Construction interactive d'arbres de décisions avec des variables mixtes

François Poulet

ESIEA – Pôle ECD
38, rue des Docteurs Calmette et Guérin
Parc Universitaire de Laval-Changé
53000 Laval
poulet@esiea-ouest.fr
http://visu.ecd.free.fr

Résumé. Nous présentons une extension d'algorithmes de construction interactive d'arbres de décisions aux cas des variables intervalle et taxonomiques. Les algorithmes présentés sont ainsi capables de traiter indifféremment des variables continues, intervalles et taxonomiques (ou un quelconque mélange de ces trois différents types). Ce type d'approche, centrée utilisateur, est ce qui caractérise la fouille visuelle de données. Ces algorithmes peuvent fonctionner en mode 100% manuel (c'est l'utilisateur seul qui créé l'arbre de décision) ou en mode mixte (coopération entre l'utilisateur et une méthode automatique pour trouver la meilleure coupe pour le noeud courant de l'arbre, ici basée sur des SVM : Séparateurs à Vaste Marge). Après avoir décrit l'adaptation de ces algorithmes aux cas des variables intervalles et taxonomiques, nous présentons les résultats que nous avons obtenus sur différents ensembles de données artificiels.

1. Introduction

L'extraction de Connaissances dans les Données (ECD) peut être définie [Fayyad et al., 1996] comme le processus non trivial de découverte de connaissances valides, nouvelles, potentiellement utilisables et compréhensibles dans les données. Dans la plupart des outils existant, la visualisation n'intervient en général que lors de deux étapes du processus de fouille : dans l'une des toutes premières étapes pour voir les données ou leur distribution, et dans l'une des toutes dernières étapes pour voir le résultat obtenu. Entre ces deux étapes, il y a exécution d'un algorithme automatique. Le rôle de l'utilisateur est donc "simplement" de régler les paramètres de l'algorithme, puis de lancer son exécution et d'attendre les résultats.

De nouvelles méthodes sont récemment apparues [Ankerst et al., 2001], [Poulet, 1999], [Wong, 2001] essayant d'augmenter l'implication de l'utilisateur dans le processus de fouille notamment par le biais d'un rôle plus important de la visualisation [Aggarwal, 2001], [Shneiderman, 2002]. Cette nouvelle approche s'appelle la fouille visuelle de données.

Nous présentons deux algorithmes de classification supervisée et plus précisément de construction interactive d'arbres de décision. Ces algorithmes de classification supervisée utilisent à la fois les capacités humaines en reconnaissance de formes et la puissance de calcul des algorithmes automatiques dans une approche centrée utilisateur.

La section 2 présente brièvement des algorithmes interactifs d'induction d'arbres de décision puis nous nous focalisons sur deux de ces algorithmes que nous allons étendre aux cas des variables intervalle et taxonomiques (CIAD et PBC). Dans la section 3 nous présentons les variables de type intervalle, comment elles peuvent être ordonnées, représentées graphiquement et comment il est possible d'en faire une classification interactive. La section 4 fournit les mêmes renseignements sur les variables de type taxonomique, puis nous présentons les résultats que nous avons obtenus dans la section 5, avant la conclusion et la présentation des travaux futurs.

2. Construction interactive d'arbres de décision

De nouveaux algorithmes de construction interactive d'arbres de décision sont apparus ces dernières années : PBC (Perception-Based Classification) [Ankerst, 2000], DTViz (Decision Tree Visualization) [Han et Cercone, 2001], [Ware et al., 2001] et CIAD (Construction Interactive d'Arbres de Décision) [Poulet, 2001]. Tous ces algorithmes essayent d'impliquer de manière plus importante l'utilisateur dans le processus de construction du modèle. L'utilisateur de ces algorithmes est le spécialiste du domaine des données et non plus un spécialiste de fouille ou d'analyse de données. Ce type d'approche présente au moins les avantages suivants :

- on bénéficie des connaissances du domaine des données tout au long du processus de construction du modèle,
- la confiance et la compréhensibilité du modèle sont accrues (puisque l'utilisateur a participé à sa création),
- on peut utiliser les capacités visuelles humaines en reconnaissance de formes.

Différentes solutions ont été retenues en ce qui concerne le coeur de ces algorithmes : PBC et DTViz utilisent des coupes univariées (la coupe est effectuée suivant la valeur d'un seul attribut, elle est perpendiculaire à l'axe, de la forme : si $Attr_i$ >valeur de coupe). Le point de départ de ces deux algorithmes est une visualisation obtenue par pixelisation des données (chaque individu est transformé en un pixel ou une ligne verticale) dans une barre horizontale. Pour créer une barre, les valeurs de l'attribut sont d'abord triées en ordre croissant et visualisées sous la forme de point ou de ligne, de la couleur de la classe, dans une barre horizontale. Chaque attribut de l'ensemble de données est donc visualisé dans une barre différente (cf. Figure 1). Une fois que cette visualisation est effectuée, l'étape de création de l'arbre de décision peut débuter. L'algorithme de classification consiste alors à effectuer des coupes univariées binaires ou n-aires dans ces barres.

Seuls PBC et CIAD sont munis d'un mécanisme d'aide permettant de trouver à l'aide d'un algorithme automatique la meilleure séparatrice pour le noeud courant de l'arbre de décision. Les autres algorithmes peuvent seulement être utilisés en mode 100% manuel.

CIAD est un algorithme de construction d'arbre de décisions permettant d'effectuer des coupes bi-variées en dessinant la séparatrice dans un ensemble de matrices de projection 2D des données [Chambers et al., 1983]. La première étape de l'algorithme est donc la création d'un ensemble de $(m-1)^2/2$ matrices de projection 2D (m étant le nombre d'attributs de l'ensemble de données). Ces matrices représentent les projections des données en deux dimensions selon toutes les paires possibles d'attributs, la couleur de chaque point correspondant à la classe. C'est une manière relativement efficace de découvrir visuellement les relations entre deux attributs. L'une des matrices 2D peut être sélectionnée et est alors visualisée à une plus grande échelle dans le coin inférieur droit de l'outil de visualisation

(comme sur l'exemple de la figure 2 qui utilise l'ensemble de données Segment de l'UCI Machine Learning Repository [Blake et Merz, 1998]). Cet ensemble de données comporte 2310 individus, 19 attributs et 7 classes. Ensuite l'utilisateur peut commencer la création interactive de l'arbre de décision en traçant les droites de séparation dans la matrice sélectionnée et créant ainsi une coupe binaire (l'espace est partagé en deux sous-espaces) mono ou bi-variée (coupe perpendiculaire à un axe ou 2D-oblique) dans le noeud courant de l'arbre de décision. La stratégie utilisée pour trouver la meilleure coupe est la suivante : on recherche d'abord la coupe permettant d'isoler la plus grande zone pure (avec une seule classe ou couleur). La droite de séparation (perpendiculaire ou non aux axes) entre cette zone pure et le reste des données est alors tracée interactivement avec la souris sur l'écran, les individus correspondant sont considérés comme appartenant à une feuille de l'arbre de décision et sont donc éliminés de l'ensemble des matrices 2D (faisant ainsi potentiellement apparaître de nouvelles zones pures). Si une seule droite n'est pas suffisante pour séparer les données en zones pures, chaque demi-espace correspondant à la première droite est traité alternativement (l'autre demi-espace est alors caché à l'utilisateur).

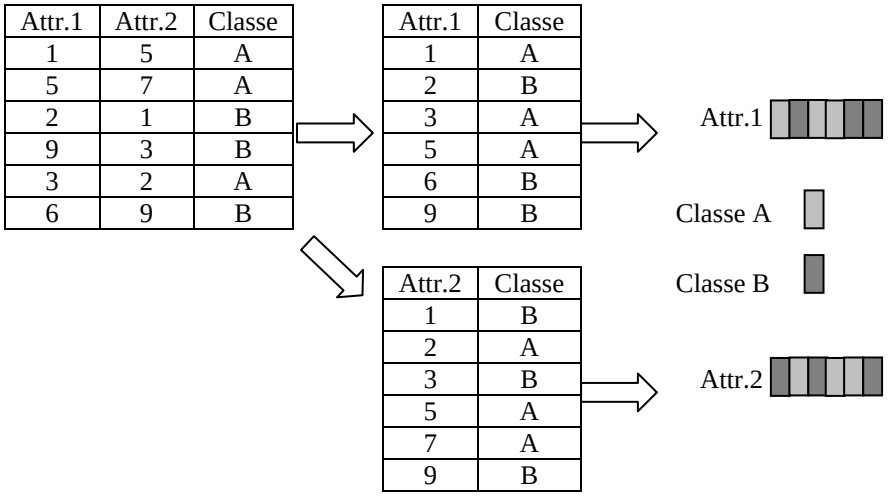

FIG. 1 – *Création des barres avec PBC*

A chaque étape de la classification des informations complémentaires peuvent être fournies à l'utilisateur comme le nombre d'individus du noeud, la pureté de la coupe ou le taux global de bonne classification (dans l'état courant de l'arbre de décisions). D'autres types d'interactions sont possibles pour aider l'utilisateur, il est possible de cacher ou sélectionner une classe, un individu ou un groupe d'individus.

Plusieurs mécanismes d'aide sont aussi disponibles. Une première possibilité est d'optimiser la position de la droite de séparation tracée (elle devient la meilleure droite de séparation, c'est à dire la plus éloignée des éléments de chaque côté de la droite, comme sur l'exemple de la figure 3). Une autre possibilité est de trouver de manière automatique la meilleure droite de séparation dans le noeud courant ou pour tous les nœuds successifs de l'arbre de décision. La droite est alors affichée à l'écran pour le noeud courant ou pour l'ensemble des noeuds successifs dans le cas d'une construction automatique de l'arbre. Ces

mécanismes d'aide à l'utilisateur sont basés sur des algorithmes de SVM (Séparateurs à Vaste Marge ou Support Vector Machine) modifiés pour trouver la meilleure séparatrice en 2D (au lieu du meilleur hyperplan en dimension m, pour un ensemble de données avec m attributs). Les algorithmes de SVM ne pouvant traiter que le cas de la classification de deux classes, on utilise l'approche "un contre le reste" dans les cas où le nombre de classes est supérieur à deux.

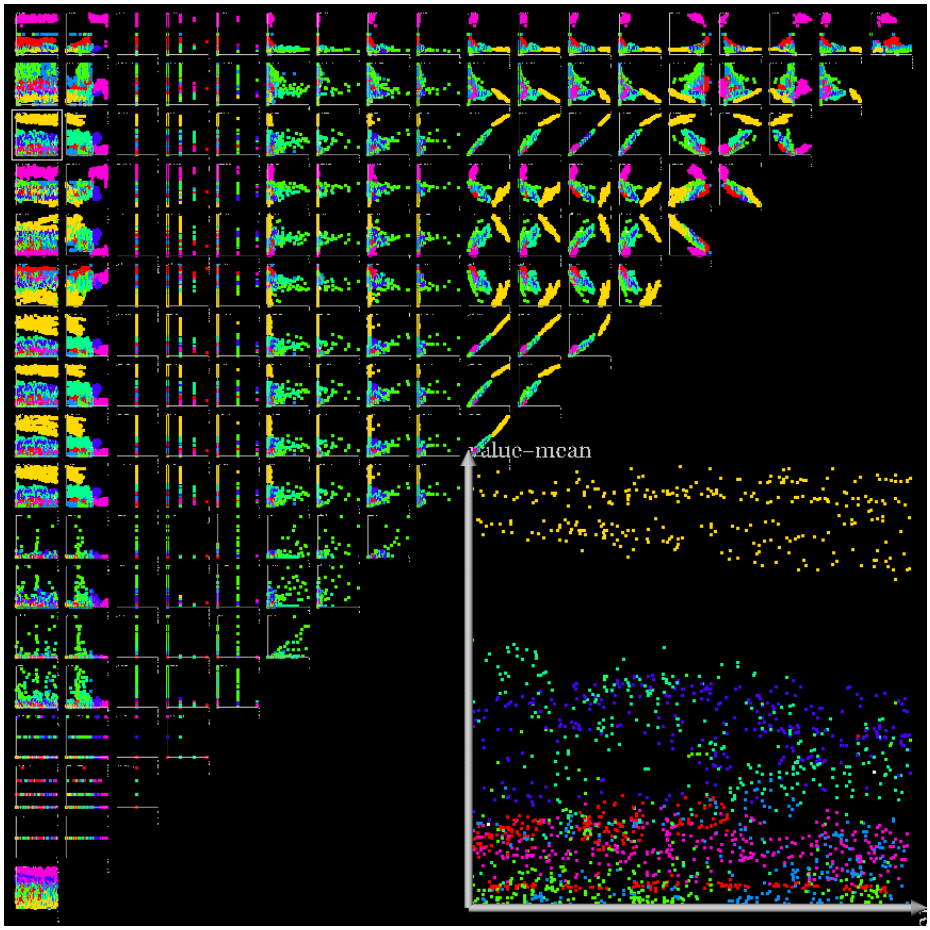

FIG. 2 – *Visualisation de l'ensemble de données Segment avec CIAD*

3. Données intervalles

Les algorithmes d'induction d'arbres de décision traitent en général des variables qualitatives ou quantitatives. Ici nous allons nous intéresser au cas des variables de type intervalle. Nous nous intéressons plus particulièrement au cas des intervalles finis. Par rapport aux méthodes automatiques existantes, notre approche est semblable à celles de [Périnel, 1996] et [Asseraf et Mballo, 2003]. En effet, dans le cas de l'extension de

l'algorithme CIAD, la coupe est effectuée sur une valeur précise de la variable (de la forme *si attribut$_i$ > valeur_seuil*) comme dans [Périnel, 1996] alors que dans le cas de PBC, on se rapprocherait plus de [Asseraf et Mballo, 2003] puisque l'on ordonne les données et que l'on coupe par rapport à un individu limite de type intervalle (de la forme si attribut$_i$ > attribut-seuil).

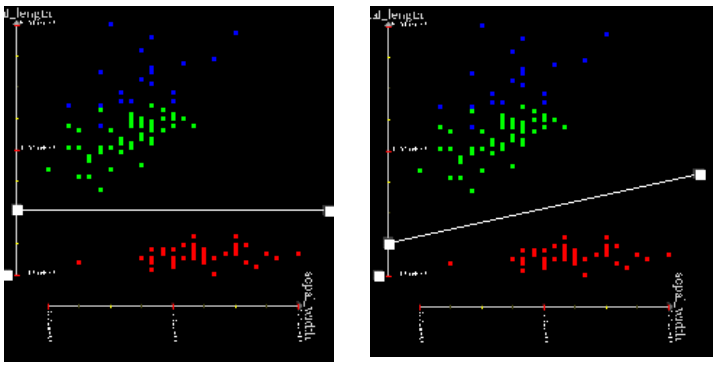

FIG. 3 – *A gauche la droite tracée par l'utilisateur, à droite la meilleure séparatrice*

3.1 Ordre sur des variables intervalle

Pour pouvoir utiliser ce type de variables avec PBC, il est nécessaire de définir un ordre sur ce type de données. Il y a essentiellement trois types d'ordre sur les intervalles [Mballo et al., 2003] : suivant les valeurs minimales, maximales ou moyennes. Si l'on considère deux intervalles I1 = [min1, max1] (moyenne m1) et I2 = [min2, max2] (moyenne m2) alors :

- si les intervalles sont ordonnés suivant les valeurs minimales :

si min1=min2 alors I1<I2 <=> max1<max2 ; si min1<>min2 alors I1<I2 <=> min1<min2

- si les intervalles sont ordonnés suivant les valeurs maximales :

si max1=max2 alors I1<I2<=>min1<min2 ; si max1<>max2 alors I1<I2<=> max1<max2

- si les intervalles sont ordonnés suivant les valeurs moyennes : I1<I2 <=> m1 < m2

Les trois relations définies sont des relations d'ordre total. Les auteurs laissent à l'utilisateur le libre choix d'utiliser l'ordre le plus convenable pour ses données. Nous faisons le même choix, pour créer les barres dans la première étape de l'exécution de l'algorithme PBC.

3.2 Représentation graphique de variables intervalle

Pour pouvoir manipuler des variables de type intervalle avec l'algorithme CIAD nous devons trouver une représentation graphique qui sera utilisée dans les matrices 2D pour le cas de deux variables intervalle et pour le cas d'une variable intervalle x une variable continue. Dans ce dernier cas, un segment est une solution évidente. Pour représenter le croisement de deux variables de type intervalle dans une matrice 2D nous avons besoin d'une primitive graphique sur laquelle nous pouvons "projeter" deux valeurs différentes sur chacun des axes (la couleur jouant toujours le rôle de la classe). Nous avons retenu la primitive la plus simple : une croix.

3.3 Classification de données intervalle avec PBC

Nous avons expliqué comment une relation d'ordre peut être définie sur des données de type intervalle dans le paragraphe 3.1. Cette méthode est utilisée dans la première étape de l'exécution de l'algorithme PBC pour créer les barres horizontales. Une fois que cela est fait pour chaque attribut, le déroulement de l'algorithme de création d'arbre de décision est absolument identique. Etant donné que le traitement est similaire dans les deux cas, il est aussi possible de mélanger les données des deux types : continues (cas particulier d'intervalle ayant le minimum égal au maximum) et intervalle.

3.4 Classification de données intervalle avec CIAD

Comme nous l'avons expliqué dans le paragraphe 2, la première étape de l'algorithme CIAD est d'afficher les données sous la forme d'un ensemble de projections 2D selon toutes les paires possibles d'attributs (la couleur correspondant à la classe). Cette première étape sera exactement la même dans le cas des variables intervalle mais en utilisant des segments ou des croix dans les cas de croisement d'une variable intervalle avec une variable continue ou de deux variables intervalle. Une fois cet affichage effectué, le déroulement de l'algorithme de création d'arbre de décision est absolument identique aux cas des variables continues.

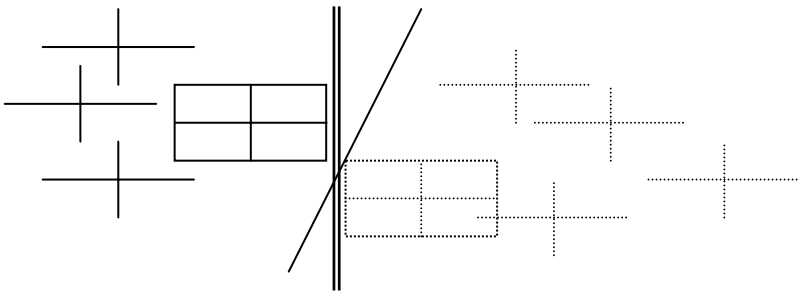

FIG. 4 – *Les meilleures séparatrices selon les centres (trait simple) et les min et max (trait double) des intervalles entre les classes continu et pointillés*

Pour le moment, le mécanisme d'aide est aussi identique. Il est toujours basé sur un algorithme dérivé des SVM (en se basant sur les centres des croix ou segments). De ce fait, il n'est pas optimal dans le cas des variables de type intervalle, puisque seul le milieu des deux intervalles est pris en compte. Pour le rendre optimal il faudrait prendre en compte les valeurs minimum et maximum de chaque intervalle, c'est-à-dire les quatre coins du rectangle correspondant (nous sommes en train de modifier le programme pour pouvoir bénéficier efficacement de cette optimisation). Sur l'exemple de la figure 4, on voit la différence que cela implique avec les meilleures séparatrices prenant en compte les milieux ou les extrémités des intervalles.

4. Données taxonomiques

Une variable taxonomique peut être définie comme une projection des données originales sur un ensemble de valeurs ordonnées. C'est l'équivalent d'une variable hiérarchique. Par

exemple une variable indiquant le lieu de résidence peut aussi bien utiliser un nom de ville, de région ou de pays. La variable taxonomique décrivant le lieu de résidence pourra utiliser indifféremment n'importe quel niveau de description (ville, région ou pays). Donc dans l'ensemble de données ou pourra trouver aussi bien des descriptions données par un nom de ville, de région ou de pays. A partir de cette description hiérarchique, on peut se ramener à un ensemble de valeurs ordonnées en utilisant un parcours d'arbre (en profondeur ou en largeur) pour numéroter les valeurs.

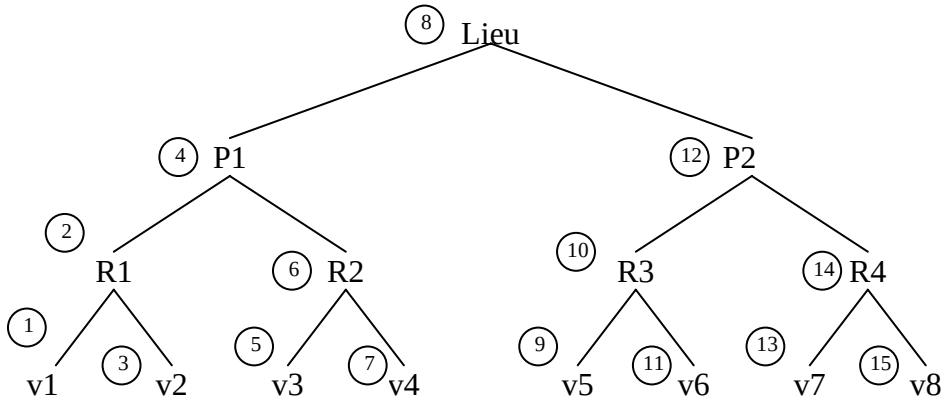

FIG. 5 – *Description hiérarchique du lieu d'habitation*

Lieu	Classe
v1	1
v2	2
v3	1
v3	1
R1	2
P1	2
v5	1
R3	2
v7	1

Lieu (profond.)	Classe
v1	1
R1	2
v2	2
P1	2
v3	1
v3	1
v5	1
R3	2
v7	1

Lieu (largeur)	Classe
P1	2
R1	2
R3	2
v1	1
v2	2
v3	1
v3	1
v5	1
v7	1

TAB 1 – *Un exemple d'ensemble de données taxonomiques, non ordonné à gauche, ordonné par un parcours en profondeur au centre et par un parcours en largeur à droite*

Voyons ce que cela peut donner sur un exemple très simple de lieu de résidence. Le lieu de résidence est défini par l'arbre binaire de la figure 5. Les feuilles de l'arbre correspondent à des noms de villes (Vi), les noeuds du niveau supérieur sont des noms de régions (Ri) et le niveau encore au-dessus correspond à des noms de pays (Pi). Dans l'ensemble de données, la description du lieu d'habitation peut utiliser n'importe lequel de ces niveaux (sauf la valeur générique de la racine). Un exemple d'un tel ensemble de données est représenté dans le tableau 1, avec une classe a priori prenant les valeurs 1 ou 2. Les deux colonnes à gauche du tableau correspondent aux valeurs originales de l'ensemble de données, les deux colonnes centrales correspondent aux mêmes données ordonnées suivant un parcours en profondeur d'abord de l'arbre (les nœuds de l'arbre seront parcourus dans l'ordre indiqué par les cercles

sur la figure 5 : v1, R1, v2, P1, v3, R2, v4, etc.) et les deux colonnes de droite correspondent aux données suivant un parcours en largeur d'abord de l'arbre (P1, P2, R1, R2, R3, R4, v1, v2, etc.).

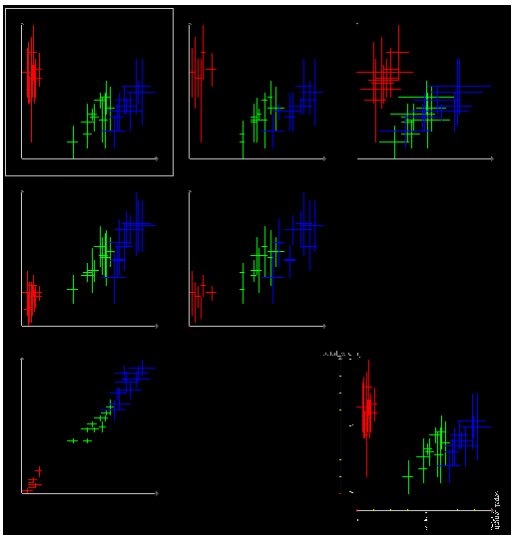

FIG. 6 – *Version intervalle de l'ensemble de données Iris*

4.1 Représentation graphique de variables taxonomiques

Une fois les données ordonnées, quel que soit le parcours d'arbre choisi, une variable taxonomique peut être considérée comme une variable intervalle. Quand la variable n'est pas une feuille de l'arbre (par exemple P1 ou R3 dans la figure 5), elle est graphiquement équivalente à l'ensemble des noeuds du sous-arbre correspondant (P1=[v1..v4] et R3=[v5..v6]). Dans le cas d'une représentation 2D nous utiliserons exactement les mêmes primitives graphiques que dans le cas des variables intervalle : une croix pour le croisement de données (taxonomique x taxonomique) ou (taxonomique x intervalle) et un segment pour le cas (taxonomique x continue) ou (intervalle x continue).

4.2 Classification interactive avec des variables taxonomiques

Là encore le mode de fonctionnement de l'algorithme PBC sera identique aux cas des variables continues ou intervalles (lorsqu'il est utilisé en mode 100% manuel). L'ordre que nous venons de définir est utilisé dans la première étape de tri suivant les valeurs des attributs.

En ce qui concerne CIAD, il n'y a pas d'étape de tri des attributs. Nous avons vu comment représenter graphiquement les variables taxonomiques. Une fois ces individus affichés, le mode de fonctionnement est exactement le même que dans le cas des variables de type intervalle (avec le même mécanisme d'aide aujourd'hui toujours basé sur les centres des croix).

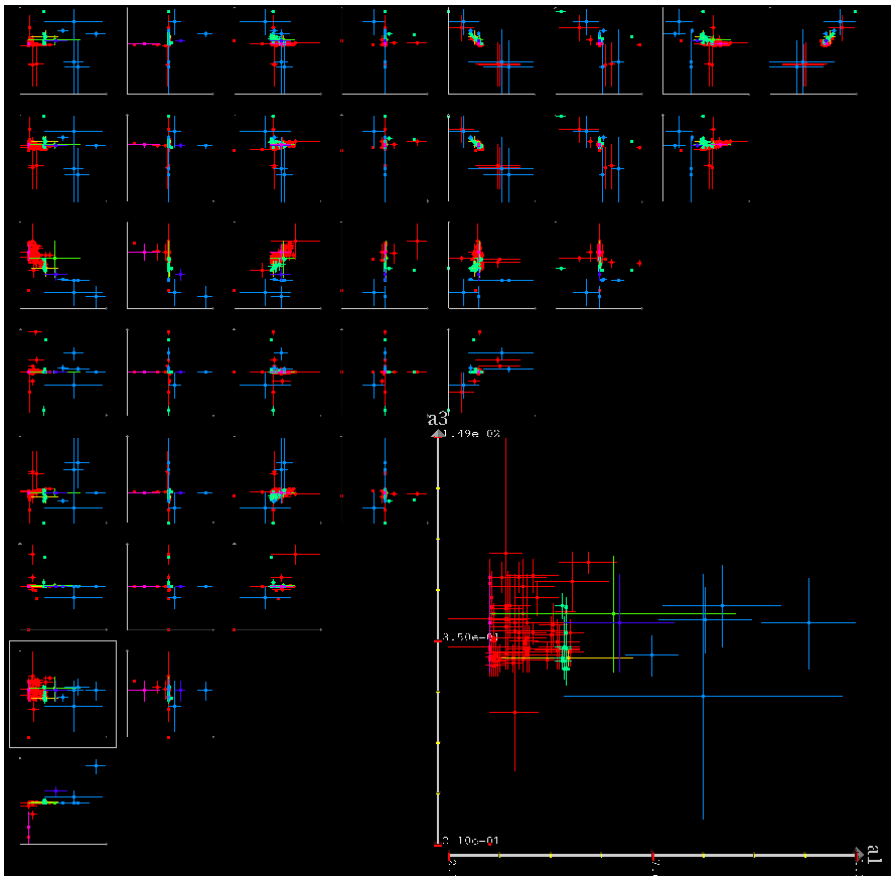

FIG. 7 – *Version intervalle de l'ensemble de données Shuttle*

5. Résultats - Discussion

Tout d'abord, à notre connaissance, à l'heure actuelle il n'existe pas de résultats concernant le taux de bonne classification des algorithmes automatiques de création d'arbres de décision pour des variables de type intervalle ou taxonomique même dans les articles les plus récents sur le sujet [Mballo et Diday, 2004].

Nous avons créé plusieurs ensembles de données de type intervalle et taxonomiques essentiellement à partir des ensembles de données (continues) de l'UCI Machine-Learning Repository (comme sur les exemples des figures 6 et 7 pour les ensembles de données Iris et Shuttle) et utilisé l'ensemble de données intervalle Wave du logiciel SODAS. Quelques mots tout d'abord sur la création des ensembles de données intervalle à partir des données continues. Nous avons utilisé un algorithme de clustering pour créer par exemple 100 clusters pour Shuttle (43500 individus, 9 variables continues et 7 classes). Les valeurs minimum et maximum sur chaque variable (continue) des individus d'un même cluster servent ensuite de bornes pour créer la variable intervalle.

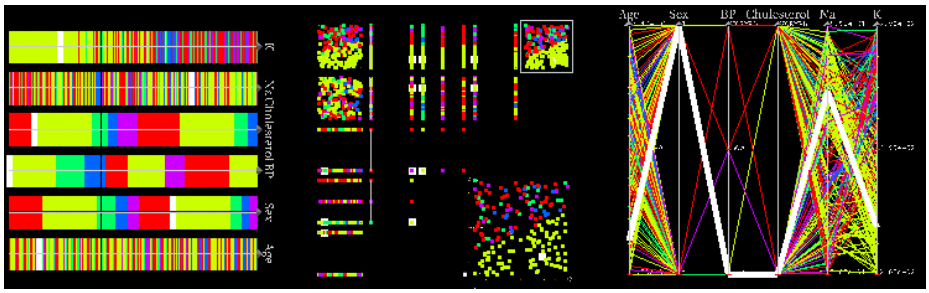

FIG. 9 – *Plusieurs représentations liées des mêmes données : sur la gauche les BarCharts, au centre les matrices de scatter-plot et sur la droite les coordonnées parallèles*

Les résultats sont conformes à ce que nous attendions. En ce qui concerne l'ensemble de données Iris, le taux de bonne classification est le même que pour sa version continue (100%), mais cet ensemble de données a plus été utilisé pour des besoins d'illustration de notre méthode que pour sa validation (l'arbre obtenu est représenté Figure 10). Nous nous sommes aussi intéressés à l'ensemble de données Shuttle. A partir de cet ensemble de données nous en avons créé une version intervalle avec 100 individus. Cet exemple est intéressant car il montre que l'on peut utiliser un algorithme de classification non supervisée en pré-traitement pour réduire le nombre d'individus en passant de variables continues à des variables de type intervalles représentant les valeurs minimums et maximums de chaque cluster. Il est ainsi potentiellement possible de traiter des fichiers de données de très grandes tailles (dans le cas de Shuttle, on est passé de 43500 à 100 individus). La classification s'effectue alors en considérant des groupes d'individus et non plus sur de simples individus. Dans ce cadre d'utilisation, la représentation graphique des groupes choisie n'est sans doute pas optimum, il faudrait prendre en compte le nombre d'individus présents dans le groupe (le coût de bonne/mauvaise classification varie d'un groupe d'individus à l'autre). La solution envisagée est d'utiliser l'épaisseur des traits pour représenter le nombre d'individus.

```
petal_length <= -5.6 * sepal_length + 19.2 : Iris_setosa (10)
petal_length > -5.6 * sepal_length + 19.2 :
|    petal_width <= sepal_length + -2.2 : Iris_virginica (10)
|    petal_width > sepal_length + -2.2 : Iris_versicolor (10)
Taux de bonne classification : 100.00 %
Nombre de noeuds : 5 (dont 3 feuilles)
```

FIG. 10 – *Arbre de décision sur la version intervalle d'Iris*

Nous n'avons pas pu effectuer de comparaison par rapport à des algorithmes automatiques, les résultats concernant les taux de bonne classification n'étant pas encore disponibles.

Les algorithmes que nous avons présentés peuvent traiter simultanément des variables de types quantitatives, qualitatives, intervalles ou taxonomiques, ceci est dû au fait que le type de la séparatrice est indépendant du type des données comme pour l'approche de [Périnel, 1996] ce qui n'est pas le cas dans les algorithmes automatiques développés par [Asseraf et al,

2004]. Par contre l'extension des algorithmes interactifs à d'autres types de variables symboliques (comme des histogrammes) semble moins simple à mettre en œuvre.

L'ensemble des algorithmes interactifs ont été inclus dans l'environnement graphique de fouille de données que nous continuons à développer [Poulet, 2002]. Tout le développement se fait en C/C++ et ne fait appel qu'à des bibliothèques "Open-Source" (Open-GL, Open-Inventor et Open-Motif) afin d'assurer une bonne portabilité.

6. Conclusion et travaux futurs

Nous avons présenté dans cet article deux algorithmes de création interactive d'arbres de décision pouvant traiter des variables standards (numériques ou qualitatives) mais aussi des variables de type taxonomique ou intervalles ou un mélange de n'importe lesquelles des catégories précédemment citées.

Ces algorithmes peuvent fonctionner indifféremment en mode 100% manuel (l'utilisateur doit tracer interactivement sur l'écran chaque coupe de l'arbre de décision), en mode mixte, une aide est disponible pour l'utilisateur pour trouver la meilleure coupe dans le nœud courant de l'arbre (basée sur des SVM dans le cas de CIAD) ou en mode automatique.

A l'heure actuelle les résultats (concernant le taux de bonne classification et la taille des arbres) des algorithmes automatiques traitant de tels types de variables ne sont pas encore disponibles. Nous n'avons donc pas été en mesure de comparer notre approche par rapport aux méthodes automatiques pour les variables de types intervalle et taxonomie. Cependant ces comparaisons ont été effectuées par rapport au traitement de données "standard", les taux de bonne classification sont comparables avec souvent une taille d'arbre un peu plus faible pour les algorithmes interactifs.

Mais le taux de bonne classification n'est qu'un critère de comparaison. L'autre intérêt de notre approche est que l'utilisateur est beaucoup plus impliqué dans la construction du modèle. Sa compréhension du modèle est ainsi accrue de même que sa confiance dans le modèle en question.

L'inconvénient de notre approche est que l'on peut pas traiter facilement des ensembles de données ayant un trop grand nombre de dimensions (typiquement plus d'une cinquantaine). Nous envisageons de faire coopérer des méthodes automatiques (pour réduire le nombre de dimensions à traiter) avec notre méthode de construction interactive d'arbres de décision pour pouvoir traiter des ensembles de données ayant un nombre plus important de dimensions.

Références

[Aggarwal, 2001] C.Aggarwal, Towards Effective and Interpretable Data Mining by Visual Interaction, in *SIGKDD Explorations* 3(2), 11-22, accessed from www.acm.org/sigkdd/explorations/.

[Ankerst, 2000] M.Ankerst, *Visual Data Mining*, PhD Thesis, Ludwig Maximilians University of Munich, 2000.

[Ankerst et al., 2001] M.Ankerst, M.Ester, H-P.Kriegel, Toward an Effective Cooperation of the Computer and the User for Classification in proc. of KDD'2001, 179-188, 2001.

[Asseraf et Mballo, 2003] M.Asseraf, C.Mballo, Arbre de décision pour des variables de type intervalle, XXXV^{e} Journées de Statistique, Lyon, France, 2003, 117-120.

[Asseraf et al, 2004] M.Asseraf, C.Mballo, E.Diday, Binary decision trees for interval and taxonomical variables, to appear in *Student*, Presses Académiques de Neuchâtel.

[Blake et Merz, 1998] C.Blake, C.Merz, *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, 1998.

[Bock et Diday, 2000] H.H.Bock, E.Diday, Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data, Springer-Verlag, Berlin-Heidelberg, 2000.

[Carr et al., 1987] D.Carr, R.Littlefield, W.Nicholson and J.Littlefield, Scatterplot matrix techniques for large N, *Journal of the American Statistical Association* 82(398), 424-436, 1987.

[Fayyad et al., 1996] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Eds, Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996.

[Han et Cercone, 2001] J.Han, N.Cercone, Interactive Construction of Decision Trees in proc. of PAKDD'2001, *LNAI 2035*, 575-580, 2001.

[Mballo et al., 2003] C.Mballo, F.Gioia, E.Diday, Codage qualitatif de variables intervalle, *35e Journées de Statistique*, Lyon, France, 2003.

[Mballo et Diday, 2004] C.Mballo, E.Diday, Kolmogorov-Smirnov for Decision Tree on Interval and Histogram's Variables, to appear in proc. of IFCS'2004, Chicago, 2004.

[Périnel, 1996] E.Périnel, Segmentation et analyse de données symboliques Application à des données probabilistes imprécises, Thèse de l'Université Paris-IX Dauphine, Sept.1996.

[Poulet, 1999] F. Poulet, Visualization in data mining and knowledge discovery, in proc. of *HCP'99, 10th Mini Euro Conference "Human Centered Processes"*, 183-192, 1999.

[Poulet, 2001] F.Poulet, CIAD : Construction Interactive d'Arbres de Décision, SFC'2001, *8e Congrès de la Société Francophone de Classification*, Pointe-à-Pitre, 275-282, 2001.

[Poulet, 2002] F.Poulet, FullView: A Visual Data Mining Environement, IJIG: International Journal of Image and Graphics, vol.2(1), 127-144, 2002.

[Shneiderman, 2002] B.Schneiderman, Inventing Discovery Tools: Combining Information Visualization with Data Mining, in *Information Visualization* 1(1), 5-12, 2002.

[Ware et al., 2001] M.Ware, E.Franck, G.Holmes, M.Hall, I.Witten, Interactive Machine Learning: Letting Users Build Classifiers, in *International Journal of Human-Computer Studies* (55), 281-292, 2001.

[Wong, 1999] P.Wong, Visual Data Mining, in *IEEE Computer Graphics and Applications*, 19(5), 20-21, 1999.

Summary

We present extensions of existing interactive decision tree construction algorithms in order to deal with interval and taxonomical data. The presented algorithms are able to deal with qualitative, quantitative, interval and taxonomical data (or a mix of any of these). This new kind of user-centered approach is a particularity of the Visual Data Mining approach. The algorithms can be used in a manual, semi-automatic or automatic way. In the last two ways, an automatic algorithm (derived from Support Vector Machine) is used to find the best separating line in the current tree node. After the description of the algorithm adaptation to interval-valued and taxonomical data, we present some of the results we have obtained on various artificial data-sets.