

# Arbres de décisions

Synthèse de lectures

(Séminaire sur l'apprentissage automatique)

de

Benoit Lavoie

[benoit@benoit-lavoie.ca](mailto:benoit@benoit-lavoie.ca)

Programme de Doctorat en Informatique Cognitive

Université du Québec à Montréal

15 mars 2006

# Table des matières

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. USAGES DES ARBRES DE DÉCISIONS .....</b>	<b>1</b>
<b>3. REPRÉSENTATION ET INTERPRÉTATION DES ARBRES DE DÉCISIONS.....</b>	<b>1</b>
<b>4. DONNÉES D'ENTRÉES POUR LA CONSTRUCTION ET L'UTILISATION DES ARBRES DE DÉCISIONS.....</b>	<b>2</b>
ATTRIBUTS À VALEURS NUMÉRIQUES .....	2
ATTRIBUTS À VALEURS MANQUANTES .....	2
ATTRIBUTS À VALEURS BRUITÉES .....	2
<b>5. INDUCTION D'ARBRES DE DÉCISIONS .....</b>	<b>3</b>
CONSTRUCTION NON-INCRÉMENTALE (BATCH) .....	3
CONSTRUCTION INCRÉMENTALE.....	3
ÉLAGAGE D'ARBRES DE DÉCISIONS.....	4
CRITÈRES DE DIVISION DES DONNÉES .....	4
<b>6. ALGORITHMES D'INDUCTION D'ARBRES DE DÉCISIONS.....</b>	<b>5</b>
ID3: ALGORITHME FONDATEUR .....	5
C4.5: ALGORITHME POPULAIRE .....	6
ID5R : ALGORITHME INCRÉMENTAL.....	6
SALAMMBÔ: ALGORITHME DE LOGIQUE FLOUE .....	7
ALGORITHMES PROBABILISTES .....	7
<b>7. AVANTAGES ET INCONVÉNIENTS DES ARBRES DE DÉCISIONS .....</b>	<b>8</b>
AVANTAGES DES ARBRES DE DÉCISIONS .....	8
INCONVÉNIENTS DES ARBRES DE DÉCISIONS .....	8
<b>8. RÉFÉRENCES .....</b>	<b>9</b>

## **1. Introduction**

Ce document présente une synthèse de travaux sur les arbres de décisions faisant ressortir les grandes tendances de cette approche. Ce travail a été réalisé dans le cadre d'un séminaire sur l'apprentissage automatique (DIC9380) du programme de Doctorat en Informatique Cognitive de l'Université du Québec à Montréal (UQAM).

## **2. Usages des arbres de décisions**

Les arbres de décisions (Nilsson, 1996; Lounis, 2006b) sont associés à des représentations et à des algorithmes souvent utilisés en apprentissage automatique (Lounis, 2006a). Les arbres de décisions permettent la prédiction de classes (concepts ou cas) à partir d'exemples: ils sont utilisés pour l'apprentissage par inférence inductive qui est un processus permettant d'inférer des conclusions générales à partir d'exemples. On les utilise principalement dans le cadre de l'apprentissage supervisé ne requérant pas de connaissances du domaine. Un des domaines d'application populaire des arbres de décisions est celui de la fouille de données (Witten & Frank, 2005; Negnevitsky, 2002).

## **3. Représentation et interprétation des arbres de décisions**

De façon générale, un arbre de décisions est un arbre dont les nœuds internes représentent des tests sur les attributs des données d'entrées, et dont les feuilles représentent des classes (concepts ou cas) correspondantes aux données (Nilsson, 1996; Lounis, 2006b). Les tests des nœuds d'un arbre de décisions visent à couvrir les valeurs des attributs des données d'entrées de façon exhaustive. Chaque test peut porter sur une seule variable (c'est le cas en général) ou sur plusieurs. Les tests sur les valeurs peuvent être exclusifs ou être pondérés avec des probabilités.

L'interprétation d'un arbre de décisions correspond à celui d'une forme normale disjonctive: chaque chemin menant de la racine de l'arbre vers une feuille peut s'interpréter comme une conjonction de valeurs d'attributs; les branches menant vers des feuilles dont la classe est similaire peuvent s'interpréter comme une disjonction. Des règles en forme normale disjonctive peuvent être extraites directement à partir des arbres de décisions (Lounis, 2006b, Witten & Frank, 2005). Ces règles sont non-ambigües et insensibles à l'ordre mais peuvent être complexes et peuvent généralement être simplifiées automatiquement par des heuristiques.

Les arbres dont les classes sont nominales sont parfois appelés arbre de classification alors que les arbres dont les classes sont numériques sont parfois appelés arbres de régression (Wikipedia, 2006).

## **4. Données d'entrées pour la construction et l'utilisation des arbres de décisions**

Chaque donnée d'entrée est spécifiée par un ensemble d'attributs et est associée à une classe: cette classe est spécifiée a priori lors de l'apprentissage et doit être prédite à partir de l'arbre de décisions après apprentissage. Chaque attribut est spécifié par un nom et une valeur. Différents problèmes sont reliés au traitement des attributs (Lounis, 2006b): (i) les attributs peuvent avoir des valeurs numériques, nominales ou les deux, (ii) les valeurs d'attributs peuvent être manquantes, ou (iii) les valeurs ou noms d'attributs peuvent être bruités (i.e. qu'ils peuvent contenir des erreurs). Les fonctionnalités supportées concernant les attributs varient selon les algorithmes de traitement d'arbres de décisions (voir détails plus bas).

### **Attributs à valeurs numériques**

Les attributs à valeurs numériques se retrouvent dans beaucoup de bases de données. Cependant, les premiers algorithmes reliés au traitement d'arbres de décisions n'ont d'abord bien supportés que les attributs à valeurs nominales (Witten & Frank, 2005); les valeurs numériques étaient parfois traitées comme des valeurs nominales ce qui pouvait engendrer des arbres de grande taille. Dans les algorithmes supportant les attributs à valeurs numériques, les tests des nœuds d'un arbre de décisions portant sur des attributs numériques se font généralement en sélectionnant une valeur seuil permettant de séparer les données en deux ensembles (voir *Critères de division des données* à la Section 5 pour plus de détails). Marsala et Bouchon-Meunier (1999) proposent une approche de logique floue dont les tests sur les valeurs d'attributs (aussi bien numériques que nominales) partitionnent l'ensemble des données selon une ou plusieurs valeurs floues (voir l'algorithme de Salammbô à la Section 6).

### **Attributs à valeurs manquantes**

Les attributs à valeurs manquantes sont fréquents dans les bases de données (Witten & Frank, 2005; Quinlan, 1989). Quinlan (1989) présente une étude sur un algorithme dont les paramètres permettent de traiter de façon variée des bases de données contenant des valeurs manquantes. Cette étude indique que prendre en considération le statut particulier des attributs à valeurs manquantes permet d'offrir une meilleure performance que de l'ignorer. Cette étude indique également qu'une classification qui combine différents résultats probables à partir des valeurs connues offre la meilleure performance parmi les approches étudiées.

### **Attributs à valeurs bruitées**

De façon générale, les données peuvent être bruitées et elles peuvent contenir des erreurs au niveau des noms d'attributs ou de leurs valeurs. Les arbres de décisions peuvent répliquer ou non ces erreurs (i.e.

coller aux données) selon les algorithmes utilisés (Nilsson, 1996; Lounis, 2006b). Lors de l'apprentissage, un mécanisme d'élagage peut être utilisé afin de traiter les erreurs (voir Section 5). Les données dont les attributs sont identiques mais dont les classes sont différentes peuvent être traitées de façon à sélectionner comme classe celle qui est la plus fréquente.

## 5. Induction d'arbres de décisions

### Construction non-incrémentale (*batch*)

La construction non-incrémentale d'arbres de décisions consiste à construire l'arbre à partir d'un ensemble d'exemples de données spécifiées a priori. Il y a différents algorithmes de construction non-incrémentale d'arbres de décisions (e.g ID3, C4.5 et l'algorithme de Salammbô décrits à la Section 6). Une approche généralement utilisée consiste à construire l'arbre de façon descendante (top-down), récursive et gloutonne, à partir des exemples de données (Nilsson, 1996; Lounis, 2006b). À partir de la racine, l'algorithme utilise un critère de division (voir plus bas) afin de choisir un attribut<sup>1</sup> dont les valeurs permettent la meilleure répartition des données en différents regroupements; la motivation est de réduire la taille de l'arbre en choisissant d'abord l'attribut estimé comme étant le plus discriminatif. Les données sont ensuite répartitionnées selon les valeurs de cet attribut de manière à créer une branche pour chaque regroupement de données. Le processus s'applique récursivement pour chaque branche. Ce processus de construction s'arrête, pour une branche donnée, lorsqu'une condition d'arrêt est atteinte: e.g. les données regroupées à un noeud donné ne peuvent plus être (suffisamment) distinguées par les attributs restants. Certains algorithmes intègrent un processus d'élagage (décrit plus bas) pendant ou après la construction de l'arbre.

### Construction incrémentale

La construction incrémentale d'arbres de décisions permet la construction d'arbres de décisions puis la révision de ceux-ci à partir de nouveaux exemples d'apprentissage (Utgoff, 1989). La motivation pour réviser un arbre existant plutôt que de le reconstruire est relié au coût inférieur de la révision en terme de temps de traitement. Il y a différents algorithmes de construction d'arbres de décisions. Utgoff (1989) présente deux techniques dont ID5R décrite à la Section 6.

---

<sup>1</sup> La division des données se fait généralement à partir que d'un seul attribut à la fois à cause de la difficulté de combiner différents attributs. Cependant, certains algorithmes supportent la division à partir de plusieurs attributs.

## **Élagage d'arbres de décisions**

Certains algorithmes de construction d'arbres de décisions permettent l'élagage (Malerba et al, 1996; Esposito et al, 1997). Le but de l'élagage est de simplifier les arbres de façon à éviter de rendre les tests sur les attributs trop spécifiques aux données d'entraînement et non suffisamment généraux pour de nouvelles données. L'élagage peut se faire après la construction de l'arbre (on parle alors de post-élagage) ou pendant la construction (on parle alors de pré-élagage). Le post-élagage est la méthode la plus utilisée à cause de la difficulté à déterminer quand arrêter la croissance de l'arbre pendant sa construction.

Des travaux (Malerba et collègues, 1996; Esposito et collègues 1997) présentent une étude portant sur 8 méthodes standards de post-élagage. Les méthodes présentées offrent différentes caractéristiques: (i) certaines méthodes appliquent l'élagage de façon descendante alors que d'autres le font de façon ascendante; (ii) certaines méthodes utilisent pour l'élagage le même ensemble d'exemples d'entraînement que celui utilisé pour la construction de l'arbre, alors que d'autres utilisent des exemples d'entraînement qui sont distincts de ceux utilisés pour la construction de l'arbre; (iii) certaines méthodes n'appliquent que des opérateurs d'élagage de branches alors que d'autres permettent le rattachement de branches. Dans les travaux rapportés, ces méthodes de post-élagage ont été appliquées sur les arbres produits par le même algorithme de construction pour une douzaine de bases de données dans différents domaines variant en terme de nombre de classes, nombre et type d'attributs, taux de bruit dans les données, etc. Les arbres élagués produits par ces méthodes ont ensuite été analysés aux niveaux du taux d'erreurs et de la taille des arbres: les taux d'erreurs des arbres élagués ont été comparés avec ceux des arbres non-élagués; les tailles des arbres élagués ont été comparées avec celles d'arbres élagués obtenus avec une méthode prouvée optimale et en utilisant des corpus similaires pour l'élagage (i.e. corpus distincts ou non du corpus de construction d'arbres). Les conclusions des analyses effectuées indiquent que les méthodes d'élagage utilisant le même corpus d'entraînement que pour la construction d'arbres produisent généralement des taux d'erreurs moindres lors de tests que les méthodes utilisant un sous-ensemble distinct du corpus d'entraînement pour l'élagage; ceci est dû au nombre des données disponibles pour l'entraînement qui est souvent limité. Les méthodes d'élagage permettent de produire des résultats lors de tests dont les taux d'erreurs sont significativement moindres ou comparables aux arbres non-élagués. Certaines méthodes d'élagage ont tendance à trop élaguer alors que d'autres ont tendance à ne pas élaguer suffisamment.

## **Critères de division des données**

Différents critères peuvent être utilisés afin de déterminer comment partitionner les données à partir des attributs. De façon générale, ces critères visent à minimiser la taille de l'arbre en sélectionnant d'abord

les attributs les plus discriminants (Lounis, 2006b). Le reste de cette section décrit quelques critères de division et la Section 6 décrit des algorithmes utilisant quelques-uns de ces critères. Ces critères s'appliquent aussi bien pour des valeurs nominales que numériques. Dans le cas de valeurs numériques, ces critères s'appliquent souvent sur des valeurs seuils déterminées en comparant les moyennes entre chaque paire de valeurs numériques consécutives. Il est à noter que le partitionnement est généralement exhaustif pour les attributs nominaux alors qu'il est souvent partiel pour les attributs numériques; dans une branche donnée, un attribut nominal n'apparaîtra généralement qu'une fois alors qu'un attribut numérique sera souvent re-testé plusieurs fois avec des seuils différents.

Un des critères de division utilisé dans certains algorithmes (e.g. version originale de l'algorithme ID3 décrit à la Section 6) est celui du gain d'information. Cette métrique est basée sur la mesure d'entropie qui correspond au degré d'incertitude dans un système. Dans le contexte des arbres de décisions, l'entropie correspond au nombre de questions booléennes (vrai/faux) nécessaires afin de déterminer une classe: l'entropie est proportionnelle au degré d'incertitude sur la valeur d'une classe. Le gain en information est une mesure permettant de savoir comment un attribut permet de diminuer l'entropie (l'incertitude) concernant la valeur d'une classe en partitionnant les données avec cet attribut.

Un second critère de division utilisé dans certains algorithmes (e.g. algorithme C4.5 décrit à la Section 6) est celui de ratio de gain d'information. La métrique de gain d'information est biaisée pour les attributs ayant le plus grand nombre de valeurs (i.e. produisant le plus de branches). Le ratio de gain d'information corrige le biais du gain d'information en prenant en considération le nombre de valeurs d'un attribut et la proportion de ces valeurs dans les données.

Un troisième critère de division utilisé dans certains algorithmes (e.g. algorithme CART) est celui du coefficient d'impureté Gini. Ce coefficient pour un nœud donné est déterminé en soustrayant, au nombre 1, la somme des carrés des fréquences de chaque classe. L'attribut produisant la partition ayant le plus bas coefficient est sélectionné.

## **6. Algorithmes d'induction d'arbres de décisions**

### **ID3: algorithme fondateur**

ID3 est un algorithme non-incrémental d'induction d'arbres de décisions développé vers 1983 par Ross Quinlan (citation: Hamilton, 2006). C'est le premier algorithme populaire à avoir été employé pour les arbres de décisions. L'approche employée par ID3 se limite à la construction d'un arbre de décisions sans élagage. L'arbre est construit de façon non-incrémentale descendante gloutonne et la mesure du gain d'information est utilisée pour le partitionnement des données (voir Section 5 pour plus de détails). ID3

utilise des tests basés sur la métrique chi-carré ( $\chi^2$ ) afin de déterminer si un attribut donné est suffisamment pertinent pour la classification. Les limites principales d'ID3 incluent les suivantes:

- L'algorithme original d'ID3 utilise un critère de division (gain d'information) qui est biaisé pour les attributs ayant le plus grand nombre de valeurs (voir Section 5).
- ID3 ne supporte pas bien les attributs numériques.
- ID3 ne supporte pas bien les valeurs manquantes (qui sont traitées comme des valeurs ordinaires).
- ID3 produit des arbres pouvant coller trop près aux données d'entraînement à cause qu'il n'utilise pas de technique d'élagage.

#### **C4.5: algorithme populaire**

C4.5 est un algorithme non-incrémental d'induction d'arbres de décisions développé vers 1993 par le développeur d'ID3, Ross Quinlan (citation: Hamilton, 2006). C4.5 offre plusieurs améliorations à ID3 en résolvant les limites d'ID3 mentionnées plus haut:

- C4.5 utilise le ratio du gain d'information comme critère de division (ce qui est une amélioration au gain d'information utilisé par ID3).
- Contrairement à ID3, C4.5 supporte les attributs numériques.
- Contrairement à ID3, C4.5 traite les valeurs manquantes de façon spécifique.
- Contrairement à ID3, C4.5 utilise une technique d'élagage (i.e. post-élagage).
- Finalement, C4.5 peut également produire des règles à partir d'arbres de décisions.

Comme pour ID3, l'arbre est construit de façon non-incrémentale descendante gloutonne. C4.5 est l'un des algorithmes d'induction d'arbres de décisions les plus connus et les plus utilisés aujourd'hui.

#### **ID5R : algorithme incrémental**

ID5R est un algorithme incrémental d'induction d'arbres de décisions (Utgoft, 1989). L'algorithme révisé l'arbre courant au besoin en réponse à de nouvelles données; l'arbre est révisé simplement lorsque la nouvelle donnée n'est pas bien classifiée. ID5R utilise ID3 et peut produire les mêmes arbres qu'ID3 pour les mêmes données d'apprentissage. Le processus global de construction d'arbre (et de branches)

d'ID5 est le même que celui d'ID3: processus descendant glouton utilisant la mesure du gain d'information comme critère de sélection du meilleur attribut pour la division.

Pour la révision de l'arbre, ID5R maintient des informations pertinentes sur les données d'entraînement au niveau des nœuds et des feuilles de l'arbre. Avec ID5R, un nœud interne contient un attribut test (tout comme ID3) et l'ensemble des autres attributs non encore utilisés. Chacun de ces attributs est associé au nombre de valeurs possibles de cet attribut. Pour les révisions de l'arbre, ces informations permettent à ID5R de recalculer les gains d'information et de déplacer des nœuds de façon appropriée sans à avoir à tout reconstruire l'arbre. ID5R utilise une technique de remontage (pull-up) permettant de déplacer un attribut vers le haut sans avoir à réexaminer les données tout en gardant l'arbre consistant avec celles-ci.

### **Salammbô: algorithme de logique floue**

Salammbô (Marsala & Bouchon-Meunier, 1999) est un algorithme non-incrémental d'induction d'arbres de décisions basé sur la logique floue. Cet algorithme a été développé afin de mieux traiter les données dont les attributs contiennent des éléments numériques (continus) puisque ces données se partitionnent souvent moins bien que les données symboliques (discrètes). Les arbres de décisions induits par les algorithmes présentés précédemment dans cette section utilisent des tests disjoints portant sur des valeurs bien définies pour partitionner les données: cela résulte en des données dont l'appartenance à une partition est définie de façon booléenne (une donnée appartient ou n'appartient pas à une partition). Salammbô et d'autres algorithmes flous comparables induisent des tests flous portant sur des valeurs floues: cela résulte en des données dont l'appartenance à une partition est définie de façon floue (i.e. permettant l'appartenance à une ou plusieurs partitions à des degrés divers). Salammbô supporte différents critères de division: un de ceux supporté est basé sur la métrique du gain d'information où les fréquences sont remplacées par des mesures floues. Cependant, Salammbô ne supporte pas l'élagage.

Marsala et Bouchon-Meunier (1999) comparent les résultats expérimentaux obtenus avec Salammbô et C4.5 sur deux petits corpus de données contenant des attributs numériques. Les résultats rapportés indiquent que Salammbô semble offrir une meilleure classification que C4.5 sur ces corpus avec des taux d'erreurs inférieurs.

### **Algorithmes probabilistes**

Les algorithmes de logique floue pour induire des arbres de décisions tels que Salammbô sont comparables aux approches probabilistes (Quinlan, 1993). La motivation des approches probabilistes est de pouvoir supporter le caractère incertain de la classification. Dans les approches qui ne sont ni probabilistes et ni floues, les données sont classifiées de façon disjointe. Cependant, l'appartenance à une

classe peut être incertaine à cause de plusieurs facteurs: bruit dans les données, données manquantes, valeurs numériques près de seuils utilisés par les critères de divisions. Quinlan (1993) propose une approche probabiliste permettant à une feuille de représenter plusieurs classes selon différentes probabilités et à une observation donnée de pouvoir être mise en correspondance avec plusieurs feuilles: une combinaison des probabilités assignées par les feuilles indique la classe d'appartenance probable de la donnée. Chaque classe d'appartenance est assignée avec 3 paramètres de probabilités dont une estimation inférieure, une estimation centrale et une estimation supérieure pour qu'une donnée appartienne à la classe.

## **7. Avantages et inconvénients des arbres de décisions**

### **Avantages des arbres de décisions**

Les arbres de décisions offrent plusieurs avantages (Boukadoum, 2006; Wikipedia 2006):

- Production de règles claires et explicites; les règles sont facilement compréhensibles par pour les personnes après peu d'explications.
- Indication explicite des attributs qui sont les plus pertinentes pour la classification.
- Facilité du calcul au moment de la classification; les algorithmes appliquant les arbres de décisions sont simples et nécessitent peu de calcul.
- Bonne efficacité dans les domaines basés sur des règles; la correspondance avec une base de règles est simple.
- Capacité de manipuler des variables numériques et/ou nominales; certains autres types de représentations ne traitent que d'un type d'attribut (numérique ou nominale);
- Dérivation d'arbres compacts avec une grande capacité de classification lorsque les données se prêtent au groupement.
- Facilitation du traitement robuste de grande quantité de données en peu de temps; certains algorithmes supportent le bruit.
- Peu de prétraitement de données requis; certains algorithmes supportent les données non-spécifiées et/ou le bruit.

### **Inconvénients des arbres de décisions**

Les arbres de décisions sont aussi liés à quelques désavantages (Boukadoum, 2006):

- Manque de performance s'il y a beaucoup de classes; les arbres peuvent être très complexes et ne sont pas nécessairement optimaux.
- Sensibilité au manque de données; les arbres peuvent trop coller aux données (*data overfitting*) lorsque celles-ci ne sont pas assez représentatives et l'élagage peu ne pas entièrement résoudre le problème.
- Moins bonnes performances pour les prédictions portant sur des valeurs numériques.
- La construction et élagage des arbres de décisions nécessitent généralement beaucoup de temps calcul: le choix du meilleur partitionnement lors de la construction et la comparaison de sous-arbres lors de l'élagage sont souvent coûteux.
- La combinaison de différents attributs pour les tests n'est pas toujours bien traitée automatiquement; la plupart des algorithmes ne traitent que d'un attribut à la fois.

## 8. Références

Boukadoum, Mounir (2006). *Évaluation de variables hydro-électriques: une comparaison de différentes approches d'apprentissage automatique*. Notes de séminaire en informatique cognitive, Université du Québec à Montréal, 23 février.

Esposito, Floriana; Donato Malerba; & Giovanni Semeraro (1997). A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 476-491.

Hamilton, Howard J. (2006). *Decision Trees*. Lecture notes (Knowledge Discovery in Databases), University of Regina.

[http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/4\\_dtrees1.html](http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/4_dtrees1.html)

Lounis, Hakim (2006a). *Apprentissage automatique*. Notes de cours (Séminaire sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.

Lounis, Hakim (2006b). *Arbres de décision*. Notes de cours (Séminaire sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.

- Malerba, Donato; Floriana Esposito; & Giovanni Semeraro (1996). A Further Comparison of Simplification Methods for Decision-Tree Induction, *Learning From Data: Artificial Intelligence and Statistics V*, D. Fisher and H. Lenz, eds., Lecture Notes in Statistics. Berlin: Springer, no. 112, pp. 365-374.
- Marsala, Christophe; & Bernadette Bouchon-Meunier (1999). An Adaptable System to Construct Fuzzy Decision Trees. *Proceedings of the North American Fuzzy Information Processing Society Conference, NAFIPS'99*, New York, United States, pp. 223-297.
- Negnevitsky, Michael (2002). *Data mining and knowledge discovery*. Section 9.5 of *Artificial Intelligence: A Guide to Intelligent Systems*. Addison-Wesley, pp. 330-340.
- Nilsson, Nils J. (1996). Decision Trees. Chapter 6 of *Introduction to Machine Learning*. Draft notes available at: <http://ai.stanford.edu/people/nilsson/mlbook.html>.
- Quinlan, John Ross (1989). Unknown attribute values in induction. *Proceedings of the 6th International Workshop on Machine Learning*, Ithaca, New York, United States. pp. 164-168.
- Quinlan, John Ross (1993). Arbres de décision probabilistes. *Apprentissage symbolique*, Tome II, Kodratoff, Y., Michalski, R., Carbonell, J., Mitchell, T. (Éditeurs), Cépaduès-Éditions.
- Quinlan, John Ross (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, vol. 4, 77-90.
- Utgoff, Paul E. (1989). Incremental Induction of Decision Trees. *Machine Learning*, vol. 4, 161-186.
- Wikipedia (Editor, 2006). Decision tree. *Wikipedia*. Wikipedia Foundation, Inc. (Editor). (Online encyclopedia).  
[http://en.wikipedia.org/wiki/Decision\\_tree](http://en.wikipedia.org/wiki/Decision_tree)
- Witten, Ian H.; & Eibe Frank (2<sup>nd</sup> edition, 2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.