# Un critère d'évaluation Bayésienne pour la construction d'arbres de décision

Nicolas Voisine*, Marc Boullé*, Carine Hue **

* Orange Labs, 2 avenue Pierre Marzin 22300 Lannion
nicolas.voisine@orange-ftgroup.com, marc.boulle@orange-ftgroup.com
** GFI Informatique, 11 rue Louis de Broglie 22300 Lannion, chue@gfi.fr

**Résumé.** Nous présentons dans cet article un nouvel algorithme automatique pour l'apprentissage d'arbres de décision. Nous abordons le problème selon une approche Bayésienne en proposant, sans aucun paramètre, une expression analytique de la probabilité d'un arbre connaissant les données. Nous transformons le problème de construction de l'arbre en un problème d'optimisation : nous recherchons dans l'espace des arbres de décision, l'arbre optimum au sens du critère Bayésien ainsi défini, c'est à dire l'arbre maximum a posteriori (MAP). L'optimisation est effectuée en exploitant une heuristique de pré-élagage. Des expérimentations comparatives sur trente bases de l'UCI montrent que notre méthode obtient des performances prédictives proches de celles de l'état de l'art tout en étant beaucoup moins complexes.

## 1 Introduction

La construction d'arbres de décision à partir de données est un problème qui a commencé à être traité en 1963 en construisant le premier arbre de régression pour prédire des variables numériques (Morgan et Sonquist, 1963). Suite à leurs travaux, toute une littérature a vu le jour pour décrire des modèles d'arbre soit pour des variables à prédire numériques, les arbres de régression, soit pour des variables catégorielles, les arbres de décision. On pourra se référer à l'ouvrage *« graphe d'induction »* (Zighed et Rakotomalala, 2000) pour de plus amples détails sur les différentes méthodes d'arbres de décision. Les méthodes CHAID (Kass, 1980) et ID3 (Quinlan, 1986) du début des années 80 sont des méthodes qui restent encore des références à citer. Mais ce sont les méthodes CART (Breiman et al., 1984) et la méthode C4.5 (Quinlan, 1993) dans les années 90 qui sont les références pour évaluer les performances de nouveaux algorithmes. Les premiers algorithmes d'apprentissage automatique d'arbre de décision sont basés sur un pré-élagage. Le principe de construction consiste, à partir de la racine de l'arbre, c'est-à-dire la totalité de l'ensemble d'apprentissage, à choisir parmi toutes les variables explicatives celle qui donne la meilleure partition selon un critère de segmentation. Puis de façon récursive, on applique l'algorithme de segmentation sur les feuilles. Le processus s'arrête quand pour chaque feuille on ne peut plus améliorer le critère de segmentation. Le choix de la variable de coupure et des points de coupure caractérise le processus de segmentation. La plupart des arbres (ID3, CHAID, CART, et C4.5) utilisent la théorie de l'information ou la théorie

du test statistique comme critère pour évaluer une coupure. Toute la difficulté des algorithmes de pré-élagage consiste à savoir arrêter le développement au mieux pour être suffisamment fin pour avoir de bonnes performances et pas trop détaillé pour éviter le sur-apprentissage. A partir des travaux de Breiman, de nouveaux algorithmes basés sur un post-élagage (CART, C4.5) ont été étudiés. Le principe de construction des arbres par post-élagage se fait alors en deux étapes. La première étape consiste à construire un arbre en poursuivant le processus de segmentation le plus bas possible, même s'il n'est pas pertinent. La seconde étape consiste alors à élaguer l'arbre en supprimant les branches minimisant un critère d'élagage. Le temps d'apprentissage est plus long mais les performances de l'arbre sont meilleures (cf . C4.5). Ces méthodes utilisent un critère d'élagage basé sur l'estimation du taux d'erreur de classification. Certaines méthodes utilisent un estimateur basé sur l'ensemble d'apprentissage (C4.5) et d'autres sur un ensemble de validation (CART). Ces deux approches nécessitent de définir de façon heuristique les paramètres de choix de ces critères. Une troisième approche nettement moins utilisée consiste à utiliser le principe de Minimum Description Length (Quinlan et Rivest, 1989).

Les arbres de décision sont une classe de modèles mature pour laquelle l'amélioration des performances est désormais marginale. Les performances d'un arbre dépendent principalement de la structure des arbres. Des arbres trop petits sont trop prudents et ont des performances moindres (Breiman et al., 1984). Des arbres trop grands sur-apprennent les données d'apprentissage et ont des performances qui s'effondrent sur l'ensemble de test. Par contre, le choix des variables de segmentation et la segmentation reste un problème important. Le C4.5 permet dans sa phase descendante de prendre en compte des variables non informatives qui ne sont pas remises en cause dans la phase de post-élagage. Ceci arrive d'autant plus fréquemment que le nombre de variables explicatives est important. Toute la problématique de la construction des arbres de décision est alors de savoir dans quelle branche il faut continuer le développement de l'arbre, quelles variables utiliser pour la segmentation et quand s'arrêter. Les méthodes de référence (C4.5, CART et CHAID) utilisent plusieurs paramètres pour apprendre leur arbre "optimum" : paramètres de choix des variables, de coupure de variables numériques ou de groupage des données catégorielles, et paramètre de post-élagage de l'arbre. Aucune de ces méthodes ne propose un critère global et homogène prenant en compte la structure de l'arbre, le choix des variables de coupures et les performances de l'arbre.

Wallace et Patrick à la suite des travaux de Rivest et Quinlan utilisent l'approche MDL pour définir un critère global de l'arbre prenant en compte la structure de l'arbre et la distribution des classes dans les feuilles (Wallace et Patrick, 1993). Leur algorithme consiste à élaguer l'arbre jusqu'à ce que le critère soit optimum. Ce type de méthode est peu implémentée : même Quinlan et Rivest qui en avaient donné l'idée ne l'ont pas intégré dans la méthode C4.5. Néanmoins, cette démarche reste incomplète et ne prend pas en compte le coût du choix des variables de segmentation et du modèle face à la complexité des données. Nous étudions dans cet article le problème de l'apprentissage automatique sans paramètre des arbres de décision selon une approche Bayésienne avec un critère complet. L'objectif est de transformer le problème de classification en un problème de recherche opérationnelle du meilleur arbre dans l'espace d'une famille d'arbres de décision.

L'approche MODL a montré son intérêt pour la sélection de variables, la discrétisation supervisée de variables numériques (Boullé, 2006), le groupage supervisé de variables catégorielles (Boullé, 2005) et la classification supervisée avec le modèle Selective Naive Bayes (Boullé, 2006). Notre objectif est de développer un arbre de décision utilisant l'approche

MODL, d'évaluer et de comparer ses performances avec des méthodes alternatives, plus particulièrement les méthodes d'arbre de décision J48 et SimpleCART du package WEKA (Garner, 1995), qui est une référence académique.

L'article est organisé de la façon suivante. La section 2 rappelle l'approche MODL dans le cas univarié. La section 3 décrit l'extension de cette approche aux arbres de décision. La section 4 présente l'évaluation de la méthode. Enfin, la section 5 conclut cet article.

## 2 L'approche MODL

Cette section rappelle les principes de l'approche MODL dans le cas de la discrétisation supervisée (Boullé, 2006).

La discrétisation supervisée traite des variables explicatives numériques. Elle consiste à partitionner la variable explicative en intervalles, en conservant le maximum d'information relative aux classes (valeurs de la variable catégorielle à expliquer). Un compromis doit être trouvé entre la finesse de l'information prédictive, qui permet une discrimination efficace des classes, et la fiabilité statistique, qui permet une généralisation du modèle de discrétisation.

Dans l'approche MODL, la discrétisation supervisée est formulée en un problème de sélection de modèles. Une approche Bayésienne est appliquée pour choisir le meilleur modèle de discrétisation, qui est recherché en maximisant la probabilité $p(Modle|Donnes)$ du modèle sachant les données. En utilisant la règle de Bayes, et puisque la quantité $p(Donnees)$ ne dépend pas du jeu de données, il s'agit alors de maximiser $p(Modele)p(Donnees|Modele)$, c'est-à-dire un terme d'a priori sur les modèles et un terme de vraisemblance des données connaissant le modèle.

Dans un premier temps, une famille de modèles de discrétisation est explicitement définie. Les paramètres d'une discrétisation particulière sont le nombre d'intervalles, les bornes des intervalles et les effectifs des classes par intervalle. Dans un second temps, une distribution a priori est proposée pour cette famille de modèles. Cette distribution a priori exploite la hiérarchie des paramètres : le nombre d'intervalles est d'abord choisi, puis les bornes des intervalles et enfin les effectifs par classe. Le choix est uniforme à chaque étage de cette hiérarchie. De plus, les distributions des classes par intervalle sont supposées indépendantes entre elles.

Soient $N$ le nombre d'individus, $J$ le nombre de classes, $I$ le nombre d'intervalles, $N_{i.}$. le nombre d'individus dans l'intervalle $i$ et $N_{ij}$ le nombre d'individus de la classe $j$ dans l'intervalle $i$. Dans le contexte de la classification supervisée, le nombre d'individus $N$ et de classes $J$ sont supposés connus. Un modèle de discrétisation supervisée est entièrement caractérisé par les paramètres $\{I, \{N_{i.}\}_{1 \leq i \leq I}, \{N_{ij}\}_{1 \leq i \leq I, 1 \leq i \leq J}\}$.

En utilisant la définition de la famille de modèles de discrétisation et de sa distribution a priori, la formule de Bayes permet de calculer explicitement les probabilités a posteriori des modèles connaissant les données. En prenant le log négatif de ces probabilités, cela conduit au critère d'évaluation fourni dans la formule (1).

$$\underbrace{\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_{i.}+J+1}{J-1}}_{-\log(p(Modele))} + \underbrace{\sum_{i=1}^{I} \frac{N_{i.}!}{N_{i12}!N_{i2}!...N_{iJ}!}}_{-\log(p(Donnees|Modele))} \quad (1)$$

Les trois premiers termes représentent la probabilité a priori du modèle : choix du nombre d'intervalles, des bornes des intervalles, et de la distribution multinomiale des classes dans chaque intervalle. Le dernier terme représente la vraisemblance, c'est à dire la probabilité d'observer les classes connaissant le modèle de discrétisation.

La discrétisation optimale est recherchée en optimisant le critère d'évaluation, au moyen de l'heuristique gloutonne ascendante décrite dans (Boullé, 2006). A l'issue de cet algorithme d'optimisation, des post-optimisations sont effectuées au voisinage de la meilleure solution, en évaluant des combinaisons de coupures et de fusions d'intervalles. L'algorithme exploite la décomposabilité du critère sur les intervalles pour permettre après optimisations de se ramener à une complexité algorithmique en temps $\mathcal{O}(JN \log N)$.

# 3 Arbre de décision MODL

Dans cette section, nous appliquons l'approche MODL aux arbres de décision en explicitant la famille de modèles envisagée et en présentant un critère d'évaluation global des arbres résultant d'une approche Bayésienne de la sélection de modèles.

## 3.1 Définition

Un arbre de décision consiste à prédire une variable à expliquer catégorielle à partir de variables explicatives numériques ou catégorielles. Le problème de l'apprentissage consiste à trouver la structure d'arbre qui a les meilleures performances tout en gardant la plus petite taille possible. Toute la difficulté consiste à trouver un compromis entre performance et structure de l'arbre permettant une bonne généralisation du modèle.

L'approche MODL pour les arbres de décision consiste à trouver dans une famille d'arbres de décision celui qui maximise la probabilité du modèle connaissant les données. Comme pour la discrétisation (cf. section 2) on applique une approche Bayésienne pour choisir l'arbre de décision qui maximise la probabilité $p(Arbre|Donnees)$, ce qui revient à maximiser :

$$p(Arbre)p(Donnees|Arbre)$$

Où $p(Arbre)$ est un terme d'a priori de l'arbre de décision et $p(Donnees|Arbre)$ un terme de vraisemblance des données connaissant le modèle.

Par la suite nous utiliserons les notation suivantes :

- $T$ un modèle d'arbre de décision ;
- $\mathbb{K}$ : l'ensemble des $K$ variables explicatives ;
- $\mathbb{K}_T$ : l'ensemble des $K_T$ variables explicatives utilisées par l'arbre $T$ ;
- $\mathbb{S}_T$ : l'ensemble des nœuds internes de l'arbre $T$ ;
- $\mathbb{L}_T$ : l'ensemble des feuilles de l'arbre $T$ ;
- $N_{s.}$ : le nombre d'individus dans le nœud $s$ ;
- $X_s$ : la variable de segmentation du nœud $s$ ;

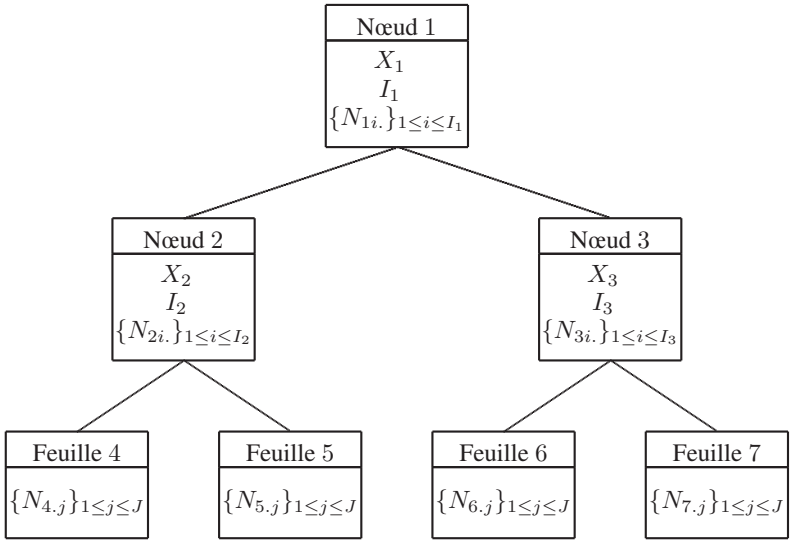


FIG. 1 – *Exemple d'arbre de décision. Les nœuds internes représentent les règles et les feuilles représentent la distribution des classes*

- $V_{X_s}$ : nombre de valeurs dans le nœud $s$, dans le cas d'une variable $X_s$ catégorielle ;
- $I_s$ : nombre de fils du nœud $s$ ;
- $N_{si.}$ : le nombre d'individus dans le $i$ème fils du nœud $s$ ;
- $N_{l.j}$ : nombre d'individus de classe $j$ dans la feuille $l$.

Le modèle d'arbre de décision MODL est représenté par sa structure, la répartition des individus dans cette structure et la distribution des classes dans les feuilles (cf. figure 1). La structure du modèle d'arbre est représentée par l'ensemble des nœuds internes $\mathbb{S}_T$ (nœuds ayant au moins deux fils), l'ensemble des feuilles (nœuds n'ayant pas de fils) et les liens entre les nœuds.

La répartition des individus dans cette structure est définie par les coupures dans les nœuds internes ainsi que les effectifs des classes par feuille. L'ensemble des paramètres de l'arbre est ainsi défini par :

- l'ensemble des variables $\mathbb{K}_T$ utilisées pour le modèle $T$, c'est-à dire le nombre de variables $K_T$ et le choix des $K_T$ variables prises parmi $K$ ;
- la nature des nœuds ;
- la répartition des individus dans les nœuds internes $s$ :
  - la variable de segmentation $X_s$, le nombre d'intervalles ou de groupes $I_s$ ;

  - la distribution des exemples $N_{si.}$ sur ces $I_s$ intervalles ou groupes ;

- la répartition $N_{l.j}$ des classes dans les feuilles.

## 3.2 Critère d'évaluation

Le critère d'évaluation que nous proposons est le log négatif de la probabilité a posteriori de l'arbre connaissant les données. La probabilité des données étant constante quel que soit le modèle, le critère est défini par :

$$c(Arbre) = -\log p(Arbre)p(Donnees|Arbre)$$

Nous choisissons la probabilité a priori du modèle $p(Arbre)$ en exploitant une hiérarchie parmi les paramètres de modélisation. Cette hiérarchie a pour objectif de définir les relations de dépendance entre les paramètres. Le choix des a priori s'inspire des extensions hiérarchiques de l'approche Bayesienne. Dans le cas d'un paramétrage complexe, on exprime l'incertitude sur les paramètres de plus haut niveau, puis, conditionnellement, l'incertitude sur les paramètres de plus bas niveau. La loi de Bayes nous permet alors de formuler $p(Arbre)$ selon un principe de parcimonie proche de l'approche Minimum Description Length et selon des a priori de distribution de ces paramètres.

Il existe plusieurs voies pour définir la hiérarchie des paramètres. L'une consisterait à définir la structure, puis les coupures, puis la répartition des classes dans les feuilles. Dans cet article, nous proposons d'exploiter la hiérarchie implicite de l'arbre en définissant le modèle au niveau de la racine indépendamment de ses fils. Puis de façon récursive, on continue à définir les fils de la racine jusqu'aux feuilles de l'arbre. On peut ainsi définir la probabilité d'un arbre de décision MODL par :

$$\begin{aligned} p(Arbre) =& p(\mathbb{K}_T)\times \\ & \prod_{s\in\mathbb{S}_T} p(s)p(X_s|\mathbb{K}_T)p(I_s|\mathbb{K}_T, X_s, N_{s.})p(N_{si.}|\mathbb{K}_T, X_s, N_{s.}, I_s)\times \\ & \prod_{l\in\mathbb{L}_T} p(l)p(N_{l.j}|\mathbb{K}_T, N_{l.}) \end{aligned} \quad (2)$$

On choisit de sélectionner $K_T$ variables selon un a priori uniforme allant de 0 à $K$, 0 s'il n'y a pas de variable informative, ce qui donne $K+1$ choix de sélection de variables. Prenant comme hypothèse que le choix du nombre de variables est uniforme, on en déduit la probabilité a priori pour le nombre de variables sélectionnées. On prend comme hypothèse que chaque groupe de variables sélectionnées est équiprobable et que le nombre de sélections possibles est égal au nombre de combinaisons avec remise de $K_T$ variables parmi $K$. Ce qui donne :

$$P(\mathbb{K}_T) = \frac{1}{K+1}\frac{1}{\binom{K+K_T-1}{K_T}}$$

Connaissant les variables utilisées on peut définir la nature de chaque nœud de l'arbre de décision, soit interne soit feuille. En prenant comme hypothèse l'uniformité des états, $p(s)$ et $p(l)$ sont égaux et valent $0.5$.

On considère que pour chaque nœud interne le choix de la variable de segmentation est indépendant et équiprobable sur les $K_T$ variables explicatives sélectionnées. Connaissant la nature de la variable de sélection (numérique ou catégorielle) ainsi que le nombre de parties, on peut définir la probabilité a priori du nœud interne. Pour une variable numérique, de façon analogue au cas de la discrétisation univariée MODL, on obtient :

$$\frac{1}{2}\frac{1}{K_T}\frac{1}{N_s}\frac{1}{\binom{N_s+I_s-1}{I_s-1}}$$

Pour une variable catégorielle, de façon analogue au groupement de valeurs univarié MODL, on définit la probabilité du nœud interne par :

$$\frac{1}{2}\frac{1}{K_T}\frac{1}{V_{X_s}}\frac{1}{B(V_{X_s}, I_s)}$$

Pour finir, il nous faut définir la probabilité a priori d'une feuille, c'est-à-dire la probabilité de la distribution des classes dans la feuille. Considérant les distributions équiprobables, cela revient à calculer le nombre de distributions multinomiales de $N_{l.}$ individus en $J$ classes.

$$\frac{1}{2}\frac{1}{\binom{N_{l.}+J-1}{J-1}}$$

Il nous faut maintenant expliciter la probabilité d'observer les données connaissant le modèle. La probabilité des données dépend uniquement des feuilles de l'arbre. Connaissant le modèle de distribution multinomiale défini sur une feuille, on en déduit la probabilité d'observation :

$$p(Donnees|Arbre) = \prod_{l \in L} \frac{N_{l.}!}{N_{l.1}!N_{l.2}!...N_{l.J}!}$$

### 3.3 Amélioration du critère

Pour éviter que l'arbre soit trop prudent il faut augmenter la probabilité a priori du modèle $p(T)$. La probabilité du nombre de coupures d'un nœud interne $P(I_s|\mathbb{K}_T, X_s, N_{s.})$, qui vaut $\frac{1}{N_s}$ dans la cas numérique, peut être très faible quand le nombre d'individus (ou de valeurs de variables) est élevé. Plus le nombre d'individus est important, plus la probabilité de choisir un nombre de parties $I_s$ est faible. Cela peut rendre le modèle trop prudent et réduire ses performances de prédiction. L'amélioration que nous proposons s'appuie sur l'approche MDL. Rissanen propose un codage optimal des entiers naturels positifs non bornés et en donne la probabilité (Rissanen, 1978). La taille optimale en bits d'un entier positif non borné $I_s$ vaut :

$$c_{ris}(I_s) = log_2(2.865) + log_2(I_s) + log_2(log_2(I_s)) + ...$$

On peut alors selon l'approche MDL écrire la probabilité d'avoir $I_s$ coupures par :

$$2^{-c_{ris}(I_s)}$$

De plus, en exploitant le fait qu'un nœud interne est nécessairement partitionné en au moins deux fils, on peut éviter de modéliser explicitement la nature du nœud (interne ou feuille). On

se limite alors à modéliser le nombre de fils de chaque nœud, soit un fils pour une feuille, soit entre 2 et $N_s$ fils pour un nœud interne. Le coût optimisé de l'arbre est alors égal à :

$$C_{opt}(T) = \log(K+1) + \log\binom{K+K_T-1}{K_T} + \tag{3}$$

$$+ \sum_{s\in\mathbb{S}_{T_n}} \log K_T + c_{ris}(I_s)\log 2 + \log\binom{N_{s.}+I_s-1}{I_s-1} + \tag{4}$$

$$+ \sum_{s\in\mathbb{S}_{T_c}} \log K_T + c_{ris}(V_{X_s})\log 2 + \log B(V_{X_s}, I_s) + \tag{5}$$

$$+ \sum_{l\in\mathbb{L}_T} C_{ris}(1)\log 2 + \log\binom{N_{l.}+J-1}{J-1} + \tag{6}$$

$$+ \sum_{l\in\mathbb{L}_T} \log\frac{N_{l.}}{N_{l.1}N_{l.2}...N_{l.J}} \tag{7}$$

où $\mathbb{S}_{T_n}$ et $\mathbb{S}_{T_c}$ sont les ensembles des nœuds internes utilisant une variable de segmentation soit numérique soit catégorielle.

Les quatre premiers termes d'a priori correspondent au choix de la structure du modèle et le dernier terme correspond à son aptitude à épouser les données. Le terme 3 correspond au choix de l'ensemble des variables de segmentation. Les deux termes suivants 4 et 5 correspondent au choix des partitions dans les nœuds internes pour les variables numériques et catégorielles. Le quatrième terme d'a priori représente le choix de la distribution multinomiale des classes dans chaque feuille. Le dernier terme représente la vraisemblance, c'est-à-dire la probabilité d'observer les classes dans les feuilles connaissant le modèle de l'arbre de décision supervisé.

# 4 Construction de l'arbre optimum

La recherche de l'optimum global du critère a une complexité maximale en temps exponentielle par rapport au nombre d'individus. Il est donc impossible d'utiliser un algorithme exhaustif pour trouver l'optimum. Dans notre article nous étendons une heuristique classique de construction d'arbre basée sur un pré-élagage.

L'algorithme 1 consiste à rechercher pour chaque feuille de l'arbre la meilleure partition suivant les variables explicatives, puis de recommencer jusqu'à ce qu'il n'y ait plus de feuille à partitionner. Pour une feuille $l$ on construit les partitionnements $P_X(l)$ pour chaque variable $X$ selon l'approche MODL univariée (discrétisation ou groupage), puis on évalue l'ajout de ces partitions dans la structure de l'arbre avec le coût global $C_{opt}(T)$. On recherche donc l'optimum de l'arbre par une série de recherche d'optimum locaux au niveau des feuilles. Ce type d'algorithme est proche de ceux utilisés pour optimiser les arbres de décision (ID3 et CHAID). Sa différence réside dans le fait que l'accroissement dans deux feuilles de l'arbre n'est pas indépendante mais se fait selon le critère global. Une feuille de l'arbre peut ainsi ne jamais se développer parce qu'elle n'est jamais le meilleur choix. Il est à noter qu'entre deux itérations, seuls les fils du nœud développé sont à réévaluer. L'algorithme ne garantit pas d'arriver à l'optimum global mais il a une complexité maximal en $\mathcal{O}(KJN^2Log(N))$, dans le

**Algorithm 1** Algorithme descendant d'optimisation d'un arbre

**ENTRÉES:** $T$ la racine de l'arbre
**SORTIES:** l'arbre $\hat{T}$ qui optimise le critère
  $T^* \leftarrow T$
  **tantque** amélioration **faire**
    $\hat{T} \leftarrow T^*$
    **pour toute** feuille l de l'arbre **faire**
      $T' \leftarrow T^*$
      **pour toute** variable $X$ de $\mathbb{K}$ **faire**
        Recherche de la règle sur la feuille $l$ suivant $X$ qui améliore au mieux le critère
        $T_X \leftarrow T^* + P_X(l)$
        **si** $c(T_X) < c(T')$ **alors**
          $T' \leftarrow T_X$
        **finsi**
      **fin pour**
      **si** $c(T') < c(T^*)$ **alors**
        $T^* \leftarrow T'$
      **finsi**
    **fin pour**
  **fin tantque**

cas d'un arbre déséquilibré. Cette complexité se réduit à $\mathcal{O}(KJNLog(N))$ dans le cas d'un arbre équilibré. Cet algorithme est déterministe, il trouve à chaque fois le même optimum.

# 5 Expérimentation

Cette section présente des résultats d'expérimentation permettant d'évaluer notre méthode de construction d'arbres de décision supervisés.

## 5.1 Protocole

Les expérimentations sont menées en utilisant 30 jeux de données de l'UCI (Blake et Merz, 1996) décrits en table 2, représentant une grande diversité de domaines, de nombres d'individus, de variables explicatives (numériques et/ou catégorielles) et de nombres de classes. Afin d'évaluer notre critère de qualité des arbres, nous avons testé deux variantes de l'algorithme 1. La première consiste à n'avoir aucune contrainte sur l'arbre Ktree et la deuxième consiste à imposer à l'arbre une structure binaire Ktree(2), en se limitant à deux parties pour chaque nœud interne. Nous avons comparé notre approche avec J48 et SimpleCART qui sont des implémentations de C4.5 et CART dans l'environnement d'analyse de données WEKA (Garner, 1995). Nous avons pris comme paramètres ceux définis par défaut dans l'application. Nous avons évalué le taux de bonne prédiction en test, l'AUC en test, le nombre de nœuds internes ainsi que le temps de calcul. Les critères d'évaluation sont évalués au moyen d'une validation croisée stratifiée à 10 niveaux.

### 5.2 Résultats

Les résultats d'évaluation sont résumés de façon synthétique dans le tableau 1, en reportant pour chaque méthode la moyenne géométrique des critères d'évaluation sur les 30 jeux de données de l'UCI. Au regard de la grande dispersion des résultats selon les domaines d'application, nous préférons étayer notre analyse sur la moyenne géométrique. Elle permet de comparer les ratios entre les différentes méthodes. La moyenne arithmétique est tout de même affichée dans la figure 2.

| Méthode | Acc. Test | AUC Test | Arbre Size | Temps | $C_{opt}(T)$ |
|---|---|---|---|---|---|
| Ktree(2) | 0.819 | 0.889 | 17.5 | 0.5 | 524.6 |
| Ktree | 0.813 | 0.884 | 19.4 | 0.5 | 565.3 |
| sCART | 0.822 | 0.876 | 30.7 | 1.0 | $\times$ |
| J48 | 0.834 | 0.881 | 77.1 | 0.1 | $\times$ |

TAB. 1 – *Moyennes géométriques des résultats expérimentaux sur les bases de l'UCI : taux de bonne prédiction et AUC en test, nombre de nœuds, temps de calcul lors de l'apprentissage et coût de l'arbre KTree*

On constate globalement que le taux de bonne prédiction est à l'avantage de J48 et que l'AUC est légèrement à l'avantage de Ktree. Ces faibles différences ne sont pas surprenantes : les arbres de décision sont une technologie mature et les différences de performance sont souvent marginales. En revanche, la complexité de la structure des arbres est environ quatre fois moindre avec Ktree qu'avec J48 et deux fois moindre qu'avec SimpleCART. Cette propriété rend l'interprétation de l'arbre nettement plus aisée pour l'expert, et son déploiement sur des bases test plus rapide. Au niveau du temps de calcul, Ktree est en moyenne cinq fois plus lent que J48 et deux fois plus rapide que SimpleCART. En ce qui concerne les différences entre Ktree et Ktree(2), l'avantage est à l'arbre binaire qui obtient de meilleurs taux de bonne prédiction et AUC tout en restant faiblement complexe. On constate également que le critère est lui aussi plus faible, ce qui montre que la performance des arbres KTree est clairement corrélée avec la valeur du critère d'évaluation.

En analysant les résultats par base détaillés dans la table 2, on s'aperçoit que les performances pour Ktree sont moins bonnes sur les domaines avec des variables explicatives corrélées tel que Tictactoe, Letter et les bases de segmentation d'images. Par contre pour des base marketing tels que Adult, Ktree est légèrement meilleur tout en étant dix fois moins complexe que J48.

## 6 Conclusion

Le critère Bayésien présenté dans cet article permet d'évaluer un arbre de décision en prenant en compte la structure de l'arbre, le choix des variables explicatives, des coupures ainsi que les distributions des classes dans chaque feuille. Ce critère complet est sans aucun paramètre. La méthode de construction d'arbre décrite dans cet article se base sur une heuristique de pré-élagage en sélectionnant et partitionnant (en intervalles ou groupes de valeurs) chaque variable explicative.

| Domaine | Information données | | | | Acc Test | | | | Tree Size | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Var.** | **Inst.** | **Val.** | **Maj.** | **Ktree(2)** | **Ktree** | **sCart** | **J48** | **Ktree(2)** | **Ktree** | **sCart** | **J48** |
| Yeast | 9 | 1484 | 10 | 0.31 | 0.569 | 0.544 | 0.309 | 0.503 | 17.8 | 23.8 | 1.4 | 96.6 |
| Wine | 13 | 178 | 3 | 0.40 | 0.928 | 0.922 | 0.894 | 0.939 | 8.8 | 7.8 | 9.2 | 9.8 |
| WaveformNoise | 40 | 5000 | 3 | 0.34 | 0.743 | 0.744 | 0.767 | 0.751 | 62.6 | 73.3 | 121.4 | 580.4 |
| Waveform | 21 | 5000 | 3 | 0.34 | 0.749 | 0.747 | 0.777 | 0.759 | 76.2 | 95.2 | 136.6 | 541.8 |
| Vehicle | 18 | 846 | 4 | 0.26 | 0.677 | 0.651 | 0.701 | 0.726 | 25 | 26.3 | 104.8 | 136 |
| TicTacToe | 9 | 958 | 2 | 0.65 | 0.815 | 0.729 | 0.932 | 0.851 | 22.2 | 12 | 67.2 | 135.4 |
| Thyroid | 21 | 7200 | 3 | 0.93 | 0.995 | 0.994 | 0.996 | 0.997 | 14.4 | 24.2 | 22.4 | 30.6 |
| Spam | 57 | 4307 | 2 | 0.65 | 0.916 | 0.915 | 0.922 | 0.935 | 46.4 | 53.5 | 131.2 | 192.2 |
| Sonar | 60 | 208 | 2 | 0.53 | 0.715 | 0.701 | 0.712 | 0.712 | 4.8 | 4.9 | 14.2 | 29.2 |
| SickEuthyroid | 25 | 3163 | 2 | 0.91 | 0.978 | 0.977 | 0.977 | 0.979 | 11 | 13.3 | 14 | 26.2 |
| Segmentation | 19 | 2310 | 7 | 0.14 | 0.936 | 0.937 | 0.958 | 0.971 | 32 | 46.8 | 76.6 | 82.6 |
| Satimage | 36 | 6435 | 6 | 0.24 | 0.852 | 0.854 | 0.868 | 0.873 | 73.8 | 73.9 | 165.2 | 551.4 |
| Pima | 8 | 768 | 2 | 0.65 | 0.741 | 0.751 | 0.751 | 0.738 | 9 | 8.1 | 16.2 | 37.4 |
| PenDigits | 16 | 10992 | 10 | 0.10 | 0.944 | 0.907 | 0.963 | 0.966 | 169 | 259 | 363.4 | 375.6 |
| Mushroom | 22 | 8416 | 2 | 0.53 | 1.000 | 1.000 | 1.000 | 1.000 | 12.2 | 14 | 13.4 | 29.8 |
| Letter | 16 | 20000 | 26 | 0.04 | 0.766 | 0.741 | 0.869 | 0.879 | 464.4 | 554.4 | 2091.2 | 2321.6 |
| LED17 | 24 | 10000 | 10 | 0.11 | 0.737 | 0.737 | 0.735 | 0.722 | 77 | 77 | 123.8 | 890 |
| LED | 7 | 1000 | 10 | 0.11 | 0.710 | 0.710 | 0.725 | 0.729 | 29.4 | 29.4 | 110 | 62.2 |
| Iris | 4 | 150 | 3 | 0.33 | 0.920 | 0.920 | 0.953 | 0.960 | 5 | 4.2 | 8 | 8.4 |
| Ionosphere | 34 | 351 | 2 | 0.64 | 0.900 | 0.889 | 0.898 | 0.915 | 6.2 | 8.1 | 8.8 | 27.4 |
| Hypothyroid | 25 | 3163 | 2 | 0.95 | 0.992 | 0.992 | 0.992 | 0.992 | 6 | 10.4 | 10.8 | 11.8 |
| HorseColic | 27 | 368 | 2 | 0.63 | 0.862 | 0.862 | 0.875 | 0.878 | 5 | 5 | 10 | 19.6 |
| Hepatitis | 19 | 155 | 2 | 0.79 | 0.806 | 0.806 | 0.786 | 0.838 | 3 | 3 | 9.6 | 17.8 |
| Heart | 13 | 270 | 2 | 0.56 | 0.767 | 0.756 | 0.785 | 0.767 | 8 | 8.2 | 14.2 | 33.8 |
| Glass | 9 | 214 | 6 | 0.36 | 0.607 | 0.654 | 0.705 | 0.659 | 8.4 | 8.4 | 20 | 47 |
| German | 24 | 1000 | 2 | 0.70 | 0.692 | 0.692 | 0.750 | 0.739 | 3.2 | 3.2 | 19.4 | 140.6 |
| Crx | 15 | 690 | 2 | 0.56 | 0.861 | 0.861 | 0.852 | 0.861 | 7 | 7 | 3.6 | 27.1 |
| Breast | 10 | 699 | 2 | 0.66 | 0.936 | 0.957 | 0.949 | 0.946 | 9.6 | 9.1 | 15.8 | 23.4 |
| Australian | 14 | 690 | 2 | 0.56 | 0.852 | 0.852 | 0.857 | 0.852 | 6.6 | 6.9 | 5.8 | 46.2 |
| Adult | 15 | 48842 | 2 | 0.76 | 0.863 | 0.862 | 0.863 | 0.860 | 115 | 177 | 120.2 | 1099 |
| **Moy. Géo.** | 17.6 | 1503.5 | 3.3 | 0.4 | 0.819 | 0.813 | 0.822 | 0.834 | 17.5 | 19.4 | 30.7 | 77.1 |
| **Moy. Ar.** | 21.0 | 4828.6 | 4.5 | 0.5 | 0.828 | 0.822 | 0.837 | 0.843 | 44.6 | 54.9 | 127.6 | 254.4 |

TAB. 2 – *Résultats expérimentaux sur les bases de l'UCI : taux de bonne prédiction et AUC en test, nombre de nœuds.*

Des évaluations sur 30 jeux de données de l'UCI démontrent que le critère permet de créer des arbres de décision équivalents à l'état de l'art en performance mais beaucoup moins complexe en nombre de nœuds générés. On constate aussi que l'algorithme générant des arbres binaires est meilleur en moyenne que celui générant des arbres n-aires. Ceci laisse à penser que l'optimisation de l'algorithme peut être une voie d'amélioration des performances non négligeable. Par exemple nous envisageons d'utiliser une heuristique de post-élagage, en forçant le développement de l'arbre, puis élaguant l'arbre obtenu en se basant toujours sur notre critère d'évaluation globale de l'arbre.

# Références

Blake, C. et C. Merz (1996). UCI repository of machine learning databases. http ://www.ics.uci.edu/mlearn/MLRepository.html.

Boullé, M. (2006). a Bayes optimal discretization method for continuous attributes. *Machine Learning 65*, 131–165.

Boullé, M. (2005). A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research 6*, 1431–1452.

Boullé, M. (2006). An enhanced selective naive Bayes method with optimal discretization. In I. Guyon, S. Gunn, M. Nikravesh, et L. Zadeh (Eds.), *Feature Extraction: Foundations And Applications*, Chapter 25, pp. 499–507. Springer.

Breiman, L., J. H. Friedman, R. A. Olshen, et C. J. Stone (1984). *Classification And Regression Trees*. New York: Chapman and Hall.

Garner, S. R. (1995). Weka: The waikato environment for knowledge analysis. In *In Proc. of the New Zealand Computer Science Research Students Conference*, pp. 57–64.

Kass, G. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics 29*(2), 119–127.

Morgan, J. et J. A. Sonquist (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association 58*, 415–435.

Quinlan, J. et R. Rivest (1989). Inferring decision trees using the minimum description length principle. *Inf. Comput. 80*(3), 227–248.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*, 81–106.

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

Rissanen, J. (1978). Modeling by shortest data description. *Automatica 14*, 465–471.

Wallace, C. et J. Patrick (1993). Coding decision trees. *Machine Learning 11*, 7–22.

Zighed, D. et R. Rakotomalala (2000). *Graphes d'induction*. France: Hermes.

## Summary

In this paper, we present a new automatic training algorithm for decision trees. We exploit a parameter-free Bayesian approach and propose an analytic formula for the evaluation of the probability of a decision tree given the data. We thus transform the training problem into an optimisation problem in the space of decision tree models, and search for the best tree, which is the maximum a posteriori (MAP) one. The optimisation is performed using a top-down heuristic. Extensive experiments on 30 UCI databases show that our method obtains predictive performance similar to that of alternative state-of-the-art methods, with far more simple trees.