Expliquer à l'utilisateur le résultat d'un arbre de décision

Explaining the result of a decision tree to the end-user

I. Alvarez[1] D. Lacondemine[2] F. Stermann[1]

[1]Cemagref-LISC
[1]LIP6
[2]Université de Reims

LIP6 – Université de Paris VI 8 rue du capitaine Scott 75015 Paris
Cemagref –LISC 24 avenue des Landais - BP 50085 63172 Aubière CEDEX 1

Résumé

Les arbres de décision sont des outils de discrimination qui peuvent être utilisés pour prédire la classe de nouveaux cas. Pour apprécier ce résultat, l'utilisateur ne dispose en général que de la trace du classement et d'une estimation du taux d'erreur. Malheureusement la trace ne contient pas les informations nécessaires à une bonne compréhension de la situation. Nous proposons d'apporter à l'utilisateur une information sur la robustesse du résultat et une représentation de la situation dans le cas de données numériques. Pour cela nous recherchons les tests de l'arbre de décision qui sont les plus sensibles à une modification des données d'entrée. Cette analyse de sensibilité est effectuée par une étude géométrique de la surface de décision (la frontière de l'image inverse des différentes classes). La métrique sous-jacente sert de support pour décrire la situation à l'utilisateur. Celui-ci peut suggérer des changements de métriques et les tests les plus sensibles sont recalculés.

Mots Clef

Explication, arbre de décision

Abstract

Decision trees can be used as decision support systems, in order to classify input data and to provide the output class as a result. In order to appraise this result, the end-user can rely mainly on the trace of the classification, or on some estimation of the error-rate. But the trace doesn't contain all the information that is necessary to understand the case at hand. We propose a new method to qualify the result given by the decision tree for a new case, when the input data are continuous. We first identify in the tree, (and not in the trace), the tests that are the most relevant to explain the output class, by the way of sensitivity analysis. This is performed by a geometric study of the decision surface (that is the frontier of the inverse image of the different classes). The underlying metric is then used as a medium to describe the global situation to the end-user. He can suggest a more relevant metric and the most sensitive tests are computed for the latter.

Keywords

Explanation, decision tree

1 Introduction

Les arbres de décisions (AD) sont des outils populaires pour l'aide à la décision et la fouille de données. Ils sont utilisés comme systèmes d'aide à la décision (SIAD) dans de nombreux domaines, en particulier en médecine, pour le scoring bancaire, et dans de nombreux problèmes de discrimination. (Voir une liste d'applications réelles dans [13]). De nombreux logiciels intègrent les algorithmes dérivés de CART [3] ou de ID3 et de ses successeurs comme C4.5 [15]. Ces logiciels (voir à titre d'exemple les logiciels des listes de http://www.kdnuggets.com ou http://www.mlnet.org/), permettent de générer, élaguer et manipuler les arbres de décision. Mais l'utilisateur final est aussi intéressé par des informations sur la fiabilité et la pertinence du résultat. En ce qui concerne la fiabilité, c'est principalement par une estimation du taux d'erreur qu'il est possible d'apporter des indications à l'utilisateur. Or on constate que la plupart des travaux sur l'erreur de prédiction concernent l'étape de construction de l'arbre et non son utilisation effective (voir [3] ; [15] ; [7] ; [12] ; [11]). Cependant ces méthodes d'estimation de l'erreur, basés sur les méthodes de rééchantillonage (voir un exposé complet dans [6]), pourraient sans difficultés être adaptées au besoin de l'utilisateur final, et lui apporter des éléments d'information appropriés à chaque cas.

Permettre à l'utilisateur d'évaluer la pertinence du résultat est plus difficile. En effet, c'est une tâche qui dépend à la fois de l'utilisateur et du système. Les travaux sur l'intelligibilité de l'arbre de décision (principalement les méthodes d'élagage, voir [7]) participent de cet objectif. C'était aussi une des premières préoccupations pour la construction des arbres flous (comme dans [20]). Cependant ces méthodes s'intéressent à l'intelligibilité de l'arbre en tant que modèle. La pertinence d'un résultat

particulier n'est accessible que par l'intermédiaire de la trace du classement, c'est-à-dire le chemin parcouru dans l'arbre.

Nous verrons dans la section suivante que la trace du classement ne contient pas les informations nécessaires à une bonne compréhension de la situation. Elle présente les mêmes limitations que la trace du raisonnement dans les systèmes experts. La trace du classement présente l'ensemble (ou un sous-ensemble) des tests qui ont permis le classement d'un point. Mais la modification de certains de ces tests peut être sans incidence sur le résultat. A l'inverse, la modification d'attributs qui n'apparaissent pas dans la trace peut être déterminante pour modifier le résultat. La pertinence des éléments de la trace pour expliquer le résultat à l'utilisateur est donc difficile à établir. En effet, l'arbre de décision réalise une partition de l'espace des données en autant de régions (connexes ou non) qu'il y a de classes. La trace du classement n'exploite pas les informations contenues dans cette partition.

Nous proposons dans la section 3 de rechercher les tests de l'arbre de décision qui sont les plus sensibles à une modification des données initiales. C'est l'ensemble de ces tests qui est pertinent pour expliquer le classement d'un point, et non pas la trace du classement. Notre méthode est basée sur l'étude géométrique de la surface de décision (la frontière entre les différentes régions de la partition de l'espace de départ). Dans le cas des données numériques et des arbres à tests parallèles aux axes, un algorithme très simple permet de projeter les données initiales sur la surface de décision. On déduit immédiatement du point de projection les tests de l'arbre de décision qui sont pertinents pour les données initiales.

Cette méthode repose sur la définition d'une métrique, dont le rôle est discuté dans la section 4. L'impact d'un changement de métrique peut être parfaitement contrôlé pour une large catégorie de transformations. Rendue explicite, la métrique peut servir de support de communication entre le système et l'utilisateur. Ce dernier, au vu des éléments qui lui sont présentés, peut suggérer des changements de métriques que le système exploite pour recalculer les tests pertinents. Un exemple tiré de l'entrepôt de données du UCI est présenté dans cette section.

En conclusion, nous examinons l'extension de la méthode géométrique à d'autres types d'arbres et à d'autres techniques connexes. Nous examinons aussi la possibilité de décrire plus globalement la situation en définissant un prototype de classe au voisinage du point à expliquer.

2 Les limites de la trace comme support d'explication

La plupart des logiciels intégrant des algorithmes dérivés de C4.5 ou de CART permettent de visualiser la trace du classement d'un nouveau cas. La trace indique comment le cas a été classé. Cependant la trace n'est pas toujours lisible, d'autant moins que la profondeur de l'arbre est importante. Elle présente par ailleurs les mêmes limites comme support d'explication que la trace du raisonnement dans les systèmes experts à base de règles. En effet, on passe facilement d'un arbre de décision à une suite ordonnée de règles, en parcourant successivement tous les chemins de la racine aux différentes feuilles. Or les nombreux travaux sur la trace du raisonnement dans les années 1980 (voir [17], [16], [9], [10]) se sont finalement orientés vers une reconstruction de l'explication ([5], [14], [21]). En effet, la trace ne rend pas compte de l'ensemble des informations contenues dans la partition que l'arbre réalise dans l'espace des données. Des informations essentielles à la compréhension de la situation ou à un élagage intelligent peuvent donc ne pas être présentes dans la trace. C'est ce qu'illustrent les figures 1 à 3 avec des exemples très simples. La figure 1 représente un arbre de décision binaires à deux classes, dont les tests sont numériques et linéaires.

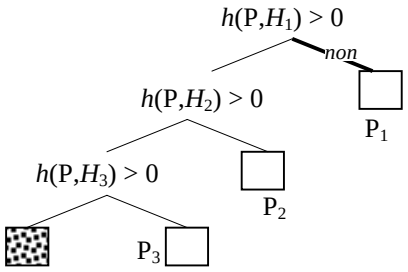

Figure 1. Un exemple d'arbre de décision à tests linéaires (ADL). Un test consiste à calculer la distance algébrique *h* du point à classer à un hyperplan *H*. Le test est vérifié suivant le signe de *h*(P,*H*). Une feuille est l'intersection des demi-espaces définis par les tests. On voit la trace du classement des points P_1 à P_3.

Les figures 2 et 3 représentent des exemples de partition correspondant à l'arbre de la figure 1 dans l'espace des données d'entrée. On examine pour plusieurs points **P** la trace du classement et la classe renvoyée par l'arbre dans un voisinage du point P, une boule ouverte $B(\mathbf{P}, r)$ de centre **P** et de rayon r que l'on fait augmenter progressivement (pour de faibles valeur de r, tous les points de B sont classés comme P).

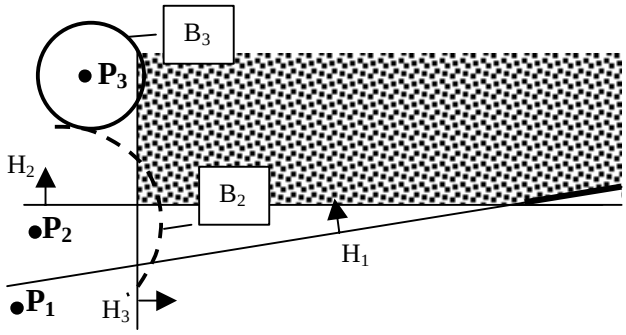

Figure 2. Une partition associée à l'arbre de la figure 1. La trace du classement des points P_1, P_2, P_3 décrit mal la situation (cf. texte). En particulier, la boule B_3 de centre P_3 contient les points proches de P_3. Seul l'hyperplan H_3 sépare les deux classes dans B_3, et non H_1 et H_2 pourtant présents dans la trace du classement de P_3.

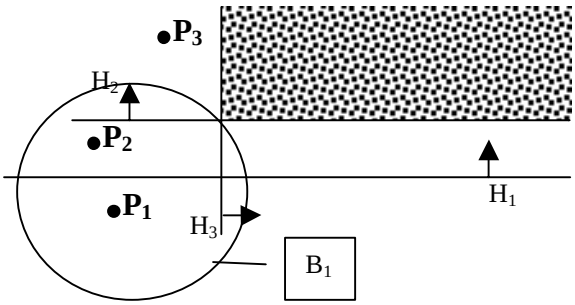

Figure 3. Une partition associée à l'arbre de la figure 1, avec des tests parallèles aux axes. La boule B_1 de centre P_1 contient les points proches de P_1 de même classe que ce point. L'hyperplan H_1 qui classe P_1 ne sépare pas les classes dans B_1.

Le premier exemple montre qu'un test de la trace peut ne pas être pertinent pour décrire la situation au voisinage du point classé. Le point P_1 est classé au premier test, qui n'est pas vérifié. La trace du classement, dans la figure 1, est limitée à $h(P,H_1) \leq 0$. Cependant on remarque que dans le cas de la figure 2, l'hyperplan H_1 délimite la surface de décision dans une zone de l'espace très éloignée de P_1 (en gras sur la figure). La situation est pire dans le cas de la figure 3 : l'hyperplan H_1 classe le point P_1 alors qu'il ne participe pas à la définition de la surface de décision. Dans les deux cas de figure, dans un voisinage de P_1, les points pour lesquels le premier test $h(P,H_1) > 0$ est vérifié (par exemple P_2), sont de la même classe que P_1. H_1 ne sépare donc pas les classes dans un voisinage de P_1. En ce sens, le test de la trace du classement de P_1 :

$$h(P_1,H_1) \leq 0$$

n'est pas un test pertinent pour représenter la situation au voisinage de P_1.

Le deuxième exemple montre que certains tests de la trace n'apportent pas d'informations. Par exemple, la trace du point P_3 est la suivante :

$h(P_3,H_1) > 0$
|_____ $h(P_3,H_2) > 0$
|_____ $h(P_3,H_3) \leq 0$

La figure 2 montre que dans un voisinage de P_3 (par exemple dans la boule B_3), seul le test $h(P,H_3) \leq 0$ sépare les deux classes. La trace comporte donc des tests superflus pour décrire la situation. Dans la boule B_3 tous les points vérifient $h(P,H_1) > 0$ et $h(P,H_2) > 0$, quelle que soit leur classe. La valeur de ces tests n'apporte donc pas d'information pour rendre compte de la classe d'un point.

L'exemple de P_2 montre que des tests qui n'apparaissent pas dans la trace sont pourtant essentiels pour décrire la situation. La trace du classement de P_2 est :

$h(P_2,H_1) > 0$
|_____ $h(P_2,H_2) \leq 0$

Comme dans le cas de P_1, H_1 n'est pas pertinent pour décrire la situation. Mais l'hyperplan H_2 ne suffit pas pour discriminer les deux classes dans un voisinage de P_2 (par exemple dans la boule B_2 en pointillé dans la figure 1). Les points proches de P_2 qui ne sont pas de la même classe que P_2 vérifient :

$$h(P,H_2) > 0 \text{ et } h(P,H_3) > 0.$$

C'est donc l'ensemble $\{H_2, H_3\}$ qui permet de rendre compte de la situation au voisinage du point P_2.

Ces exemples géométriques simples montrent que la trace n'est pas un bon support pour apporter à l'utilisateur des éléments de description de la situation. Les exemples précédents suggèrent une méthode pour rechercher les tests pertinents pour décrire la situation au voisinage de chaque point. Par exemple, on a vu sur la figure 2 que l'hyperplan H_3 sépare les classes au voisinage de P_3 (dans la boule B_3). Or c'est cet hyperplan qui porte le point de la surface de décision le plus proche de P_3. De même, sur la figure 3 on remarque que la boule B_1 atteint la surface de décision à l'intersection de H_2 et H_3. Comme dans le cas de P_2, ce sont bien ces deux hyperplans qui permettent de séparer les classes au voisinage de P_1 lorsque le rayon de la boule B_1 augmente.

C'est donc l'étude de la sensibilité au voisinage du point à classer qui permet d'identifier les tests pertinents pour décrire la situation.

3 Rechercher les tests sensibles pour expliquer un classement

3.1 Projeter les données sur la surface de décision

A partir des observations de la section précédente, on va s'intéresser au point de la surface de décision le plus proche du cas à expliquer. On se place dans le cas des arbres de décision opérant dans des espaces numériques, dont les tests sont linéaires (ADL), c'est-à-dire portés par des hyperplans. Alors la surface de décision est une hypersurface Γ, affine par morceaux (elle est constituée de morceaux d'hyperplans). Elle est continue et continûment dérivable presque partout. Le point de Γ le plus proche d'un point $\mathbf{x}$ de l'espace de départ E est en fait sa projection orthogonale sur Γ.

Soit $\mathbf{x} \in E$ un point de l'espace de départ, soit $p(\mathbf{x})$ le point[1] de Γ le plus proche de $\mathbf{x}$. Γ est définie localement en $p(\mathbf{x})$ par une liste d'hyperplans $L(\mathbf{x}) = (H_i)_{i \in I}$ (qui se réduit souvent à un seul hyperplan). Par exemple, dans la figure 3, la surface de décision est soit sur H_2, soit sur H_3, soit le point $H_2 \cap H_3$. Elle est donc définit localement soit par (H_2), soit par (H_3), soit par (H_2, H_3).

Si E est un espace euclidien, on montre facilement que le vecteur $\overrightarrow{\mathbf{x}p(\mathbf{x})}$ est orthogonal au vecteur normal de chaque hyperplan de la liste. L'ensemble de ces vecteurs forme justement une base de l'espace supplémentaire orthogonal de l'espace vectoriel associé à $\bigcap_{i \in I} H_i$.

On a donc :

(i) $\overrightarrow{\mathbf{x}p(\mathbf{x})}$ est orthogonal à $\bigcap_{i \in I} H_i$

On a de plus une propriété sur les tests correspondant aux hyperplans de $L(\mathbf{x})$:

Si Γ ne sépare que deux classes[2] au point $p(\mathbf{x})$, on peut montrer (voir [1]) que dans un voisinage de $\mathbf{x}$ assez petit, les points $\mathbf{y}$ dont la classe $c(\mathbf{y})$ est différente de celle de $\mathbf{x}$ sont du côté opposé de tous les hyperplans de la liste $L(\mathbf{x}) = (H_i)_{i \in I}$. C'est la propriété (ii).

On note d la distance de $\mathbf{x}$ à Γ. On a :

(ii) $\exists \varepsilon > 0, \forall \mathbf{y} \in B(\mathbf{x}, d+\varepsilon), \mathbf{y} \notin \Gamma,$
$c(\mathbf{y}) \neq c(\mathbf{x}) \Leftrightarrow (\forall i \in I, h(\mathbf{y}, H_i) \cdot h(\mathbf{x}, H_i) < 0)$

Les tests correspondants à ces hyperplans seront appelés les tests sensibles.

Par exemple dans le cas de P_2, dans la figure 2, pour changer de classe au voisinage de P_2, il est nécessaire et suffisant de traverser les hyperplans H_2 et H_3. Il y a donc deux tests sensibles pour P_2 :

$$h(P,H_2) \leq 0 \text{ et } h(P,H_3) \leq 0.$$

L'intérêt de proposer à l'utilisateur les tests sensibles en un point P pour expliquer le classement de ce point vient du fait que la projection est un opérateur continu pour les points génériques (c'est-à-dire ceux qui sont en dehors du squelette). Une petite modification des données entraîne une petite modification de la projection. La projection reste sur le(s) même(s) hyperplan(s) puisque Γ est affine par morceaux. (A l'exception des cas où P appartient à l'un de ces hyperplans). Les tests sensibles sont donc peu affectés par le bruit sur les données d'entrée.

3.2 Ordonner les tests sensibles dans le cas des arbres à tests parallèles aux axes (ADPA)

L'intérêt de présenter les tests sensibles à l'utilisateur est d'autant plus grand que la signification des tests lui est facilement accessible. C'est le cas des arbres de décision dont les tests portent sur un unique attribut. C'est un cas particulier des arbres à tests linéaires : les hyperplans qui portent les tests ont leur normale parallèle aux axes. Les hyperplans qui définissent la surface de décision sont donc tous parallèles ou orthogonaux.

Dans le cas des ADPA, il va être possible d'ordonner les tests sensibles en fonction de la contribution de la marge du test dans le calcul de la distance à la surface de décision. C'est la propriété (iii).

En effet, un algorithme très simple permet de calculer la projection d'un point $\mathbf{x}$ sur la surface de décision. Soit F l'ensemble des feuilles f dont la classe est différente de celle de $\mathbf{x}$. Il suffit, pour chaque feuille f de calculer la projection $p_f(\mathbf{x})$ de $\mathbf{x}$ sur cette feuille. La projection de $\mathbf{x}$ sur une feuille est obtenue en projetant récursivement le point courant $\mathbf{y}$ sur un des demi-espaces définissant la feuille, auquel $\mathbf{y}$ n'appartient pas. Quand le point courant $\mathbf{y}$ n'est plus à l'extérieur d'aucun des demi-espaces qui définissent la feuille, il est sur la surface de décision et $\mathbf{y} = p_f(\mathbf{x})$. La projection sur la surface de décision est le point $p_f(\mathbf{x})$ le plus proche de $\mathbf{x}$ (pour f parcourant F). Cette dernière opération consiste à calculer les distances $d(\mathbf{x}, p_f(\mathbf{x}))$ et à les trier.

Les opérations réalisées sont très simples (les projections sur les hyperplans consiste à affecter le biais de l'hyperplan à l'attribut correspondant du point courant). La complexité de l'algorithme est dans le pire des cas

---

[1] Le point le plus proche est unique sauf sur un ensemble très « petit », le squelette (qui est complémentaire d'un ouvert dense dans le cas des ADL, voir [1]).

[2] On se place dans le cas très général où une classe n'est jamais définie dans des zones de volume nul.

linéaire dans le nombre de tests. La dimension de l'espace des données n'intervient que dans la dernière étape, pour le calcul des distances entre le point initial et ses projections.

Cet algorithme permet, pour une métrique donnée, de présenter à l'utilisateur la liste des tests sensibles pour le cas étudié. (L'influence de la métrique est discutée dans la section suivante). On notera que si n est la dimension de l'espace de départ, il y a au plus n tests sensibles, quelle que soit la profondeur de l'arbre de décision. De plus, les tests sensibles peuvent être ordonnés en fonction de la distance du point à l'hyperplan correspondant au test. C'est la « marge » $|h(\mathbf{x},H)|$ du test.

DEFINITION D'UN ORDRE SUR LES TESTS SENSIBLES

Soient T(H_1) et T(H_2) deux tests sensibles pour le point **x**, portés par les hyperplans H_1 et H_2. On dira que T(H_1) est plus sensible que T(H_2) si et seulement si la marge de T(H_1) est supérieure à celle de T(H_2) :

(iii) $\quad T(H_1) \succ T(H_2) \Leftrightarrow \left|h(\mathbf{x}, H_1)\right| \geq \left|h(\mathbf{x}, H_2)\right|$

En effet, plus la marge d'un test est importante, plus elle contribue au calcul de la distance du point à la surface de décision, puisqu'on a :

$$d(\mathbf{x}, \Gamma) = \sqrt{\sum_{i \in I} \left|h(\mathbf{x}, H_i)\right|^2}$$

Plus la marge d'un test est importante, plus un changement de valeur du test contribue à rapprocher le point **x** de la surface de décision. Par exemple, dans la figure 3, pour le point P_2 de la surface de décision, il y a deux tests sensibles, portés par les hyperplans H_2 et H_3. On voit cependant que la projection de P_2 sur H_3 sera beaucoup plus proche de la surface de décision que sa projection sur H_2. En ce sens le test porté par H_3 contribue davantage au classement de P_2.

Cette propriété fait de la liste des tests sensibles un outil simple d'utilisation pour décrire la situation au voisinage d'un cas. Cette liste devrait être préférée à la trace du classement, non seulement parce que les tests de la liste ont davantage de pertinence, mais aussi parce que la contribution de chaque test peut être mesurée facilement (ce qui n'est pas le cas dans la trace). Il est ainsi possible de tronquer intelligemment la liste.

Comme illustration, on a utilisé la base Pima Indian Diabetes [18], de l'entrepôt de données du UCI. Les attributs de cette base sont des caractéristiques de patientes (âge, nombre de grossesses, indice de masse corporelle, etc.) et des résultats d'analyse médicale visant à expliquer un état de diabète. Les enregistrements proviennent de la population des Indiens Pima, chez qui la prévalence du diabète est importante. Plusieurs arbres de décision ont été construit suivant des algorithmes dérivés de C4.5 (le logiciel See5, v.1.06, et Weka [22]), sur une base d'apprentissage (2/3 des cas, tirés au hasard en respectant la prévalence du diabète dans la base, soit 35%). On constate dans le tableau 1 que le nombre des tests sensibles est plus petit que la taille de la trace.

Nombre de feuilles	28	25	19	11
Profondeur maximale	10	9	8	7
Dimension de l'espace (nombre d'attributs de l'arbre)	8	8	8	5
Taille de la Trace				
Moyenne	4.85	4.345	4.375	3.115
Médiane				
Nombre de tests sensibles				
Moyenne	1.531	1.752	1.71	1.71
Médiane				

Tableau 1. Caractéristiques de la trace du classement et du nombre de tests sensibles, calculés sur la base de test Pima (256 cas). Les tests sensibles ont été calculés avec une métrique de type min/max.

On notera que le nombre moyen de tests sensibles reste faible même lorsque la taille et la profondeur de l'arbre augmente. Cependant, pour d'autres bases de cas, comprenant de nombreux attributs, l'ensemble des tests sensibles pourrait devenir trop grand pour être compréhensible par l'utilisateur. Il serait alors nécessaire de tronquer la liste. Cette opération peut être réalisée aisément en présentant les tests dont la marge est la plus importante.

4 Utiliser la métrique comme support d'échange avec l'utilisateur

4.1 Influence de la métrique

La définition des tests sensibles pour un point dépend de la norme choisie pour l'espace des données. Cependant, comme on l'a vu dans la section 3.2, dans les cas des arbres à tests parallèles aux axes, la norme intervient principalement pour le calcul de la distance entre le point et ses projections sur les feuilles de classe différente de la sienne.

En effet, la projection sur les feuilles est invariante par dilatation, c'est-à-dire pour tout changement de coordonnées du type :

(iv) $\quad \mathbf{y} = {}^t\mathbf{a}.(\mathbf{x} - \mathbf{b})$

où ${}^t\mathbf{a}$ désigne le transposé d'un vecteur **a** de coefficients strictement positifs et **b** un point. Ce type de transformation conserve les hyperplans dont la normale est parallèle aux axes, les projections sur chaque feuille

sont donc inchangées. Seul le calcul de la distance entre le point initial et sa projection doit être effectué à nouveau.

Le choix de la première métrique nécessite un minimum d'information sur la distribution des données, car il est nécessaire d'homogénéiser les différents attributs (Le calcul des projections s'effectue sur des axes sans dimension). Par exemple, il faut disposer pour chaque attributs *i* d'une estimation des bornes (minimum et maximum), ou bien de la moyenne $\mathrm{E}(x_i)$ et de l'écart-type s_i. Le tableau 2 indique les changements de coordonnées correspondant. Si on utilise une normalisation bornée, quand un cas se présente en dehors des bornes, il est nécessaire d'effectuer un changement de variable de type (iv), ce qui entraîne uniquement le recalcul des distances du point à ses projetés (les points restent inchangés, seules les distances varient).

min/max	moyenne/écart-type (MET)
$y_i = \frac{x_i - \min_i}{\max_i - \min_i}$	$y_i = \frac{x_i - \mathrm{E}(x_i)}{s_i}$

Tableau 2. Changement de coordonnées pour les métriques initiales.

Dans le cas de la base Pima, le choix d'une métrique min/max ou d'une métrique moyenne/écart-type (MET) comme première métrique entraîne peu de changements, comme le montre le tableau 3. Trois métriques ont été utilisées : une métrique de type min-max dont les paramètres ont été calculés sur la totalité de la base, et deux métriques MET, dont les paramètres ont été estimés sur la totalité de la base ou sur la base de test.

Comparaison de Métrique	MET (base complète)	MET (base de test)
min/max (base complète)	2.34	3.13
MET (base complète)	-	1.95

Tableau 3. Pourcentage de points de la base de test Pima ayant une projection différente lors d'un changement de métrique.

4.2 La métrique comme support de description d'un cas

A partir de la métrique initiale, il est possible de présenter à l'utilisateur de l'arbre de décision les tests sensibles correspondant au point dont il veut prédire la classe.
Par exemple, on considère deux cas de la base test de la base Pima, dont les attributs sont décrits dans le tableau 4, et on utilise pour les classer un arbre de 11 feuilles (figure 4). Ces deux cas sont classés par la même feuille (voir la trace du classement dans le tableau 5). Le calcul des tests sensibles, suivant l'algorithme décrit dans la section 3.1, donne la même liste de tests pour ces deux cas, et ce résultat est identique pour une métrique initiale de type min/max ou de type MET. Ces tests sont rassemblés dans le tableau 6.

Attributs	***cas #188***	***cas #63***
Concentration de glucose dans le plasma (test oral)	90	93
Pression diastolique (mmHg)	68	50
Indice de masse corporelle (kg/m^2)	38.2	28.7
Antécédents diabétiques	0.503	0.356
Âge	27	23
Classe réelle	1 diabétique	0 non malade
Classe prédite	0	0

Tableau 4. Attributs des cas de la base Pima.

```
Concentration du glucose > 137.5: 1 (136/38) [C1]
Concentration du glucose ≤ 137.5:
|__Indice de masse corporelle ≤ 26.3: 0 (98/1)
   Indice de masse corporelle > 26.3:
   |__Age> 28.5:
      |_ Concentration du glucose ≤ 94.5:
      |  |__Concentration du glucose ≤ 43.2: 1 (2) [C2]
      |  |  Concentration du glucose > 43.2: 0 (30/3)
      |  Concentration du glucose > 94.5:
      |  |__Antécédents diabétiques ≤ 0.546: 0 (75/29)
      |     Antécédents diabétiques > 0.546: 1 (28/5) [C3]
      Age ≤ 28.5 :
      |__Concentration du glucose ≤ 127.5:0 (125/14)  *
         Concentration du glucose > 127.5:
         |__Pression diastolique > 73: 0 (4)
            Pression diastolique ≤ 73:
            |__Age ≤ 21.5: 0 (2)
               Age > 21.5:
               |__ Antécédents diabétiques ≤ 0.188: 0 (2)
                  Antécédents diabétiques > 0.188: 1 (10/1) [C4]
```

Figure 4. Arbre construit sur la base d'apprentissage issues des données Pima (coefficient d'élagage de 5%). La classe apparaît en gras, suivie du nombre de cas classés par la feuille et du nombre de cas mal classés. (La feuille qui classe les cas #188 et #63 est repérée par un astérisque, les feuilles de la classe opposée sont étiquetées C_1 à C_4,).

```
Concentration de glucose ≤ 137.5:
|__ Indice de masse corporelle > 26.3:
   |__ Age ≤ 28.5:
      |__Concentration de glucose ≤ 127.5: 0
                                    (125.0/14.0)
```

Tableau 5. Trace du classement des cas #188 et #63.

On peut vérifier sur la figure 4 que la modification des tests de la trace n'entraîne pas systématiquement de changement de classe, sauf pour le premier test (si la

concentration *g* de glucose devient supérieure à 137) ; Par exemple, une modification de l'indice de masse corporelle n'entraîne aucun changement, indépendamment des valeurs des autres attributs. Une modification de l'âge, couplée ou non à une modification modérée de la concentration *g* de glucose reste sans effet sur la classe.

TESTS SENSIBLES
Age $\leq$ 28.5
Antécédents diabétique (d) $\leq$ 0.546
Concentration de glucose (g) $\leq$ 94.5

Tableau 6. Tests sensibles pour les points #188 et #63.

Les tests sensibles ne reprennent qu'un seul test de la trace (le test sur l'âge), mais ils comprennent d'autres tests de l'arbre, qui définissent réellement la surface de décision au voisinage des cas #188 et #63. En fait ces cas sont dans une configuration qui ressemble à celle du point P_2 des figures 2 et 3, dans un espace de dimension plus élevée. On peut vérifier sur la figure 4 que la modification de l'ensemble des tests sensibles, conformément à la propriété (iii), entraîne bien un changement de classe (les cas seront alors classés par la feuille C_3).

Le tableau 7 présente les marges respectives de chaque test sensible en fonction de la métrique.

TESTS SENSIBLES	MARGE (MIN/MAX)		MARGE (MSE)	
	#188	#63	#188	#63
Age $\leq$ 28.5	0,025	0,092	0,133	1,012
$d \leq$ 0.546	0,018	0,081	0,127	0,453
$g \leq$ 94.5	0,022	0,007	0,138	0,963
distance à Γ	0,038	0,123	0,223	1,474

Tableau 7. Marge des tests sensibles.

Le cas #63 est éloigné de la surface de décision. Avec la métrique min/max comme avec la métrique MSE, le test le plus sensible (au sens de la définition de la section 3, c'est-à-dire le test qui contribue le plus à la distance à la frontière) est le test sur l'âge. Par contre, le cas #188 est très proche de la surface de décision, c'est pourquoi l'ordre des tests sensibles est modifié par le changement de métrique.

A partir de cette description initiale des cas, l'utilisateur peut proposer une modification de la métrique. Trois types de modifications paraissent intéressants :

- le blocage d'un attribut : l'utilisateur peut interdire la projection sur un ou plusieurs attributs, en fonction des connaissances dont il dispose. Par exemple, l'évaluation des antécédents diabétiques pourra dans certains cas être considérée comme fixe.

- Le blocage d'un sens de projection sur un (ou plusieurs) attribut(s) : l'utilisateur peut interdire un sens de projection. L'attribut correspondant peut alors seulement diminuer ou seulement augmenter. C'est le cas du nombre de grossesse ou de l'âge.

- La prise en compte d'un coût local de déplacement différent suivant les attributs : l'utilisateur peut indiquer, dans un voisinage du point qu'on cherche à décrire, un coût de déplacement relatif entre les attributs. Ce coût est exprimé sous la forme d'un vecteur de coefficients de pondération $(w_1,...w_n)$, où n est le nombre d'attributs.

Ce dernier type de contrainte se traduit par un changement de variable local autour du point **P** dont on décrit la situation. Les nouvelles coordonnées z_i se déduisent des coordonnées initiales y_i (obtenues par l'application de la métrique « initiale ») par les équations suivantes :

$$\begin{cases} z_i - z_i(\mathbf{P}) = \frac{1}{w_i}\left(y_i - y_i(\mathbf{P})\right) \\ z_i(\mathbf{P}) = y_i(\mathbf{P}) \end{cases}$$

On montre facilement que ce type de transformation est de type (iv), il suffit donc de recalculer les projections sur chaque feuille de classe différente de celle de P pour obtenir le nouveau résultat. En fait la pondération est appliquée sur les coordonnées déjà transformées (qui sont donc homogènes et sans unités).

La modification de la métrique se fait par l'intermédiaire d'un menu spécifique. L'utilisateur peut ajouter des contraintes en sélectionnant un attribut et en choisissant le type de contrainte bloquante qu'il désire définir : interdire la modification de la valeur de l'attribut, interdire la diminution ou l'augmentation de la valeur de l'attribut. Il peut aussi fixer un poids relatif entre les attributs, en renseignant un vecteur de poids.

Pour les cas #188 et #63 de la base Pima, il n'y a que quatre feuilles de l'arbre qui permettent de les classer différemment de leur classe actuelle : ce sont les feuilles étiquetées C_1 à C_4 (voir figure 4). Le tableau 8 montre comment les cas #188 et #63 se projettent sur ces feuilles, c'est-à-dire les plus petites modifications nécessaires pour amener les cas sur ces feuilles. Pour chaque feuille, on applique l'algorithme de projection : on ne retient que les hyperplans correspondant aux tests qui ne sont pas vérifiés par le cas projeté.

Feuille	*Hyperplans définissant la surface de décision au point de projection*
C_1	Concentration de glucose = 137.5
C_2	Concentration de glucose = 43.2 Âge = 28.5
C_3	**Âge = 28.5** **Concentration de glucose = 94.5** **Ascendance diabétique = 0.546**
C_4	Concentration de glucose = 127.5

Tableau 8. Points de projection des cas #188 et #63 sur les feuilles de classe 1 « diabétique » (les attributs qui n'apparaissent pas gardent leur valeur pour chaque cas).

En l'absence de contrainte, les cas #188 et #63 se projettent sur la feuille C_3 (on reconnaît les tests sensibles du tableau 6). Si l'utilisateur définit une contrainte de blocage pour l'attribut « ascendance diabétique », le tableau 8 montre qu'une modification des valeurs des attributs des cas #188 et #63 ne leur permet plus d'atteindre la feuille C_3. En effet, la valeur de l'ascendance diabétique ne peut être amenée à 0.546, elle restera fixée aux valeurs initiales qui apparaissent dans le tableau 4 (0.503 et 0.306 pour les cas #188 et #63 respectivement). Pour tenir compte d'une contrainte de blocage, l'algorithme de projection doit être appliqué sur les feuilles qui restent atteignables. Il s'agit donc de déterminer le point de projection le plus proche parmi ces feuilles. Le tableau 9 donne la distance (min/max) des cas aux différentes feuilles de classe 1 « diabétique ».

Feuille	***Distance au cas # 188***	*distance au cas #63*
C_1	0.239	0.224
C_2	0.236	0.266
C_3	-	-
C_4	0.188	0.173

Tableau 9. Distance des cas aux différentes feuilles (métrique initiale min/max).

La feuille C_3 ne pouvant plus être atteinte, c'est la feuille C_4 qui devient la plus proche. Si l'utilisateur définit une contrainte bloquante sur l'attribut « ascendance diabétique », un seul test sensible lui sera présenté, comme indiqué dans le tableau 10. (Le résultat est identique pour la métrique MSE).

TEST SENSIBLE (ASC. DIABÉTIQUE FIXE)
Concentration de glucose $\leq$ 127.5

Tableau 10. Tests sensibles pour les cas #188 et #63 en présence de contrainte bloquante.

Si l'utilisateur définit une contrainte sur le sens de projection de l'âge, le résultat reste inchangé. En effet, les cas #188 et #63 sont plus « jeunes » que la valeur de l'âge qui définit leur projection sur les feuilles de classe 1 « diabétique » (28.5 contre 27 et 23 respectivement). Si l'utilisateur définit une contrainte sur le sens de projection de la concentration de glucose, le résultat peut être modifié. Par exemple, si la concentration de glucose ne peut augmenter, seule la feuille C_2 sera atteignable.

En revanche, même si l'utilisateur modifie le poids relatif des attributs (par exemple en diminuant le poids de l'âge), il n'est pas possible dans ces exemples de modifier l'ordre des feuilles. En effet, on constate en examinant le tableau 8 que l'attribut mesurant la concentration de glucose dans le plasma est présent dans la définition de la surface de décision au point de projection sur chacune des feuilles. Un changement de métrique qui éloigne les hyperplans définis par cet attribut éloigne toutes les feuilles de la même manière. Il serait donc possible dans ce cas particulier de renseigner l'utilisateur sur la robustesse des tests sensibles face à un changement de métrique.

Par ailleurs, il est possible d'indiquer à l'utilisateur la position relative des cas par rapport à la surface de décision, en comparant la distance au point projeté à une grandeur caractéristique de la zone des points de même classe. Par exemple, la distance du cas #188 à la surface de décision est de 0.038, alors que la distance maximale possible est d'environ 0.6. Le cas #63, en revanche, est plus éloigné (sa distance est de 0.123). L'utilisateur peut ainsi tenir compte de la distance à la surface de décision (par exemple en présence de bruit sur les données) pour accepter la classe proposée par l'arbre de décision.

Conclusion

Nous avons présenté une approche basée sur la géométrie pour rechercher dans un arbre de décision des éléments d'information permettant d'expliquer à l'utilisateur le classement d'un point. Cette approche est simple à mettre en œuvre dans le cas des arbres de décision numériques dont les tests sont parallèles aux axes. Elle permet de proposer un nombre limité de tests, ordonnés suivant leur importance, qui doivent être modifiés pour entraîner une modification du résultat. Dans le cas des arbres obliques (voir par exemple [2]), la même méthode pourrait être utilisée, mais elle impose l'élimination dans l'algorithme de projection des hyperplans qui ne définissent pas la surface de décision au voisinage du point le plus proche. Cette méthode peut aussi théoriquement s'appliquer aux familles d'arbres (comme le boosting ou le bagging, voir [4] et [8]). Mais son application impose le calcul des feuilles de l'arbre correspondant. Des tests sont aussi nécessaires pour vérifier que la mise en œuvre reste possible quand le nombre de tests de l'arbre est grand et pour des espaces comprenant un très grand nombre d'attributs. En effet, dans ces espaces, le calcul de distance euclidienne peut devenir coûteux.

Les possibilités d'échange avec l'utilisateur par l'intermédiaire de la métrique doivent aussi être testés dans des espaces de plus grande dimension et de manière systématique, en particulier pour tester les contraintes de pondération entre les attributs.
Un autre aspect intéressant est la possibilité de définir un prototype géométrique dans un voisinage du point étudié. En dilatant la surface de décision vers le point, il est possible de trouver une boule maximale de même classe que ce point et qui le contienne. Cette boule maximale permet de faire le lien avec des zones éloignées de la surface de décision et de s'abstraire de la feuille qui classe le point en explorant les feuilles connexes. Dans le cas des arbres à tests parallèles aux axes, il sera ainsi possible de décrire davantage la zone dans laquelle se trouve le point à classer.

Bibliographie

[1] I. Alvarez. *Explication géométrique du résultat dans les arbres de décision.* Revue d'Intelligence Artificielle. A paraître.

[2] K. Bennett & J. Blue, *Optimal decision trees*. (Technical Report 214), R.P.I. Math. Science Dept., Troy, NY 12180, 1996.

[3] L.Breiman, J. H. Friedman, R. A. Olshen, & C. J. Stone. *Classification and Regression Trees*, Belmont: Wadsworth, 1984.

[4] L. Breiman. Bagging Predictors, *Machine Learning*, Vol. 24, No. 2, pp. 123-140, 1996.

[5] B.Chandrasekaran, M.G. Tanner, J.R. Josephson, Explaining control strategies in problem solving, *IEEE expert*, pp. 9-24, Printemps 1989.

[6] B. Efron & R.J. Tibshirani. *An introduction to the bootstrap*, Chapman and Hall, 1993.

[7] F. Esposito, D. Malerba & G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), 476-491, 1997.

[8] Y. Freund & R.E. Schapire. Experiments with a New Boosting Algorithm, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, Morgan Kaufmann, 1996.

[9] Hasling D.W., Clancey W.J., Rennels G. Strategic explanations for a diagnostic consultation system, *Int J. Man-Machine Studies*, vol 20, pp. 3-19, 1984.

[10] G. Kassel. CQFE, Thèse de l'Université de Paris XI, Décembre 1986.

[11] M. J. Kearns & D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pp. 152-162, Morgan Kaufmann, 1997.

[12] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1137-1143, San Mateo: Morgan Kaufmann, 1995.

[13] S.K. Murthy. Automatic construction of decision trees from data: A multidisciplinary survey. *Data Mining and Knowledge Discovery*, *2*, 4, 345-389, 1998.

[14] C. Paris, M.R. Wick, W.B. Thompson. The line of Reasoning versus the line of Explanation, AAAI'88 Workshop on Explanation, pp. 4-87, 1988.

[15] J. R. Quinlan. *C4.5: Programs for Machine Learning*, San Mateo: Morgan Kaufmann, 1993.

[16] B. Safar. Le problème des explications négatives dans les systèmes experts: le système Pourquoi-Pas, Thèse de l'Université de Paris XI, Décembre 1987.

[17] E.H. Shortliffe. *Computer-based medical consultations: MYCIN*, New York, Elsevier, 1976.

[18] V. Sigillito, Pima Indian Diabetes Database, *UCI Repository of machine learning databases*. University of California, Department of Information and Computer Science. Irvine, CA, 1990.

[19] W.R. Swartout. "XPLAIN: a system for creating and explaining expert consulting systems". *Artificial Intelligence*, 21(3), pp.285-325, 1983.

[20] M. Umano, K. Okomato, I. Hatono, H. Tamura, F. Kawachi, S. Umezu, J. Kinoshita Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems, *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, pp. 2113-2118, 1994.

[21] M.R. Wick, J.R. Thomson, Reconstructive expert system explanation, *Artificial Intelligence*, 54, pp. 33-70, 1992.

[22] I. Witten, E. Frank, *Data Minig: Practical Machine Learning Tools and Techniques with Java Implementation*, Morgan Kaufmann, 2000.