie et biométrie, Lille.

Lobo O.O. and Numao M. (1999): Ordered estimation of missing values, Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Lobo O.O. and Numao M. (2000): Ordered estimation of missing values for propositional learning, Japanese Society for Artificial Intelligence, JSAI, vol.15, no.1.

Lobo O.O. and Numao M. (2001): Suitable Domains for Using Ordered Attribute Trees to Impute Missing Values. IEICE TRANS. INF. & SYST., Vol.E84-D, NO.2.

Quinlan J.R. (1989): Unknown attribute values in induction. Proc. Sixth International Machine Learning Workshop, Morgan Kaufmann.

Quinlan J.R. (1986): Induction of decision trees. Machine Learning, 1, pp.81-106.

Quinlan J.R. (1990): Probabilistic decision trees, in Machine Learning: an Artificial Intelligence Approach, ed.Y.Kodratoff, vol.3, Morgan Kaufmann, San Mateo, pp.140-152.

Quinlan. J.R (1993) : C4.5 Programs for Machine Learning, Morgan Kaufmann.

Shannon C.E., Weaver W. (1949): Théorie Mathématique de la communication, les classiques des sciences humaines.

White A.P. Liu W.Z., Thompson S.G. and Bramer M.A. (1997): Techniques for Dealing with Missing Values in Classification. LNCS 1280, pp. 527-536.