Arbres de décision sur des données de type intervalle : évaluation et comparaison

Chérif Mballo [*,**] & Edwin Diday [**]

[*] ESIEA Recherche, 38 Rue des Docteurs Calmette et Guérin 53000 Laval France
mballo@esiea-ouest.fr

[**] LISE-CEREMADE, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris cedex 16, France
diday@ceremade.dauphine.fr

Résumé. Le critère de découpage binaire de Kolmogorov-Smirnov nécessite un ordre total des valeurs prises par les variables explicatives. Nous pouvons ordonner des intervalles fermés bornés de nombres réels de différentes façons. Notre contribution dans cet article consiste à évaluer et à comparer des arbres de décision obtenus sur des données de type intervalle à l'aide du critère de découpage binaire de Kolmogorov-Smirnov étendu à ce type de données (Mballo et al. 2004). Pour ce faire, nous axons notre attention sur le taux d'erreur mesuré sur l'échantillon de test. Pour estimer ce paramètre, nous divisons aléatoirement chaque base de données en deux parties égales en terme d'effectif (à un objet près) pour construire deux arbres. Ces deux arbres sont d'abord testés par un même échantillon puis par deux échantillons différents.

1 Introduction

Dans le domaine de la discrimination par arbre de décision binaire, les variables explicatives sont souvent quantitatives ou qualitatives classiques. Le critère de découpage binaire de Kolmogorov-Smirnov a été introduit par (Friedman 1977 ; Utgoff et Clouse 1996) pour une partition binaire à expliquer avec des variables explicatives quantitatives classiques. Ce critère a été étendu aux variables explicatives qualitatives classiques par (Asseraf 1998). Cependant, depuis quelques années, avec l'avènement de l'analyse des données symboliques (Bock et Diday 2000), on assiste à la mise au point de méthodes de construction d'arbres de décision sur des données symboliques, notamment de type intervalle et histogramme (Périnel 1996 ; Yapo 2002). Ces auteurs utilisent les critères de découpage classiques (entropie, Gini, gain ratio, likelihood) pour construire l'arbre de décision. Nous privilégions ici la méthode basée sur le critère de découpage binaire de Kolmogorov-Smirnov. Ce critère est basé sur un ordre total des valeurs prises par les variables explicatives. Nous pouvons ordonner des intervalles fermés bornés de $\Re$ (ensemble des nombres réels) de différentes façons (Diday et al. 2003) et chacune des relations d'ordre proposées est totale sur l'ensemble des intervalles fermés bornés. Nous présentons ici une approche exploratoire de construction d'arbres de décision. Cette approche consiste à construire un arbre pour chaque ordre et à comparer ces arbres obtenus selon le taux d'erreur réel mesuré sur l'échantillon de test. Pour estimer ce paramètre, nous utilisons l'approche suivante : chaque base de données utilisée est divisée aléatoirement en deux parties pour construire deux arbres et ces arbres sont d'abord testés par un même échantillon puis par deux échantillons différents (section 5). Comme les

échantillons de test et d'apprentissage sont indépendants et pris aléatoirement dans le fichier de données, la précision de l'estimation ne dépend que du nombre d'objets de l'ensemble de test et de la valeur du risque réel (Cornuéjols et Miclet 2002).

2 Présentation du type de données des variables explicatives

Désignons par $\Im$ l'ensemble des intervalles fermés et bornés de $\Re$ et par Ω l'ensemble des individus de la population. Pour un intervalle fermé x, notons r(x) sa borne supérieure et l(x) sa borne inférieure. Une variable de type intervalle (Bock et al. 2000) X est une application de $\Omega \longrightarrow \Im$ telle que, pour tout $w \in \Omega$, il existe deux nombres réels α et β (avec $\alpha \leq \beta$) tels que $X(w) = [\alpha, \beta]$.

2.1 Ordonner des intervalles (Diday et al. 2003)

Différentes méthodes permettent d'ordonner des intervalles selon leur positionnement. Un ordre d'intervalles est une relation anti-réflexive et transitive (un ordre d'intervalles est alors un ordre strict du fait que la relation est anti-réflexive).

2.1.1 Intervalles disjoints

Soit D un ensemble d'intervalles fermés disjoints ($D \subset \Im$). Désignons par xRy pour indiquer que l'intervalle x est « *strictement avant* » l'intervalle y (avec x, y des éléments de D). Nous utilisons le qualificatif « *strictement* » pour faire ressortir le fait que les intervalles sont disjoints. Mathématiquement, la relation R sur D se définit par : $xRy \Leftrightarrow r(x) < l(y)$. Cette relation R est transitive et anti-réflexive et définit ainsi un ordre total strict sur D.

2.1.2 Intervalles non disjoints

Par analogie au cas disjoint, nous utilisons ici le qualificatif « *presque* » pour faire ressortir le fait que les intervalles sont non disjoints. Soient x et y deux intervalles non disjoints. Suivant leur positionnement, nous distinguons deux cas :

– *Ordonner* « *par la borne inférieure* » : désignons par xIy : l'intervalle x est « *presque avant* » l'intervalle y. Nous distinguons deux cas: si les deux bornes inférieures sont égales, alors l'ordre sera déterminé par la position des bornes supérieures et si les deux bornes inférieures sont différentes, alors l'ordre sera déterminé par la position de ces bornes inférieures. Mathématiquement, la relation I se définit par :

si $l(x) = l(y)$ alors $xIy \Leftrightarrow r(x) < r(y)$ et si $l(x) \neq l(y)$ alors $xIy \Leftrightarrow l(x) < l(y)$.

La relation I est transitive et anti-réflexive et définit un ordre total strict sur $\Im$.

– *Ordonner* « *par la borne supérieure* » : désignons par xSy : l'intervalle y est « *presque après* » l'intervalle x. Nous distinguons deux cas comme précédemment : si les deux bornes supérieures sont égales, alors l'ordre sera déterminé par la position des bornes inférieures et si les deux bornes supérieures sont différentes, alors l'ordre sera déterminé par la position de ces bornes supérieures. Mathématiquement, la relation S se définit par :

si $r(x) = r(y)$ alors $xSy \Leftrightarrow l(x) < l(y)$ et si $r(x) \neq r(y)$ alors $xSy \Leftrightarrow r(x) < r(y)$.

La relation *S* est transitive et anti-réflexive et définit un ordre total strict sur $\Im$.

Remarque : Soient deux intervalles *x* et *y* tels que l(*x*) < l(*y*) et r(*y*) < r(*x*) (l'intervalle *y* est strictement inclus dans l'intervalle *x*).

En ordonnant par la relation *I*, on dira que *x* est « *presque avant* » *y* (*xIy*) car l(*x*) < l(*y*) et en ordonnant par la relation *S*, *x* est « *presque après* » *y* (*ySx*) car r(*x*) > r(*y*). Pour n'importe quelle autre configuration de deux intervalles *x* et *y* (mis à part ce cas d'inclusion stricte), si *x* est « *presque avant* » *y* (*xIy*), alors *y* est « *presque après* » *x* (*xSy*). Cela veut dire que *I* et *S* indiquent une même relation de précédence entre deux intervalles. La principale différence entre les deux relations *I* et *S* est le cas de l'inclusion stricte entre intervalles.

Une méthode alternative simple consisterait à ordonner les intervalles par leurs centres.

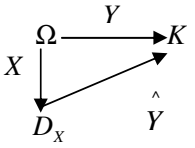
2.2 Présentation du tableau de données en entrée

L'ensemble Ω est constitué d'objets (individus) destinés à être classés à l'aide d'un arbre de décision utilisant le critère de découpage binaire de Kolmogorov-Smirnov. Au départ, l'ensemble Ω est muni d'une partition en k classes (les classes a priori). Désignons par $K = \{1,2,...,k\}$ l'ensemble de ces classes et par Y la variable déterminant l'appartenance d'un objet $w \in \Omega$ à l'une de ces classes (variable classe). Cette variable définit une fonction de classement Y de $\Omega \longrightarrow K$ telle que $Y(w) = i \in K$ pour tout $w \in \Omega$. La variable Y est la variable à expliquer. Chaque objet est aussi décrit par un vecteur X de p variables intervalles $(X_1, X_2, ..., X_P)$ appelées variables explicatives ou prédicteurs. Chaque variable explicative X_j a pour espace de description $D_{X_j} \subset \Im$ (D_{X_j} est un ensemble fini d'intervalles fermés bornés de $\Re$). Le vecteur X est alors défini de $\Omega \longrightarrow D_X = \prod_{j=1}^{p} D_{X_j}$ telle que $X(w) = (x_1, x_2, ..., x_p) \in D_X$ pour tout $w \in \Omega$ (x_j est un intervalle fermé borné de $\Re$ pour tout $j = 1,..., p$). Avec ces applications X et Y, un objet $w \in \Omega$ est alors modélisé par le couple $(x, y) \in D_X$ x K. Au départ du processus de construction d'un arbre de décision par le critère de Kolmogorov-Smirnov, notre tableau de données en entrée se présente alors comme le tableau (TAB 1) ci-dessous.

Variables / *Individus (objets)*	X_1	……	X_p	Y
Objet_1	$[a_1, b_1]$	……..	$[c_1, d_1]$	1
…….	…….	…….		…..
Objet_n	$[a_n, b_n]$	…….	$[c_n, d_n]$	j

TAB 1 – *Format standard du tableau de données en entrée*

A partir d'un tel tableau, nous pouvons appliquer les méthodes exposées précédemment (section 2.1) pour ordonner les intervalles afin de dérouler l'algorithme de Kolmogorov-Smirnov pour la construction d'arbres binaires de décision. A l'arrêt de la construction de l'arbre de décision, chaque objet est affecté à une classe (classe majoritaire du nœud terminal auquel appartient l'objet). Si on note par $\hat{Y}$ cette fonction permettant d'affecter une classe à un objet à l'arrêt du processus de construction de l'arbre ($\hat{Y}$ est définie sur l'ensemble d'arrivée D_X de X à valeurs dans l'ensemble d'arrivée K de Y), alors le problème se présente sous la forme suivante :



L'objectif est alors de rechercher (ou de s'approcher au mieux) de la situation idéale suivante : $Y = \hat{Y} \circ X$ où « o » désigne la composition des applications.

3 Le critère de découpage binaire de Kolmogorov-Smirnov

Dans cette section, nous utiliserons certaines notations de la section 2.2. Pour construire un arbre de décision, nous avons besoin d'un critère d'évaluation de la coupure d'un nœud en deux nœuds fils. Le critère de Kolmogorov-Smirnov (noté KS dans la suite) a été introduit par (Friedman 1977) pour une partition binaire à expliquer (les variables explicatives sont quantitatives classiques). Pour utiliser ce critère KS, Friedman suppose que les coûts de mauvais classement et les probabilités a priori des classes sont inversement proportionnels (pour ne pas faire des estimations des probabilités a priori). Ce critère KS permet de séparer une population en deux sous-populations plus homogènes en se basant sur les deux fonctions de répartition des classes a priori (cas de deux classes) pour chaque variable explicative. Dans le cas où le nombre de classes a priori est k avec k>2, les fonctions de répartition sont induites par le regroupement de ces k classes a priori en deux groupes appelés super classes par la méthode « twoing splitting process » (Breiman et al. 1984). Il y a ($2^{k-1}-1$) possibilités de regrouper *k* classes en deux groupes, mais cette complexité exponentielle a été réduite à une complexité polynomiale par (Asseraf 1998), qui a d'ailleurs étendu ce critère KS au cas où les variables explicatives sont qualitatives classiques. Cette méthode « twoing splitting process » est utilisée pour générer deux super classes G_1 et G_2 auxquelles sont associées deux fonctions de répartitions F_1 et F_2 d'une variable aléatoire (variable explicative). La fonction de répartition a pour rôle de compter toutes les valeurs inférieures à un certain seuil. Les deux fonctions de répartition théoriques F_1 et F_2 ne sont pas connues en pratique. S'il n'y a pas d'ordre sur les observations, on ne peut pas estimer la fonction de répartition théorique F_i par la fonction de répartition empirique $\hat{F}_i$ (i = 1,2) comme dans le cas continu. Par ailleurs, on peut estimer car on a un ordre des observations (intervalles

fermés bornés de $\Re$) et l'ensemble $\{y \in D_{X_j} / y \leq x\} \cap \{y \in D_{X_j} / y \in G_i\}$ est toujours fini en pratique ($j = 1,2,..., p$; $i = 1,2$ et « $\leq$ » un ordre d'intervalles). Selon l'ordre choisi pour ordonner les observations d'une variable explicative de type intervalle X_j, la fonction de répartition empirique $\hat{F}_i^j$ (i = 1,2) qui estime F_i^j en $x \in D_{X_j}$ est donnée par :

$$\hat{F}_i^j(x) = \frac{Cardinal\left(\{y \in D_{X_j} / y \leq x\} \cap \{y \in D_{X_j} / y \in G_i\}\right)}{Cardinal\left(\{y \in D_{X_j} / y \in G_i\}\right)}$$

Ce sont les proportions réelles des observations pour chaque variable explicative X_j relative à une classe a priori (ou super classe) G_i. Ainsi le critère KS est défini par :

$$KS = \sup_{x \in D_{X_j}} \left| \hat{F}_1^j(x) - \hat{F}_2^j(x) \right| \quad \forall j = 1,2,..., p .$$

C'est une extension naturelle du critère KS, seulement l'argument sélectionné pour le seuil de coupure est ici un intervalle et non un réel comme dans le cas classique. On peut donc utiliser toutes les autres étapes (communes à tout type de variable) pour construire l'arbre de décision. Les auteurs ayant construit des arbres de décision sur des variables de type intervalle (Périnel 1996) avec les critères classiques (Gini, likelihood et gain ratio) prennent le centre de l'intervalle comme seuil de coupure.

4 Un exemple illustratif

Nous présentons dans cette section un exemple pour illustrer le mécanisme de construction d'un arbre de décision binaire à l'aide du critère KS sur des variables explicatives de type intervalle. Le tableau (TAB 2) ci-dessous est un extrait de la base de données « développement des pays du monde »[1]. C'est une base de données sur quelques pays du monde en 2000, constituée à partir des indicateurs composites du développement acceptés par la banque mondiale, le fond monétaire international et les Nations Unies. Au début, il y avait une cinquantaine de pays répertoriés sur les cinq continents. Ces pays sont divisés en deux catégories économiques : pays développés (catégorie 0) et pays en développement (catégorie 1). Le croisement de ces deux catégories économiques avec les cinq continents donne dix concepts (ces concepts sont les individus de cet exemple). Par exemple les concepts DEUR et SDEUR signifient respectivement « pays développés » et « pays en développement » dans le continent Européen. Les variables explicatives sont : X_1 = *Population* (en milliers); X_2 = *Taux de croissance de la population* (par an); X_3 = *Superficie totale du pays* (en milliers de kilomètres carrés); X_4 = *Espérance de vie* (en années) et X_5 = *Taux d'analphabétisme des femmes*. La variable à expliquer est Y = *Nom_catégorie*.

[1] Rapport de Stage (DESS Informatique Décisionnelle) de Ravelomanantsoa H., Université Paris Dauphine (2002), disponible à l'URL http://www.ceremade.dauphine.fr/~touati/exemples.htm

Variables / *Individus*	X_1	X_2	X_3	X_4	X_5	Y
DAMQ	[15211;283230]	[0.9;1.8]	[757;9976]	[67.2;78.5]	[0;14.6]	0
SDAMQ	[2576;25662]	[0.8;2.6]	[11;1285]	[64;76]	[4.3;38.7]	1
DEUR	[4469;82017]	[0;0.5]	[41;547]	[77.2;79.3]	[0;0]	0
SDEUR	[1988;145491]	[-0.4;0.2]	[20;17075]	[66.1;78.2]	[0;0]	1
DOCE	[814;19138]	[0.9;1.2]	[18;7682]	[68.4;78.7]	[0;9.2]	0
SDOCE	[159;4809]	[0;2.7]	[3;462]	[55.6;70.5]	[21;52]	1
DAFR	[1161;67884]	[0.8;1.9]	[2;1221]	[56.7;70.7]	[15.4;63.9]	0
SDAFR	[1757;35119]	[0.9;2.9]	[196;945]	[40.6;52.3]	[15.3;72.3]	1
DASI	[4913;1275130]	[0.3;2.9]	[20;9597]	[65.1;80.5]	[1;23.7]	0
SDASI	[16189;1008940]	[1.4;2.6]	[185;3287]	[62.3;71.9]	[4.8;54.6]	1

TAB 2 – *Données initiales.*

Le critère KS requiert un ordre des valeurs des variables explicatives. Pour simplifier, nous présentons ici les résultats obtenus en ordonnant les intervalles par la borne inférieure (TAB 3, le nombre entre parenthèses est la classe a priori de l'intervalle selon la variable *Y*).

X_1	X_2	X_3	X_4	X_5
[159;4809] (1)	[-0.4;0.2] (1)	[2;1221] (0)	[40.6;52.3] (1)	[0;0] (0)
[814;19138] (0)	[0;0.5] (0)	[3;462] (1)	[55.6;70.5] (1)	[0;0] (1)
[1161;67884] (0)	[0;2.7] (1)	[11;1285] (1)	[56.7;70.7] (0)	[0;9.2] (0)
[1757;35119] (1)	[0.3;2.9] (0)	[18;7682] (0)	[62.3;71.9] (1)	[0;14.6] (0)
[1988;145491] (1)	[0.8;1.9] (0)	[20;9597] (0)	[64;76] (1)	[1;23.7] (0)
[2576;25662] (1)	[0.8;2.6] (1)	[20;17075] (1)	[65.1;80.5] (0)	[4.3;38.7] (1)
[4469;82017] (0)	[0.9;1.2] (0)	[41;547] (0)	[66.1;78.2] (1)	[4.8;54.6] (1)
[4913;1275130] (0)	[0.9;1.8] (0)	[185;3287] (1)	[67.2;78.5] (0)	[15.3;72.3] (1)
[15211;283230] (0)	[0.9;2.9] (1)	[196;945] (1)	[68.4;78.7] (0)	[15.4;63.9] (0)
[16189;1008940] (1)	[1.4;2.6] (1)	[757;9976] (0)	[77.2;79.3] (0)	[21;52] (1)

TAB 3 – Ordre des intervalles par la borne inférieure

Nous obtenons la figure (FIG 1) en déroulant l'algorithme KS sur le tableau (TAB 3).

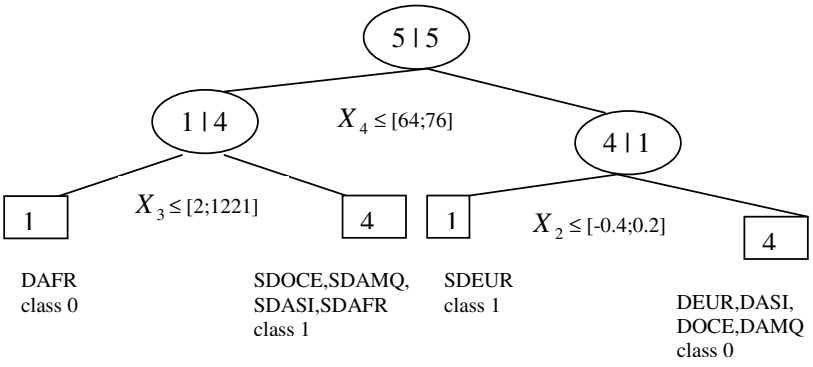


FIG. 1 – *Arbre de décision obtenu en ordonnant les intervalles par la borne inférieure*

5 Evaluation et comparaison

Dans cette section, l'objectif consiste à explorer les différents ordres d'intervalles afin de voir celui qui présente les meilleurs résultats en terme de risque réel (mesuré sur la matrice de confusion de l'ensemble de test). Les fichiers[2] que nous utilisons sont présentés au tableau (TAB 4). Ce sont des fichiers obtenus à partir de bases de données classiques à l'aide du module DB2SO (Data Base two Symbolic Objects) du logiciel libre Sodas[3]. En fait nos individus (ou objets) ne sont pas des individus au sens classique du terme (individus de premier ordre), mais des concepts (individus de second ordre) comme ceux présentés au tableau (TAB 2). La colonne « *Nb_cl* » indique le nombre de classes a priori de la variable à expliquer (variable nominale) et la colonne « *Répartition par classe a priori* » donne la répartition des objets par classe a priori. Par exemple la notation (20 ;10) indique que vingt (20) objets sont de la classe « 1 » et dix (10) de la classe « 2 » pour une variable nominale ayant deux modalités notées « 1 » et « 2 ». La dernière colonne « *Nb_var* » indique le nombre de variables explicatives (toutes de type intervalle).

Le procédé que nous utilisons pour estimer le risque réel est le suivant : pour chaque fichier, nous partageons aléatoirement le nombre d'objets en deux parties égales (en terme d'effectif) à un objet près (nous prenons deux parties car la taille de nos fichiers est relativement modeste) et nous construisons ainsi un arbre pour chaque partie. Le taux d'erreur réel estimé $\hat{\hat{R}}_r$ sera alors la moyenne des deux taux d'erreur estimés obtenus des deux arbres. Pour la construction des deux arbres, nous utilisons deux approches : le cas où les deux arbres sont testés par un même échantillon (section 5.1) et le cas où ils sont testés par des échantillons différents (section 5.2). Le nombre d'objets minimum d'un nœud terminal (fixé à 2 pour les fichiers F_1 à F_9 ; 5 pour les fichiers F_10 à F_12 ; 10 pour les fichiers F_13 à F_15 et 25 pour le fichier F_16) permet d'arrêter le développement de l'arbre. Nous désignons par n la taille totale d'un fichier de données et t celle de son ensemble de test. Dans le cas où le cardinal de l'ensemble de test est assez grand (au-delà de 100 objets), l'intervalle de confiance de l'estimateur $\hat{\hat{R}}_r$ à x% est donné par (Cornuéjols et Miclet 2002) : $[\ \hat{\hat{R}}_r - \varphi(x)\sqrt{\frac{\hat{R}_r}{t}(1-\hat{R}_r)}\ ;\ \hat{\hat{R}}_r + \varphi(x)\sqrt{\frac{\hat{R}_r}{t}(1-\hat{R}_r)}\ ]$ où $\varphi(x)$ prend en particulier les valeurs du tableau TAB 5. Cela signifie que la probabilité que le risque réel soit à l'intérieur de cet intervalle est supérieur à x%. Les intervalles de confiance ne dépendent que de la taille t de l'échantillon de test. Nous présenterons les intervalles de confiance des estimateurs des risques réels des fichiers ayant un ensemble de test de plus de cent objets.

Dans la légende de chaque figure, « Borne sup » signifie que les valeurs des variables explicatives (c'est-à-dire les intervalles) ont été ordonnées par la borne supérieure (idem pour « Borne inf » et « Moyenne »).

[2] Fichiers Sodas disponibles à l'URL http://www.ceremade.dauphine.fr/~touati/exemples.htm
[3] disponible à l'URL http://www.ceremade.dauphine.fr/%7Etouati/sodas-pagegarde.htm

Numéro	*Nom du fichier*	*Taille du fichier*	*Nb_cl*	*Répartition par classe a priori*	*Nb_var*
F_1	Voyage	21	5	(5 ;4 ;3 ;6 ;3)	2
F_2	Wine	23	9	(2 ;2 ;2 ;2 ;6 ;5 ;2 ;1 ;1)	21
F_3	Joueur	29	2	(11 ;18)	7
F_4	Iris de Fisher	30	3	(10 ;10 ;10)	4
F_5	Wave	30	3	(10 ;10 ;10)	21
F_6	Auto	33	4	(10 ;8 ;8 ;7)	8
F_7	Football	45	4	(16 ;21 ;7 ;1)	7
F_8	Accident	48	3	(36 ;9 ;3)	5
F_9	Temperature_1988	60	10	(8 ;4 ;7 ;7 ;8 ;7 ;4 ;5 ;6 ;4)	12
F_10	Shuttle	102	7	(78 ;1 ;1 ;15 ;5 ;1 ;1)	9
F_11	Cholesterol	193	2	(85 ;108)	2
F_12	Age_color	231	2	(126 ;105)	10
F_13	Glucose	690	2	(335 ;355)	2
F_14	Prof-size	720	5	(36 ;441 ;32 ;35 ;176)	1
F_15	Temperature_74-88	900	2	(475 ;425)	12
F_16	Regions_NU	10000	4	(1900 ;2300 ;3620 ;2180)	4

TAB 4 – *Inventaire des fichiers utilisés*

x%	50%	68%	80%	90%	95%	98%	99%
φ(*x*)	0,67	1	1,28	1,64	1,96	2,33	2,58

TAB 5 – *Valeurs de φ(x) en fonction de x*

5.1 Cas où les deux arbres sont testés par un même échantillon

Pour chaque fichier de données, le principe est le suivant :

- on tire aléatoirement t objets. On partage aléatoirement le nombre d'objets restants $(n-t)$ en deux parties égales en terme d'effectif (à un objet près) pour construire deux arbres par la méthode du « hold-out method ». Comme la taille de l'ensemble de test doit être le tiers de celle de l'ensemble d'apprentissage selon le principe de « hold-out method », on voit que t est la division entière de n par sept.
- on construit un arbre pour chaque partie (les deux parties ont le même effectif à un objet près, l'effectif est $\frac{n-t}{2}$) et on teste chaque arbre par les objets retenus à l'ensemble de test.

Les résultats obtenus sur les estimateurs des risques réels pour chaque ordre d'intervalles sont présentés à la figure (FIG 2). Les tableaux (TAB 6 ; TAB 7 et TAB 8) donnent les intervalles de confiance pour les fichiers F_14, F_15 et F_16 (cardinal de l'ensemble de test t au-delà de 100).

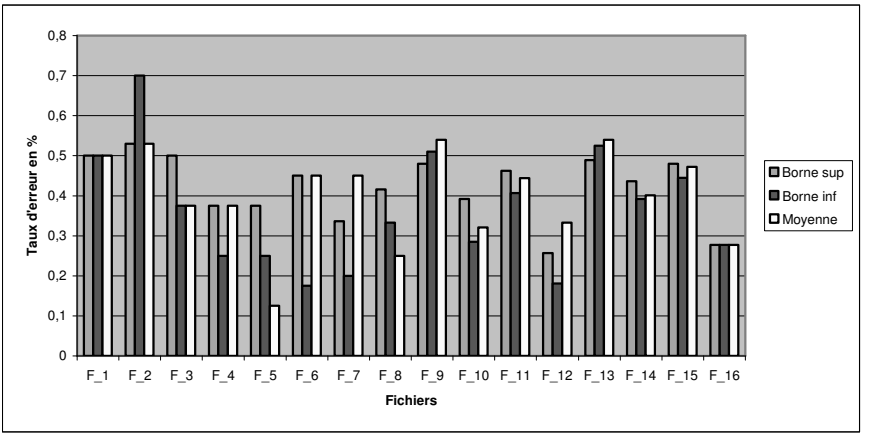


FIG 2 - *Estimation du risque réel pour chaque ordre*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_14	[0.40;0.47]	[0.39;0.48]	[0.37;0.50]	[0.35;0.52]	[0.34;0.53]	[0.31;0.56]
F_15	[0.45;0.50]	[0.43;0.52]	[0.42;0.53]	[0.40;0.55]	[0.39;0.56]	[0.36;0.59]
F_16	[0.27;0.28]	[0.26;0.28]	[0.26;0.29]	[0.25;0.29]	[0.25;0.30]	[0.24;0.30]

TAB 6 – *Intervalles de confiance des risques réels pour l'ordre par la borne supérieure*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_14	[0.35;0.42]	[0.34;0.44]	[0.33;0.45]	[0.31;0.47]	[0.29;0.48]	[0.26;0.51]
F_15	[0.41;0.47]	[0.40;0.48]	[0.38;0.50]	[0.37;0.51]	[0.35;0.53]	[0.33;0.55]
F_16	[0.27;0.28]	[0.26;0.28]	[0.26;0.29]	[0.25;0.29]	[0.25;0.30]	[0.24;0.30]

TAB 7 – *Intervalles de confiance des risques réels pour l'ordre par la borne inférieure*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_14	[0.36;0.43]	[0.35;0.44]	[0.33;0.46]	[0.32;0.48]	[0.30;0.49]	[0.27;0.52]
F_15	[0.44;0.50]	[0.42;0.51]	[0.41;0.52]	[0.39;0.54]	[0.38;0.55]	[0.35;0.58]
F_16	[0.27;0.28]	[0.26;0.28]	[0.26;0.29]	[0.25;0.29]	[0.25;0.30]	[0.24;0.30]

TAB 8 – *Intervalles de confiance des risques réels pour l'ordre par la moyenne*

5.2 Cas où les deux arbres sont testés par des échantillons différents

Le principe est le suivant :

- on partage aléatoirement le fichier de données en deux parties égales en terme d'effectif (à un objet près) ;
- pour chaque partie, on applique la méthode du « hold-out method » pour construire l'arbre (on prend aléatoirement un tiers de l'effectif pour le test et le reste est destiné à l'apprentissage).

Pour les intervalles de confiance des estimateurs des risques réels, seuls les fichiers F_13 à F_16 ont un ensemble de test de cardinal supérieur à 100. Nous obtenons les résultats suivants (FIG 3 ; TAB 9 ; TAB 10 et TAB 11) :

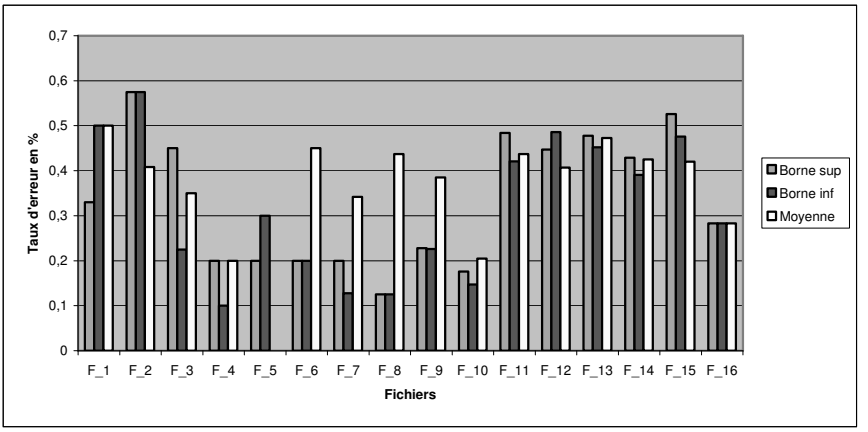


FIG 3 - *Estimation du risque réel pour chaque ordre*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_13	[0.44;0.50]	[0.43;0.52]	[0.41;0.53]	[0.40;0.55]	[0.38;0.56]	[0.35;0.59]
F_14	[0.39;0.45]	[0.38;0.47]	[0.37;0.48]	[0.35;0.50]	[0.34;0.51]	[0.31;0.54]
F_15	[0.49;0.55]	[0.48;0.56]	[0.47;0.57]	[0.45;0.59]	[0.44;0.60]	[0.42;0.63]
F_16	[0.27;0.29]	[0.27;0.30]	[0.26;0.29]	[0.27;0.30]	[0.26;0.30]	[0.25;0.31]

TAB 9 – *Intervalles de confiance des risques réels pour l'ordre par la borne supérieure*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_13	[0.42;0.48]	[0.40;0.49]	[0.39;0.51]	[0.37;0.52]	[0.36;0.54]	[0.33;0.57]
F_14	[0.36;0.42]	[0.34;0.43]	[0.33;0.44]	[0.31;0.46]	[0.30;0.47]	[0.27;0.50]
F_15	[0.44;0.50]	[0.43;0.51]	[0.42;0.52]	[0.40;0.54]	[0.39;0.55]	[0.37;0.58]
F_16	[0.27;0.29]	[0.27;0.30]	[0.26;0.29]	[0.27;0.30]	[0.26;0.30]	[0.25;0.31]

TAB 10 – *Intervalles de confiance des risques réels pour l'ordre par la borne inférieure*

x% / Fichiers	50%	68%	80%	90%	95%	99%
F_13	[0.44;0.50]	[0.42;0.51]	[0.41;0.53]	[0.39;0.54]	[0.38;0.56]	[0.35;0.59]
F_14	[0.39;0.45]	[0.37;0.47]	[0.36;0.48]	[0.35;0.49]	[0.33;0.51]	[0.30;0.54]
F_15	[0.39;0.44]	[0.37;0.46]	[0.36;0.47]	[0.35;0.48]	[0.34;0.49]	[0.31;0.52]
F_16	[0.27;0.29]	[0.27;0.30]	[0.26;0.29]	[0.27;0.30]	[0.26;0.30]	[0.25;0.31]

TAB 11 – *Intervalles de confiance des risques réels pour l'ordre par la moyenne*

5.3 Commentaire des résultats

Comme les échantillons de test et d'apprentissage sont aléatoires et indépendants, la précision de l'estimation ne dépend que du nombre d'objets t de l'ensemble de test et de la valeur du risque réel estimé $\hat{R}_r$ (Cornuéjols et Miclet 2002). Les résultats des sections (5.1 et 5.2) nous montrent des risques réels très similaires pour les trois méthodes d'ordre d'intervalles mais variant fortement d'un fichier à un autre suivant la taille (très élevés pour les fichiers de petite taille notamment). La même tendance est observée sur les deux approches (section 5.1 et section 5.2). Les risques réels diminuent en fonction de l'augmentation du nombre d'objets de l'ensemble de test. Plus la taille de l'échantillon de test est grande, plus l'intervalle de confiance est réduit (exemple de F_16 ayant des intervalles de confiance très réduits par rapport aux intervalles de confiance des autres fichiers).

6 Conclusion et Perspectives

Cette approche exploratoire du critère de Kolmogorov-Smirnov a permis de l'étendre aux données de type intervalle en proposant un seuil de coupure différent (un intervalle) de celui utilisé par les différents auteurs (seuil de coupure classique) ayant construit des arbres de décision sur ce type de données. Nous envisageons de :

- prospecter une autre approche pour la construction d'arbres de décision par ce critère KS : une approche décisionnelle qui consistera à sélectionner le meilleur ordre d'intervalles à chaque nœud (meilleur en terme d'homogénéité des nœuds fils générés) pendant la construction de l'arbre (dans ce cas l'utilisateur n'aura pas à sélectionner un ordre d'intervalles) ;
- comparer ce critère avec quelques critères classiques (Gini, entropie).

Le but est de synthétiser un nouveau tableau de données (à partir des nœuds et des règles de décision) qui peut être étudié à son tour par une autre méthode d'analyse de données.

Références

Asseraf M. (1998), Extension et optimisation pour la segmentation de la distance de Kolmogorov-Smirnov, Thèse de Doctorat, Mathématiques Appliquées, Université Paris IX Dauphine, France.

Bock H. H. et Diday E. (2000), Analysis of symbolic data : Exploratory methods for extracting statistical information from complex data, Springer-Verlag, Berlin-Heidelberg.

Breiman L., Freidman J. H., Ohlsen R. A. et Stone C. J. (1984), Classification and regression trees, The Wadsworth Statistics/Probability Series, Belmont, CA.

Cornuéjols A. et Miclet L. (2002), Appretissage Artificiel : concepts et algorithmes, eyrolles.

Diday E., Gioia F. et Mballo, C. (2003), Codage qualitatif d'une variable intervalle, Journées de Statistique, 35, pp 415-418, Lyon, France.

Friedman J. H. (1977), A recursive partitioning decision rule for non parametric classification; IEEE Transactions on Computers, C-26, Number 4, pp 404-408.

Mballo C. et Diday E. (2004), The criterion of Kolmogorov-Smirnov for binary decision tree: Application to interval valued variables; Proceedings of the Workshop of Symbolic and Spatial Data Analysis: Mining complex Data Structures; Editors: P. Brito et M. Noirhomme-Fraiture; ECML/PKDD 2004; Pisa, Italy, September, 20-24, 2004; pp 79-90.

Mballo C. et Diday E. (2004), Kolmogorov-Smirnov for decision tree on interval and histogram variables; in Studies in classification, Data Analysis and Knowledge organization: Classification, Clustering and Data Mining Applications, editors: D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul; Springer; pp 341-350, Proceedings of the International Federation of the Classification Societies; Chicago, USA, July, 15-18; 2004.

Mballo C., Asseraf M. et Diday E. (2004), Binary decision trees for interval and taxonomic variables ; A Statistical Journal for Graduates Students (incorporating Data & Statistics), Volume 5, Number 1, April 2004, pp 13-28.

Perinel E. (1996), Segmentation et Analyse de données symboliques : Application à des données probabilistes imprécises, Thèse de Doctorat, Spécialité Mathématiques Appliquées, Université Paris Dauphine, France.

Utgoff P. E. et Clouse J. A. (1996), A Kolmogorov-Smirnov metric for decision tree induction, University of Massachusetts, technical report 96-3.

Yapo J. P. A. (2002), Méthodes de segmentation sur un tableau de variables aléatoires, Thèse de Doctorat, Spécialité Mathématiques Appliquées, Université Paris Dauphine, France.

Summary

The binary splitting criterion of Kolmogorov-Smirnov (Friedman 1977) require a total order of the values of the explanatory variables. We can order closed and bounded intervals of real numbers in different ways (Diday et al. 2003). Our contribution in this paper consists of evaluating and comparing decision trees obtained on explanatory variables of interval type with the binary splitting criterion of Kolmogorov-Smirnov. This criterion have been extended to this type of variables (Mballo and al. 2004; Mballo and Diday 2004). We center our attention on the rate of misclassification obtained in the test set. To estimate this parameter, we divide randomly each data base in two equal parts in term of size (to within an object) to construct two decision trees. These two decision trees are first tested by a same test set and then by two different test sets.