Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Abou Bekr Belkaid
Faculté de Science
Département Informatique

MÉMOIRE DE FIN D'ÉTUDE

pour obtenir le grade de

Master en Informatique

Spécialité : **MID**

présenté et soutenu publiquement
par

Melle Asma HAMMYANI
Melle Soumia ALLIOUA

le 02 Juillet 2013

Titre:

Amélioration des forêts aléatoires : Application au diagnostic médical

Jury

Président du jury.	Mr Mortada BENAZZOUZ,	UABB Tlemcen
Examinateur.	Mr. Fethallah HADJILA,	MCA UABB Tlemcen
Examinateur.	Mr Amine BELABED,	MCA UABB Tlemcen
Directeur de mémoire.	Pr. Chikh Mohamed Amine,	UABB Tlemcen
Co-Directeur de mémoire.	Mr Mostafa EL HABIB DAHO,	UABB Tlemcen

Je dédie ce modeste travail à
Mes très chers parents qui m'ont éclairés mon chemin et qui m'ont encouragés et soutenue toute au long de mes études,
que dieu les récompense et les garde,
Ma sœur et mes frères,
Tous mes amis,
Mes collègues de promotion,
A tous ceux qui me connaissent de près ou de loin,
Et a tous ceux qui occupent une place dans mon cœur,
Merci d'être toujours la pour moi.

HAMMYANI Asma

Je dédie ce modeste travail à Mes très chers parents, en témoignage de leur amour,et dont le soutien de tous les instants m'a permis de mener à bien ce travail.J'espère que Dieu tout puissant me donne la force et le courage pour que je puisse rendre leurs sacrifices,
Mes frères et mes sœurs ;
Mes grands parents ;
Mes oncles et mes tantes ;
Mes collègues de promotion ;
Mes enseignants et à tout ceux qui me sont chers.

ALLIOUA Soumia

Remerciements

Qu'ils trouvent ici l'expression de toute ma reconnaissance.
Nous remercions tout d'abord Dieu pour l'accomplissement de ce mémoire.

En préambule à ce mémoire, nous souhaitions adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons dans un premier temps à remercier sincèrement Mr CHIKH M.A, Professeur à l'université de Tlemcen, en tant que directeur de mémoire de l'attention qu'il a porté sur notre travail et ces judicieux conseils.

Nos remerciements s'adressent également à Monsieur EL HABIB DAHO M, en tant que Co-encadreur, pour ses conseils, son aide, et son suivi durant la réalisation de ce travail.

Nous remercions également Melle Settouti Nesma pour sa disponibilité, ses conseils, ses suggestions, et son encadrement.

Nous exprimons notre haute considération aux membres de jury d'avoir accepté d'examiner notre travail :
Mr Benazzouz M qui nous a fait l'honneur de présider notre jury.
Mr Hadjila F et Mr Belabed A d'avoir bien voulu examiner ce travail.

Nous remercions infiniment tous les enseignants qui nous ont aidé durant tout notre cycle d'études.

Résumé

Forêts Aléatoires (RF) est une technique de prévision d'ensemble réussie qui utilise le vote majoritaire ou une moyenne en fonction de la combinaison. Cependant, il est clair que chaque arbre dans une forêt aléatoire peut avoir une contribution différente au traitement d'une certaine instance ;

Dans ce projet, nous démontrons que les performances de prédiction des RF's peuvent encore être améliorées par le remplacement de l'indice de GINI par un autre indice (twoing ou deviance). Nos expériences démontrent également que le vote pondéré donne de meilleurs résultats par rapport au vote majoritaire .

Mots clés

Forêts Aléatoires ;arbre de décision ;critère de segmentation,vote majoritaire ;vote pondéré ;UCI Machine Learning.

Abstract

Random Forests (RF) are a successful ensemble prediction technics that used majority voting or averaging as a combination function. However, it is clear that each tree in random forests may have a different contribution in processing a certain instance.

In this project, we show that the prediction performances of RF are improved by replacing GINI index with another index (twoing or deviance). Our experiments also demonstrate that the weighted voting gives better results than majority voting.

Keywords

Random forests, decision tree ;segmentation criterion ;majority voting ;weighted voting ;UCI Machine Learning Database.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Le data mining ou fouille de données est la recherche d'informations pertinentes pour l'aide à la décision et la prévision. Elle met en œuvre des techniques statistiques et d'apprentissage artificiel en tenant compte de la spécificité de grands ensembles de données.
L'apprentissage automatique de la machine consiste à concevoir des systèmes de classification performants à partir d'un ensemble d'exemples représentatifs d'une population de données.
Parmi les types de l'apprentissage automatique on trouve l'apprentissage supervisé pour produire automatiquement des règles à partir d'une base de données d'apprentissage étiquetées. Cette technique a pour But de prédire la classe de nouvelles données observées, utilisant des modèles de classifications (classifieurs) comme les Arbres de décision, réseaux bayésiens, réseaux de neurones, k-plus proches voisins, etc...

Plusieurs travaux prouvent qu'un modèle de classification induisant une seul hypothèse possède des problèmes. Ils ont proposé de combiner chacun de ses classifieurs individuels faibles pour former un unique système de classification appelé Ensemble de Classifieurs [Dietterich,2000].
Les Ensembles de Classifieurs (EoC) est une des approches multi-classifieurs les plus populaires et les plus efficaces qui consiste à combiner un ensemble de classifieurs de même type (un ensemble de réseaux de neurones, un ensemble d'arbres de décision, ou un ensemble de discriminants). Il existe aujourd'hui un grand nombre de méthodes capables de générer automatiquement des ensembles de classifieurs : Bagging, Boosting, Random Subspaces..., et chacune de ces procédures d'induction d'EoC dispose d'hyper-paramètres pour le contrôle de la construction des EoC.

Parmi les méthodes d'induction d'EoC, on trouve la méthode des forêts aléatoires [Breiman,2001]. Cette méthode est un bagging amélioré au niveau des hyper-paramètres. Elle est basées sur la combinaison de classifieurs élémentaires de types arbres de décision. Individuellement, ces classifieurs ne sont pas efficaces, mais ils possèdent des propriétés intéressantes à exploiter au sein d'un EoC : ils sont particulièrement instables La spécificité des arbres utilisés dans les forêts aléatoires est que leur induction est perturbée d'un facteur aléatoire, et ce dans le but de générer de la diversité dans l'ensemble. C'est sur la base de ces deux éléments : utiliser des arbres de décision comme classifieurs élémentaires et faire intervenir l'aléatoire dans leur induction qu'a été introduit le formalisme des forêts aléatoires.

Problématique Breiman a proposé un algorithme d'induction des forets aléatoire appelé Forest-RI [réf Breiman 2001], cet algorithme est considéré comme algorithme

de référence qui est compétitif avec les principales méthodes d'ensembles. Cependant, son utilisation pose toujours aujourd'hui d'importantes difficultés. Les classifieurs de la foret aléatoire sont des arbres de décisions de CART, et l'agrégation de ces classifieurs définit par un vote ordinaire.

Malgré le succès des méthodes de foret aléatoire, plusieurs travaux ont proposé de l'améliorer. Individuellement, chaque classifieurs donne des faibles prédictions. Ils ont proposé le renforcement de chaque classifieurs sans sacrifier la variété entre eux et de diminuer la variance sans sacrifier la force.

Pour d'autres travaux l'amélioration de chaque classifieur individuel est insuffisante, ils ont proposé de changer la méthode d'agrégation classique par d'autres techniques.

La difficulté de l'utilisation de l'algorithme de foret aléatoire nous a conduits à proposer d'autres améliorations de cette méthode de classification.

Ce travail de fin d'étude se situe dans le contexte général de l'apprentissage statistique, utilisant les forêts aléatoires. Cette méthode a pour but de classifier des gènes biologiques à partir des nouvelles données observées, il est composé de trois chapitres :

Chapitre1 présente globalement les méthodes d'ensemble, il permet en particulier de détailler la méthode utilisée dans notre mémoire.

Chapitre 2 concerne l'Etat de l'art, il présente les techniques utilisées dans la litérature pour améliorer ces méthodes.

chapitre3 concerne l'Expérimentation, il présente l'approche proposée, les bases de données utilisée, les expérimentations réalisées, les résultats obtenus avec leurs interprétations et une étude comparative avec d'autres résultats de certains travaux portant sur le même sujet.

En dernier lieu, une conclusion générale et des perspectives pour ce travail de master sont présentées.

Chapitre 1

Les méthodes d'ensembles

introduction

L'idée de combiner plusieurs classifieurs pour former un comité est assez ancienne. Les premiers travaux allant dans ce sens datant du milieu des années 60 [Nil65].De nombreuses approches ont été proposées pour construire des ensembles composés de classifieurs complémentaires [KR00, KR01, RK02, WR03, RKW04, OPKR05, HKR07, BKR09].L'efficacité d'un ensemble de classifieurs repose sur la combinaison de classifieurs complémentaires ou divers. Chaque classifieurs doit être relativement bon et différent des autres classifieurs, pour donner un bon prédicteur et améliorer les prédictions. On trouve plusieurs méthodes de cette combinaison, dans ce chapitre nous présentons les forets aléatoires de Leo Breiman [Bre01] qui se caractérise par l'utilisation d'une combinaison d'arbres de décision.

1 Les méthodes d'ensembles

Les méthodes d'ensembles sont des méthodes permettent de construire une collection de prédicteurs et agréger l'ensemble de leurs prédictions. Dans un cadre de classification, l'agrégation revient par exemple à faire un vote majoritaire parmi les classes fournies par les classifieurs.
L'objectif de ces méthodes est de trouver un classifieur final. Ce classifieur soit meilleur que chacun des classifieurs individuels [Lec07].Pour qu'une méthode d'ensemble soit performante, elle doit réussir à construire une collection de prédicteurs qui vérifie ces deux points :

- Chaque prédicteur individuel doit être relativement bon.
- Les prédicteurs individuels doivent être différents les uns des autres

L'utilisation des méthodes d'ensemble garantie :

- La précision : de meilleurs classifieurs peuvent être obtenus en combinant les prédictions de plusieurs classifieurs (même faiblement efficaces)
- L'Efficacité : un problème complexe peut être décomposé en de multiples sous problèmes plus simple à résoudre (approche diviser pour régner)[Gen10].

2 Les types des méthodes d'ensemble

Il existe deux types des méthodes ensemble sont :

2.1 Les méthodes d'ensemble hétérogènes :

sont des méthodes qui fonctionnent d'une manière séquentielle, sont principe est d'utiliser un ensemble d'apprentissage et le traiter par des différents modèles de classification.

2.2 Les méthodes d'ensembles homogènes :

Son principe de fonctionnement est de traiter plusieurs ensembles d'apprentissage en parallèle utilisant un seul modèle de classification.

Nous pouvons citer quelques exemples de méthodes d'ensemble homogènes apparues avant les forêts aléatoires (que nous détaillerons dans la Section 3). Ces algorithmes aléatoires (bagging, random forest) permettant d'améliorer l'ajustement par une combinaison ou agrégation d'un grand nombre de modèles.

3 Exemple des méthodes d'ensembles

3.1 Le bagging :

Le Bagging est une méthode d'ensemble introduite par Breiman (1996)[Bre96]. Le mot Bagging est la contraction des mots Bootstrap et Aggregating.

Le Bootstrap est un principe de ré-échantillonnage statistique [ET93] traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. L'idée du bootstrap est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble des données observées et ce à l'aide d'un tirage aléatoire avec remise.

Supposons que l'on dispose d'un ensemble $T = \{x_1, x_2, \ldots x_n\}$ de N données observées de notre population et que l'on s'intéresse à un statistique notée S(T). Le bootstrap va consister à former L échantillons $T_k^* = \{x_1^*, x_2^*, ...x_{N'}^*\}$ k = 1..L, où chaque T*k est constitué par tirage aléatoire avec remise de N' données dans T (figure 3.7) Ces L échantillons sont usuellement appelés les échantillons bootstrap [Ber09].

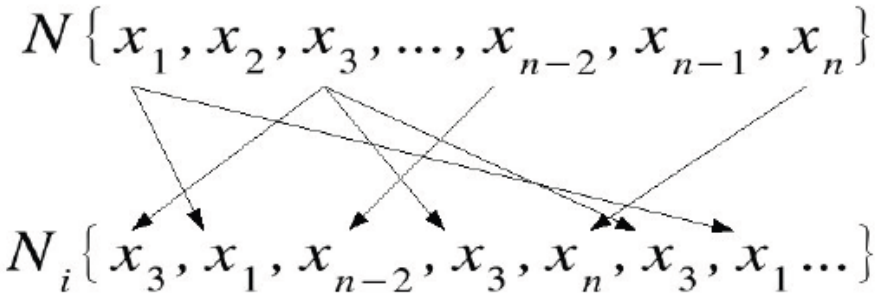

FIGURE 1.1 – Illustration d'un tirage aléatoire avec remise pour la formation d'un échantillon bootstrap.

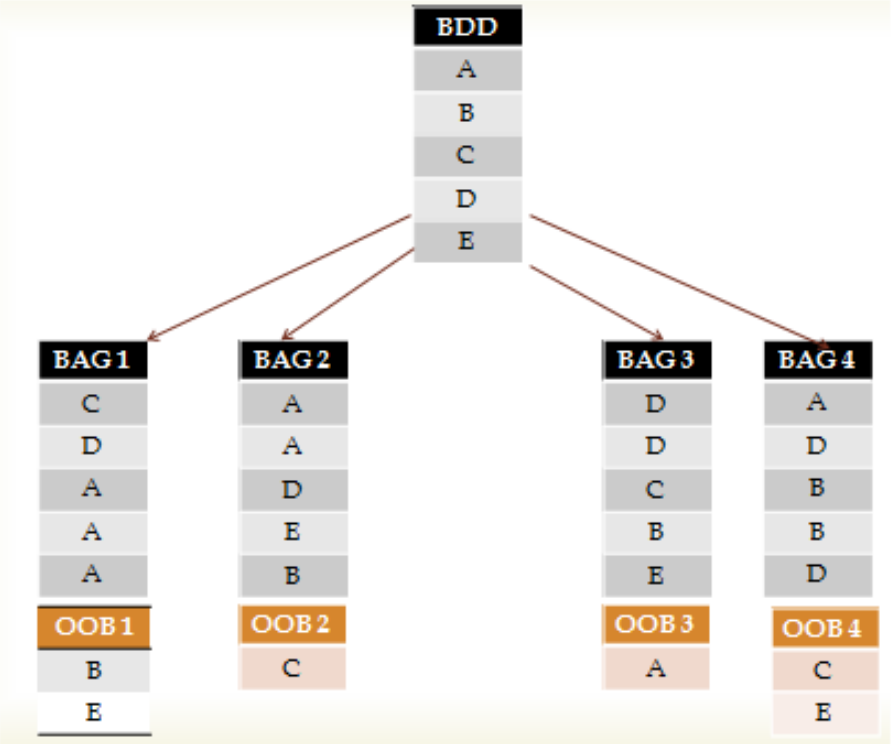

FIGURE 1.2 – Des échantillons bootstrap d'une base de données

D'après Breiman [Bre96] la statistique S(T) que l'on cherche à étudier est un algorithme d'apprentissage noté H(x). Il a appliqué alors le principe de bootstrap. Ainsi chaque classifieur élémentaire h(x) de l'ensemble sera entrainé sur un des L échantillons bootstrap de sorte qu'ils soient tous entrainés sur un ensemble d'apprentissage différent.
La figure 1.3 illustre le procédé de Bagging appliqué à un ensemble d'arbres de décision.

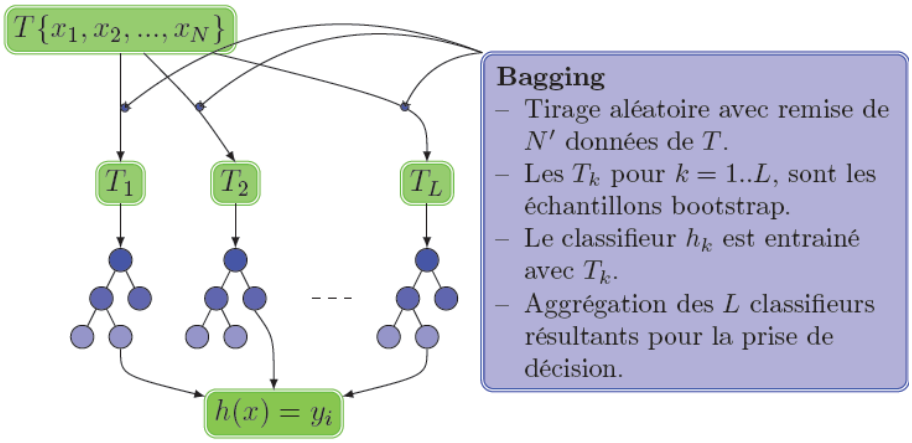

FIGURE 1.3 – Illustration du principe de Bagging pour un ensemble d'arbres de décision

Les Mesures Out-Of-Bag : = "en dehors du bootstrap" est l'ensemble des exemples qui ne sont pas sélectionnés dans les échantillons bootstrap. Ce paramètre introduit par La méthode bootstrap permet l'évaluation interne du classifieur et l'estimation de l'importance des variables pour la sélection de variables.

Soit une base d'apprentissage T_k de N exemples, après le tirage des échantillons T_k^* avec remise de N exemples, pour chaque échantillon bootstrap T_k^* , 63.2% des exemples sont uniques de T_k, le reste étant des doublons, le reste des exemples de T (le 1/3) qui ne sont pas sélectionnés sont considérés comme Out-Of-Bag [Bre96] [Ber09]
On peut faire le bagging de n'importe quel algorithme de classification à savoir RNA, SVM, ect.
Mais les plus utilisés sont les arbres de décisions. Agréger plusieurs arbres de décisions revient donc à créer ce qu'on appelle une forêt aléatoire.

3.2 Les forêts aléatoires :

Dans les cas spécifiques des modèles CART (arbres binaires), Breiman [Bre01] propose une amélioration du bagging par l'ajout d'une "randomisation". L'objectif est donc de rendre plus "indépendants" les arbres de l'agrégation en ajoutant du hasard dans le choix des variables qui interviennent dans les modèles. Avant de détailler cette méthode, nous allons d'abord donner un bref aperçu sur les arbres de décision.

Les arbres de décision : Les arbres de décisions sont des méthodes graphiques pour analyser des décisions, ils ont été conçus pour les problèmes faisant intervenir une séquence de décisions et événements successifs.

La construction des arbres de décision à partir de données est une discipline déjà ancienne, les statisticiens ont utilisé les arbres de régression dans un processus de prédiction et d'explication (AID – Automatic Interaction Detection)[MJ63].

Leur fonctionnement repose sur des heuristiques qui, tout en satisfaisant l'intuition, donnent des résultats remarquables en pratique (notamment lorsqu'ils sont utilisés en « forêts aléatoires »).

Leur structure arborescente les rend également lisibles par un être humain, contrairement à d'autres approches où le prédicteur construit est une « boîte noire » (exemple : réseau de neurone). L'arbre de décision est habituellement représenté comme des décisions ou des événements successifs représentés chronologiquement de gauche à droite :

- Les nœuds représentant des décisions sont habituellement représentés par des carrés et une branche suivant une décision sera associée à chaque possibilité de décision.
- Les nœuds représentant des événements sont habituellement représentés par des cercles et une branche suivant un événement est associé à chaque configuration envisageable

Construire un arbre de décision : La popularité de la méthode repose en grande partie sur sa simplicité. Il s'agit de trouver un partitionnement des individus que l'on représente sous la forme d'un arbre de décision. L'objectif est de produire des groupes d'individus les plus homogènes possibles du point de vue de la variable à prédire. Il est d'usage de représenter la distribution empirique de l'attribut à prédire sur chaque sommet (nœud) de l'arbre.

Choix d'une variable de segmentation : Pour choisir la variable de segmentation sur un sommet, l'algorithme teste toutes les variables potentielles et choisit celle qui maximise un critère donné. Ce critère utilisé caractérise la pureté lors du passage du sommet à segmenter vers les feuilles produites par la segmentation. Il existe un grand nombre de critères informationnels ou statistiques, les plus utilisés sont l'entropie de Shannon et le coefficient de Gini et leurs variantes. Un autre critère pour caractériser la segmentation c'est le KHI2 et ses dérivés. Il mesure le lien entre la variable candidate et la variable à prédire. Ces critères, pour peu qu'ils permettent de faire tendre le partitionnement vers la constitution de groupes purs, jouent peu sur les performances des algorithmes.

Chaque valeur de la variable de segmentation permet de produire une feuille, c'est le cas de l'algorithme C4.5 par exemple. Les algorithmes d'apprentissage peuvent différer sur ce point. Certains tels que CART produisent systématiquement des arbres binaires, il cherche donc lors de la segmentation le regroupement binaire qui optimise le critère de segmentation. D'autres tels que CHAID cherchent à effectuer les regroupements les plus pertinents en s'appuyant sur des critères statistiques. Selon la technique, nous obtiendrons des arbres plus ou moins larges [Rak05]

En résumé, la construction d'un arbre de décision se compose des étapes suivantes : (à partir de la racine de l'arbre)

1. Choix d'une variable de partitionnement parmi les attributs qui décrivent les données d'apprentissage.

2. Choix d'une ou de plusieurs valeurs de coupure de cette variable pour définir la partition.
3. Recommencer les étapes 1 et 2 avec chacun des nœuds fils qui ne remplissent pas les critères pour devenir des feuilles (Si tous les nœuds fils sont des feuilles pures, la branche courante a atteint sa taille maximale).
4. Affecter à chaque feuille une conclusion.

Donc, il existe beaucoup de type d'arbres de décisions, leurs différences se résument dans la méthode ou l'heuristique utilisée pour choisir l'attribut de division. Dans les forêts aléatoires Breiman [Bre01] propose d'utiliser les arbres de type CART qui choisissent les variables selon l'indice de GINI.

Les forêts aléatoires : Les forêts aléatoires ont été introduites par Breiman en 2001 [Bre01]]). Elles sont en général plus efficaces que les simples arbres de décision. Une forêt aléatoire est un ensemble d'arbres de décision binaire dans lequel a été introduit de l'aléatoire. Soit $\Theta_1 ... \Theta_M$ des variables, une forêt aléatoire est un ensemble de classifieurs $\{d_i(x, \Theta_i), i = 1...M\}$ où les classifieurs d_i sont construits sur le même modèle que les arbres binaires. Le nouveau classifieur correspondant à la forêt aléatoire est calculé en prenant la majorité des votes de chacun des classifieurs $d_i, i = 1..n$.

De nombreux modèles de forêts aléatoires ont été créés qui correspondent à autant de manière d'incorporer de l'aléatoire dans les arbres. Comme le Tree Bagging [Bre96] introduit de l'aléatoire dans l'échantillon initial sélectionnant certains points plutôt que d'autres et laisse grandir l'arbre jusqu'à ce que chaque nœud comporte un unique élément. Par exemple, le Tree Bagging [Bre96], le random subspace [Ho98] qui consiste à sélectionner à chaque nœud K variables de manière aléatoire et parmi celles-ci, à choisir celle qui minimise un certain critère ;il y a aussi la Random Forest [Bre01] qui consiste à mélanger le CART, le bagging et le random subspace ;ensuite on a le Random Select Split [Die98] qui sélectionne les K meilleures séparation et qui en choisit une parmi celles-ci de manière aléatoire. La position de la coupure est également calculée de manière aléatoire.

Algorithme de RF

On trouve plusieurs algorithmes le premier est l'algorithme d'induction des forêts aléatoires (Random Forests-Random Input). Cet algorithme a été introduit dans le même article (Breiman 2001) [Bre01]. Breiman a proposé le formalisme de cet algorithme et le package est codé avec Fortran 77 [BC05].

Paramètres d'algorithme Il existe deux principaux paramètres dans cette méthode :

- Le paramètre le plus important est le nombre de variables choisies aléatoirement à chacun des nœuds des arbres. Il est nommé *mtry* dans le paquet randomForest [BC05].

  Il peut varier de 1 à p (le nombre de toute les variables de la base d'apprentissage) et possède une valeur par défaut :$\sqrt{p}$ en classification, $\frac{p}{3}$ en régression.

– Le deuxième paramètre est le nombre d'arbres de la forêt. Il est nommé *ntree* et sa valeur par défaut est 500. Le choix le plus judicieux étant après plusieurs expérimentations.

Le programme permet également de régler d'autres aspects de la méthode : le nombre minimum d'observations (nommé *nodesize*) en dessous duquel on ne découpe pas un nœud, ou encore la façon d'obtenir les échantillons bootstrap (avec ou sans remise, ainsi que le nombre d'observations tirées). Nous laissons pour ces éléments les valeurs par défaut : 1 en classification et 5 en régression, et les échantillons bootstrap considérés sont tous obtenus en tirant n observations avec remise dans l'échantillon d'apprentissage L_n.

Interprétation : Comme pour tout modèle construit par agrégation ou boîte noire, il n'y a pas d'interprétation directe. Néanmoins des informations pertinentes sont obtenues par le calcul et la représentation graphique d'indices proportionnels à l'importance de chaque variable dans le modèle agrégé. C'est évidemment d'autant plus utile que les variables sont très nombreuses. Plusieurs critères sont ainsi proposés pour évaluer l'importance de la $j$ème variable.

– Le premier (*Mean Decrease Accuracy*) repose sur une permutation aléatoire des valeurs de cette variable. Il consiste à calculer la moyenne sur les observations *out-of-bag* de la décroissance de leur marge lorsque la variable est aléatoirement perturbée. La marge est ici la proportion de votes pour la vraie classe d'une observation moins le maximum des proportions des votes pour les autres classes. Il s'agit donc d'une mesure globale mais indirecte de l'influence d'une variable sur la qualité des prévisions. Plus la prévision est dégradée par la permutation des valeurs d'une variable, plus celle-ci est importante.

– Le deuxième (*Mean Decrease Gini*) est local, basé sur la décroissance d'entropie ou encore la décroissance de l'hétérogénéité définie à partir du critère de Gini. L'importance d'une variable est alors une somme pondérée des décroissances d'hétérogénéité induites lorsqu'elle est utilisée pour définir la division associée à un nœud.

– Le troisième, qui n'a pas été retenu par Breiman, le jugeant plus rudimentaire, il s'intéresse simplement à la fréquence de chacune des variables apparaissant dans les arbres de la forêt. Selon Breiman les deux premiers sont très proches, l'importance d'une variable dépend donc de sa fréquence d'apparition mais aussi des places qu'elle occupe dans chaque arbre. Ces critères sont pertinents pour une discrimination de deux classes ou, lorsqu'il y a plus de deux classes, si celles-ci sont relativement équilibrées. Dans le cas contraire, c'est-à-dire si une des classes est moins fréquente et plus difficile à discriminer, l'expérience montre que le troisième critère relativement simpliste présente un avantage : il donne une certaine importance aux variables qui sont nécessaires à la discrimination d'une classe difficile alors que celles-ci sont négligées par les deux autres critères [Agm].
les résumé du déroulement de l'algorithme d'induction des forêts est donné par [Algorithme 1], le deuxième algorithme [Algorithme 2] représente l'algorithme

de construction d'arbre de décision.

Algorithm 1 Algorithme forestRI

Entrée :T l'ensemble d'apprentissage
Entrée :L le nombre d'arbres dans la forêt
Entrée :K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud
Sortie :$foret$ l'ensemble des arbres qui composent la foret construite
pour$l$ de 1 à L **faire**
$T_l \leftarrow$ ensemble bootstrap,dont les données sont tirées aléatoirement(avec remise) de T
$arbre \leftarrow$un arbre vide, i.e composé de sa racine uniquement
$arbre.racine \leftarrow$ RndTree(arbre.racine,T_l,K)
$foret \leftarrow foret \cup arbre$
retour$foret$

Algorithm 2 RndTree

Entrée :n le nœud courant
Entrée :T l'ensemble des données associées au nœud n
Entrée :K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud
Sortie :n le même nœud, modifié par la procédure
si n n'est pas une feuille **alors**
$C \leftarrow K$ caractéristiques choisies aléatoirement
pour tout $A \in C$ $faire$
procédure CART pour la création et l'évaluation (*critère de Gini*) du partitionnement produit par A, en fonction de$T$
$partition \leftarrow$ partition qui optimise le critère de Gini
$n.ajouterFils\,(partition)$
pour tout $fils \in n.noeudFils$ **faire**
$RndTree\,(fils, fils.donnes, K)$
retour n

Conclusion

Dans ce chapitre, nous avons présenté les méthodes d'ensembles, les types des méthodes d'ensembles, la méthode *bagging* et son principe de fonctionnement. Pour qu'une méthode d'ensemble soit performante, ces classifieurs doivent être différents les uns des autres, pour cette raison nous utilisons généralement des classifieurs dits « instables ». Une méthode est instable si de petites perturbations de l'échantillon d'apprentissage peuvent engendrer de grandes modifications du classifieur obtenu. Ensuite nous avons cité les étapes de la construction d'arbre de décision, Enfin nous avons présenté les forêts aléatoires, qui sont l'amélioration du *bagging* utilisé pour les arbres de décision.

Chapitre 2

État de l'art

Introduction

A ce jour plusieurs études se sont intéressées aux forêts aléatoires, plus particulièrement à l'amélioration de l'algorithme CART au niveau du critère de segmentation, ainsi qu'à l'étape de l'agrégation des classifieurs pour obtenir un meilleur classifieur final. Nous présentons dans ce chapitre l'état de l'art des améliorations des forêts aléatoires et notre contribution dans ce domaine.

La construction d'un arbre de décision passe par plusieurs étapes ; La première étape est de choisir une variable de segmentation qui maximise un critère donné. Les travaux qui améliorent cette étape en utilisant d'autres critères de segmentation sont cités dans la section suivante

1 L'importance de variable

Dans cette partie nous citons les différents travaux réalisés dans l'état de l'art pour la classification par les méthodes d'ensemble. Parmi les travaux de la littérature ils y a ceux qui utilisent les différents critères de segmentation pour construire les arbres de décision d'une forêt aléatoire. [RS04]. D'autres travaux utilisent différents critères pour la méthode Bagging [AM09]et ceux qui utilisent une approche Bayesienne pour construire des arbres de décision comme [CMM09].

Marko Robnik-Sikonja dans [RS04]étude certaines possibilités d'augmenter ou de diminuer la force de corrélation des arbres dans la forêt. La sélection aléatoire des attributs rend les arbres individuels plutôt faibles. Le premier objectif était de renforcer individuellement les arbres sans sacrifier la variété entre eux, et d'augmenter la variance sans sacrifier la force.
Comme mesure d'évaluation l'auteur a proposé l'indice de Gini [BFOS84] et d'autres critères de segmentation comme le Gain de ratio [Qui93], ReliefF [Kon94], MDL(Minimum Description Length) [Kon95], Myopic ReliefF [Kon95, RSCK03] en tant que sélecteurs d'attribut pour diminuer la corrélation des arbres dans la forêt.
L'étude expérimentale de l'auteur repose sur deux forêts aléatoires variantes, dont la première est une méthode classique en utilisant Gini. La seconde méthode améliorée en utilisant les cinq critères suivant : (Gini, Gain de ration, MDL, RelieF et Myopic RelieF). Ses expérimentations ont été réalisées sur 16 bases de donnés de l'UCI [MA95]. Pour chaque méthode il présente les résultats des taux de classification et AUC (air sous la courbe ROC).

En utilisant le test de Wilcoxon signed-rank[Zar98] à chacune des deux RF variantes sur 17 bases de donnés, il a comparé les taux de classification qui nous donne un niveau de 0,2 de différence ceci pour une différence de AUC non significative. L'auteur a constaté qu'une amélioration a été accomplie mais qu'il ne pouvait pas être satisfait, de plus son étude était dans la région de la classification avec les forêts aléatoire.

Les travaux de Joaquin Abellan et Andrés R. Masegosa[AM09] sont des études

Indice \ Bruit	0%	5%	10%	30%
IG	85.31	83.16	80.92	68.18
IGR	85.80	83.7	81.99	70.18
GIx	85.33	82.28	81.00	68.73
IIG	86/08	85.58	84.64	75.08

TABLE 2.1 – Les moyennes des taux de classfication pour chaque méthode sur 25 bases de données en ajoutant les différents pourcentages de bruit

expérimentales utilisant des différents arbres de décision comme classifieurs par la méthode Bagging [Bre96]. Le but de ces études est de déterminer le meilleur critère de partitionnement parmi les quatre critères suivants :
Info-gain [Qui86], ratio Info-Gain [Qui93], l'indice de Gini, et imprécise Info-Gain [AM03]. Ils ont utilisés 25 bases de données de l'UCI machine Learning [MA95] avec 100 arbres sous un bruit de classification. Ils ont prouvés que la meilleure forêt Bagging et celle qui utilise le critère Imprécise Info-Gain(IIG). La plus grande valeur de taux de classification de cette méthode est 86.08%, malgré l'augmentation du bruit par rapport à celui qui est calculé par d'autres critères Tableau 2.1. L'auteur constate que par le classement de Friedman [Fri37, Fri40] le critère IIG est en première classe.

Une autre étude expérimentale d'Andres Cano et al. 2009 [CMM09]ont utilisé deux approches pour construire des arbres de décision, une approche bayésienne avec un nombre de nœuds de partitionnement K=1(K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud), et une forêt aléatoire classique dont le nombre K est variable. Pour les deux approches quatre groupes d'arbres différents ont été évaluées : 10, 50, 100 et 200. L'auteur a utilisé 23 bases de données de l'UCI [MA95] pour les expérimentations.
Après l'utilisation d'une décomposition bias-variance de l'erreur, les auteurs ont confirmés que le nombre K donne la performance pour une forêt aléatoire [Bre01]], pour une diminution de la valeur K la Variance est réduite tandis que le Bias augmente [GEW06] et vice versa. Cette tendance est rompu avec l'introduction de plus d'aléatoire dans le critère de division. Par contre ce qui donne la performance à une BRS (*Bayesian Random Split*) c'est le nombre d'arbres [CMM09], car avec un nombre plus bas d'arbre on obtient un meilleur taux d'erreur entre le bias et la variance.
Evanthia E. Tripoliti a et al. [TFAM10] utilisent trois types de forêts aléatoires : la forêt aléatoire classique(utilisant Gini), deux autres forêt aléatoire amélioré l'une avec ReliefF pour la ségmentation, et l'autres RF avec plusieurs critère d'évaluation. La motivation derrière ces méthodes d'induction de forêt est de favoriser simultanément l'augmentation des performances individuelles des arbres et l'augmentation de la diversité. Pour leurs expérimentations ils ont modélisés la base de donnés Al'zhamer Al'zhamer [AD] prétraitée par fMRI(functional magnetic resonance imaging)[JRP03, P99].
Cette base de donnée est divisé en trois sous-ensembles de base de données, un sous-ensemble pour chaque classification (deux, trois et quatre classes, respectivement).

RF	2 classes				3 classes			4 classes		
RF Gi	Var	NBtr	Se	Sp	Var	NBtr	Acc	Var	NBtr	Acc
	14	35	82	87	17	82	78	15	75	85
RF ReliefF	14	42	86	86	17	53	80	16	89	81
RF me	14	100	84	90.3	16	18	80.5	18	84	83

TABLE 2.2 – Les vlaeurs de la sensibilité, la spécificité et taux de classification des méthodes proposées

Pour les classes deux et trois, 108 instances ont été utilisés alors que pour les quatre classes 164 instances ont été utilisés. En divisant à chaque itération de la procédure les bases de donnés en deux parties une pour l'apprentissage et l'autre pour le test. Dans chacun des cas ci-dessus, un certain nombre des itérations sont effectuées avec différentes valeurs de nombre d'arbres et nombre d'attribut. Ces paramètres qui affectent la performance de l'étape de classification. Divers combinaisons de ces paramètres sont utilisées à fin de déterminer celle qui donnent des meilleurs résultats. Les valeurs optimales des paramètres sont présentées avec la moyenne des résultats des taux de classification, des valeurs de sensibilité et spécificité dans la Table 2.2. Concernant la comparaison des trois méthodes entre elles, les résultats sont difficilement interprétables. Elles semblent très proches l'une de l'autre.

Construire une collection de prédicteurs est n'est pas suffisante pour la bonne classification une autre étape est très intéressante pour construire une forêt aléatoire c'est l'agrégation de l'ensemble des prédicteurs. L'objectif visé est que ce prédicteur final soit meilleur que chacun des prédicteurs individuels. Dans l'étape suivante nous présentons les travaux qui améliorent cette étape.

2 Le mécanisme de vote

Dans une forêt aléatoire classique, l'agrégation est représentée par un vote majoritaire des prédictions des classifieurs individuels. Plusieurs travaux sont réalisés pour améliorer l'étape de l'agrégation et remplacer le vote majoritaire par d'autre mécanisme de vote. Robnik-Sikonja propose dans [RS04] d'étudier une amélioration du système de vote majoritaire classique des forêts aléatoires, dans le but d'obtenir des classifieurs finals plus performants. Son idée est de baser la sélection des arbres de décision que l'on fait participer au vote final, sur leurs performances individuelles sur des données "similaires".

Pour chaque individu de la base de test, il détermine parmi les données d'apprentissage, celles qui lui ressemblent le plus. Il décide alors de ne faire confiance qu'aux arbres de décision qui savent le mieux classer ces données dites "similaires". Pour cela il s'appuie sur la procédure utilisée par Breiman dans [Bre01] pour mesurer la similarité. Après le calcul des marges des arbres il est possible dans un premier temps d'évincer le vote final pour la classe de l'individu dont on souhaite prédire la classe.

Les arbres de décision pour lesquels la marge moyenne est strictement négative sont éliminés, puis dans un second temps de pondérer le vote final, en utilisant comme poids associé aux votes de chaque arbre leur marge moyenne.

L'auteur effectue un certain nombre d'expérimentations dans le but d'étudier les évolutions de performances des forêts aléatoires lorsqu'elles implémentent ou non le vote pondéré. Ces expérimentation on été réalisées sur 17 bases de données de UCI [MA95] utilisées par Breiman. Il suit le même protocole expérimental utilisé par Breiman [Bre01]. En utilisant le test statistique de Wilcoxon[Zar98] afin d' évaluer la différence de performances entre les forêts aléatoires utilisant un vote à la majorité classique, et celles utilisant un vote pondéré. Pour analyser ces performances, l'auteur dresse un tableau récapitulatif des taux de reconnaissances sur chaque base testée, ainsi que de leurs aires sous la courbe ROC. L'auteur prouve que le vote pondéré fournit la plupart du temps de meilleurs résultats que le vote ordinaire (11 cas sur 17). Sur les 17 nouvelles bases de données testées par souci de robustesse, le constat est identique, et le vote pondéré intégré au processus de prédiction des forêts aléatoires engendre une forte amélioration des taux de reconnaissance (dans également 11 cas sur 17) ainsi que de l'aire sous la courbe ROC [HT01].

Dans leur article [TFAM10] Evanitha et al.,présent des améliorations du système du vote ordinaire. Il s'agit de proposer six diversités de l'algorithme de vote pondéré. Le premier algorithme [RS04] est le même que celui utilisé par Breima dans [Bre01], ce principe est de baser sur les performances individuelles des arbres sur les donnés similaires. Le deuxieme algorithme [Cun08], troisieme [WM97] et le cinquième algorithme [TPC06] sont basée sur des métriques entres les données. Le quatrième [HJH$^+$06] et le sixième [SH] utilisent la pondération des arbres selon leurs taux de classification.

Les auteurs effectuent un certain nombre d'expérimentations sur la base de donnés Al'zhamer[AD] pour étudier l'évolutions de performances des forêts aléatoires lorsqu'elles implémentent le vote pondéré ou le vote majoritaire classique. Ils prennent le protocole expérimental qu'on a expliqué dans la section 2. Ensuite l'auteur a dressé leurs résultats dans les tableaux suivants en comparant les forets aléatoires avec et sans vote pondéré(wv).Voir (Table 2.3 et Table 2.4)

RF	2 classes		3 classes	4 classes
RF classique Avec vote majoritaire	Sensitivity	Specificity	Accuracy	Accuracy
	82	87	78	85
RF classique Avec vote pondéré 1	84	92.3	80	87
RF classique Avec vote pondéré 2	85	91	84	83
RF classique Avec vote pondéré 3	94	97	86	94
RF classique Avec vote pondéré 4	82	90	79	80
RF classique Avec vote pondéré 5	90	82	80	87
RF classique Avec vote pondéré 6	86	91	80	82

TABLE 2.3 – Les résultats des forêts aléatoires en utilisant l'indice de Gini avec et sans vote pondéré

RF	2 classes		3 classes	4 classes
RF améliorée Avec vote majoritaire	Sensitivity	Specificity	Accuracy	Accuracy
	84	90.3	80.5	83
RF améliorée Avec vote pondéré 1	89.3	96	82.5	86
RF améliorée Avec vote pondéré 3	91.5	98	87	88

TABLE 2.4 – Les résultats des forêts aléatoires avec et sans vote pondéré en utilisant plusieurs critères d'évaluation

Dans un processus d'induction de RF "classique" un nombre fixe d'arbres de décision randomisés sont intégrés pour former un ensemble de classifieurs. Pour cela nous présentent dans la section suivante les travaux qui améliore la sélection des arbres de décision/classifieurs.

3 La sélection des classifieurs

Dans un processus d'induction des forêts aléatoires classique, les arbres randomisés sont arbitrairement ajoutés à la forêt, mais cette méthode n'est pas la meilleure approche pour produire des bons classificateurs.

Car les capacités d'interprétation et d'analyse sont offertes par les classificateurs d'arbre de décision sont perdus en raison du principe de randomisation. Ce genre de processus où les arbres sont indépendamment ajoutés à l'ensemble n'offre aucune garantie pour que tous ces arbres seront bien coopérer dans le même comité. Cette déclaration montre que quelques arbres de décision dans un RF diminuent la performance de l'ensemble.

Cette indication a été prouvée par Simon Bernard et al. Dans [sls09], et Evanthia E. Tripoliti et al.[TFAM10] quant ils n'ont utilisés que les classifieurs performants pour faire ensuite l'estimation de leurs résultats. Andres Cano et al. dans[CMM09]

proposaient quatre groupes différents d'arbres de décisions, et il prouvaient que le nombre d'arbre donne la performance à la forêt BSR(*Bayesian Random Split*). Dans cette partie nous citons les travaux qui améliorent la sélection des classifieurs dans le but d'obtenir des bon classifieurs.

Simon Bernard et al.[sls09] ont proposés deux approches pour la selection des arbres de décision. Pour filtrer les informations ils ont utilisés les deux critères de selection "filter", "wrapper"[KJ97].L'approche "filter" consiste à sélectionner un sous-ensemble de classifieurs à l'aide d'un critère d'évaluation *a priori* qui ne prend pas en compte les performances de l'ensemble. L'approche "wrapper" en revanche réalise une sélection de sous ensembles de classifieur en optimisant *a posteriori* les performances de l'ensemble. Concernant les méthodes de sélection ils ont proposé l'approche SFS (*Sequential Forward Selection*) et SBS (*Sequential Backward Selection*). SFS est une procédure qui consiste à sélectionner un classifieur individuel parmi les classifieurs restants dans l'ensemble d'origine, de sorte que sa contribution en termes de gain de performances au sous-ensemble courant soit maximale. De la même manière, chaque itération de la procédure SBS consiste à éliminer du sous-ensemble courant le classifieur qui contribue le moins à ses performances. Sur dix bases de donnés ils ont testés ces méthodes avec un nombre initial de 300 arbres plus une méthode SRS (*Sequential Random Selection*) qui simule les inductions répétitives de RF.

Ces expérimentations ont également mis en évidence que la meilleure sous-forêt parmi celles trouvées à l'aide des deux processus de sélection, contient toujours très peu d'arbres en comparaison avec la forêt initial. Pour toutes les bases étudiées, au moins deux tiers des arbres ont du être retirés de l'ensemble initial pour atteindre le taux d'erreur le plus faible.

4 Contribution

Notre contribution porte sur l'étape d'agrégation des classifieurs. Nous proposons de remplacer le vote ordinaire classique par le vote pondéré, ce choix est justifié par le fait que le vote classique dépend du choix d'une majorité des classifieurs qui donnent une même classe pour une donnée, ces classifieurs risquent de se tromper sur la classe de cette donnée, ce qui nous a donné un classifieur final erroné. Par contre le vote pondéré sert à valider et pondérés ces classifieurs âpres les résultats. Cela nous a donné un classifieurs final plus performant.

Notre constatation que pour les travaux réalisés précédemment même en utilisant des forêts aléatoires avec des critères différents de Gini, les résultats obtenus sont presque les mêmes. En conséquence nous proposons une amélioration de l'indice de segmentation Gini, tout en gardant le même principe de Breiman dans CART(Classification and Regression Trees)[Bre01] et pour plus de précision nous avons proposé les deux diversités de Gini suivantes :(towing et deviance).

Conclusion

Beaucoup de travaux ont été réalisés en améliorant l'algorithme d'induction des forêts aléatoires et ont abouti à des résultats intéressants. Cela dit, chaque forêt aléatoire diffère d'une autre par ses paramètres expérimentaux. Dans une première partie, nous avons cité les travaux qui s'intéressent à la performance individuelle de l'arbre en se basant sur l'indice d'importance des variables calculées par les arbres de la forêt aléatoire. Cet indice permet de distinguer les variables pertinentes des variables inutiles. La procédure consiste alors à sélectionner automatiquement un sous-ensemble de variables dans un but d'interprétation ou de prédiction. Dans la deuxième partie nous présentons des travaux dont le but d'améliorer l'étape d'agrégation des arbres d'une forêt aléatoire. Cette dernière est très importante pour obtenir un classfieur final plus précis. Une troisième et dernière partie établit des méthodes de la sélection des classifieurs. Ces méthodes permettent de préciser le nombre de classifieurs pour une forêt.

Chapitre 3

Expérimentation

Introduction

Nous avons présenté dans le deuxième chapitre les travaux réalisés en améliorant l'algorithme d'induction des forêts aléatoires ainsi que notre méthode d'amélioration de cet algorithme. Dans ce troisième chapitre nous discutons les différents résultats obtenus avec les quatre bases de données.
En premier lieu nous décrivons les bases de données utilisées pour nos expérimentations, ensuite nous citons les approches proposées pour améliorer l'algorithme d'induction des forêts aléatoires, après nous passerons aux expérimentations réalisées dans le domaine et les résultats obtenus avec leurs interprétations. Enfin nous comparons nos résultats avec certains travaux portant sur différentes méthodes de classifications utilisant les mêmes bases de données, de manière à situer le travail effectué par rapport à ces travaux et mettre en évidence les contributions qu'il amène.

1 Bases de donnés

Nous utilisons, dans tout ce qui suit, quatre bases de données médicales d'UCI [MA95] pour cette expérimentation : Pima, Haberman, Breast Canceret et Liver Disorders. Chaque base de données est repartitionnée en deux parties, suivant le protocole expérimental de Breiman :

- Un ensemble d'apprentissage (2/3 de l'ensemble de données).
- Un ensemble de test (1/3 de l'ensemble de données).

1.1 La base Pima

La base de données Pima Indians Diabetes a été choisie du dépôt d'UCI [pim]. Les auteurs ont réalisé une étude sur 768 femmes Indiennes Pima (500 non diabétique 268 Diabétiques), Ces mêmes femmes, qui ont stoppé leurs migrations en Arizona, Etats Unis, adoptant un mode de vie occidentalisé, développent un diabète dans presque 50% des cas. Le diagnostic est une valeur binaire variable « classe» qui permet de savoir si le patient montre des signes de diabète selon les critères de l'organisation Mondiale de la Santé.

Les descripteurs cliniques de la base Pima : Les huit descripteurs cliniques sont :

- D1 : nombre de grossesses.
- D2 : concentration du glucose plasmatique.
- D3 : tension artérielle diastolique, (mm Hg).
- D4 : épaisseur de pli de peau du triceps, (mm).
- D5 : dose d'insuline, (mu U/ml).
- D6 : index de masse corporelle,(poids en kg/(taille m)2).
- D7 : fonction de pedigree de diabète (l'hérédité).
- D8 : âge (Année).
- Variable de classe (1 ou 2)

Analyse des données de la base PIMA :

Nombre d'attribut	Min/Max	Moyen	Ecart Type
D1	0 / 17	3 .8	3.4
D2	0 / 199	120.9	32.0
D3	0 / 122	69.10	19.4
D4	0 / 99	20.5	16.0
D5	0 / 846	79.8	115.2
D6	0 / 67.1	32	7.9
D7	0.078/2.42	0.5	0.3
D8	21 / 81	33.2	11.8

TABLE 3.1 – Informations sur les descripteurs de la base PIMA

No de la Classe	Label	Nombre D'instances
1	Tested_negative	500
2	Tested_positive	268

TABLE 3.2 – Informations sur les Instances de la base PIMA

Repartitionnement de la base PIMA :

Base d'apprentissage	Base de test
Positifs+Négatifs	Positifs+Négatifs
185+327=512	83+173=256

TABLE 3.3 – Repartitionnement de la base PIMA

1.2 La base de donnés Breast Cancer

La base de données du cancer du sein dénommée « Wisconsin Breast Cancer Database » a été collectée à l'Université du Wisconsin [Bre]. Elle contient les informations médicales de 699 cas cliniques relatifs au cancer du sein classés comme bénin ou malin : 458 patientes (soit 65%) sont des cas bénins et 241 patientes (soit 35%) sont des cas malins. La base de données contient neuf attributs qui représentent des cas cliniques, et l'attribut de la classe avec un diagnostic chiffré par 2 si le cas est bénin ,4 si le cas est malin.

Les descripteurs cliniques de la base Breast Cancer :

- D1 : l'épaisseur de la membrane plasmique d'une cellule cancéreuse.
- D2 : L'uniformité de la taille d'une cellule cancéreuse.
- D3 : L'uniformité de la forme d'une cellule cancéreuse.

- D4 : Adhesion marginale (une surexpression de la protéine integrin beta3 au niveau de la surface de la cellule cancéreuse).
- D5 : Taille cellule épithéliale (La détection des cellules épithéliales dans la moelle osseuse).
- D6 : Bare Nuclei (La détection les nucléoles qui se trouvent à l'exterieur du noyau).
- D7 : Bland Chromatin (une protéine qui induit l'expression du gène du récepteur d'œstrogènes).
- D8 : Normal Nucleoli (L'ADN protégé par une membrane nucléaire).
- D9 : Mitoses (La mitose est un processus de division cellulaire régulé).
- Variable de classe (1 ou 2).

Analyse des données de la base Breast Cancer :

Nombre d'attribut	Min/Max	Moyen
D1	1/10	4.42
D2	1/10	3.13
D3	1/10	3.2
D4	1/10	2.8
D5	1/10	3.21
D6	1/10	3.48
D7	1/10	3.44
D8	1/10	2.86
D9	1/10	1.60

TABLE 3.4 – Informations sur les descripteurs de la base Breast Cancer

No de la Classe	Label	Nombre D'instances
1	Tested_negative	458
2	Tested_positive	241

TABLE 3.5 – Informations sur les Instances de la base Breast Cancer

Repartitionnement de la base Breast Cancer :

Base d'apprentissage	Base de test
Positifs+Négatifs	Positifs+Négatifs
187+279=466	51+182=233

TABLE 3.6 – Repartitionnement de la base Breast Cancer

1.3 La base de données Haberman

L'ensemble de données contient des cas d'une étude qui a été accomplie entre 1958 et 1970 à l'Université « Chicago's Billings Hospital » sur la survivance de patients qui avaient subi la chirurgie pour le cancer du sein [hab]. La base de données contient 306 instances (225 survécu et 81 mort). Elle contient trois attributs et un quatrième pour la classe chiffré par 1 si le patient est mort pendant 5ans et 2 s'il est survécu 5ans ou plus.

Les descripteurs cliniques de la base Haberman :

- D1 : L'âge de patient au temps d'opération (numériques).
- D2 : L'année d'opération du patient (l'année - 1900, numérique)
- D3 : Le nombre de noeuds axillary positifs a découvert (numériques).
- Variable classe : Le statut de survivance (1 ou 2)

Analyse des données de la base Haberman :

Nombre d'attribut	Min/Max	Moyen
D1	30/83	52.46
D2	58/69	62.85
D3	0/52	4.14

TABLE 3.7 – Informations sur les descripteurs de la base Haberman

No de la Classe	Label	Nombre D'instances
1	Tested_negative	225
2	Tested_positive	81

TABLE 3.8 – Informations sur les Instances de la base Haberman

Repartitionnement de la base Haberman :

Base d'apprentissage	Base de test
Positifs+Négatifs	Positifs+Négatifs
56+148=204	25+77=102

TABLE 3.9 – Repartitionnement de la base Haberman

2 La base de données liver disorders

La base de données Liver disorders a été dénotée par Richard S. Forsyth dans une recherche médicale de la compagnie de soins médicaux internationaux BUPA[liv]. Elle réalise une étude médicale sur 345 individus des maladies du déséquilibre de foie (200 malades, 145 non malades). La base de données contient six attributs, les cinq premiers sont toutes les analyses de sang qui sont pensés être sensibles aux désordres de foie qui pourraient émaner de la consommation d'alcool excessive. Chaque ligne dans l'ensemble de données est constituée d'un enregistrement d'un seul individu mâle.

Il semble que les boissons$>$ 5 sont une sorte d'un sélectionneur sur cette base de données.

Les descripteurs cliniques de la base

- D1 : Le volume corpusculaire
- D2 : Phosphotase alcalin
- D3 : Alamine aminotransferase
- D4 : Aspartate aminotransferase
- D5 : Gamma-glutamyl transpeptidase
- D6 : Le nombre d'équivalents demi-d'une pinte de boissons alcoolisées bues par jour
- D7 : Variable de classe (1 ou 2)

Analyse des données de la base liver disorders :

Nombre d'attribut	Min/Max	Moyen
D1	65/103	90.16
D2	23/138	69.87
D3	4/155	30.4
D4	mai-82	24.64
D5	5/297	38.28
D6	0/20	4.8

TABLE 3.10 – Informations sur les descripteurs de la base livers disorders

No de la Classe	Label	Nombre D'instances
1	Tested_negative	145
2	Tested_positive	200

TABLE 3.11 – Informations sur les Instances de la base livers disorders

Repartitionnement de la base livers disorders :

Base d'apprentissage	Base de test
Positifs+Négatifs	Positifs+Négatifs
126+104=230	74+41=115

TABLE 3.12 – Repartitionnement de la base livers disorders

3 Approches utilisées

Pour la résolution de notre problématique, nous proposons deux améliorations de forêt aléatoire. La première est d'utiliser la diversité de Gini Twoing pour l'estimation d'attributs. La deuxième amélioration provient du mécanisme du vote, en remplaçant le vote majoritaire par le vote pondéré.

3.1 Indice d'évaluation Gini

Ce critère est celui qui est introduit et utilisé par Breiman et al. dans la méthode d'induction d'arbres CART (*Classification And Regression Tree*) [BFOS84]. C'est également le critère que Breiman a décidé d'utiliser dans ses algorithmes d'induction de forêts aléatoires pour mesurer l'impureté de noeud.
En classification, on cherche à diminuer l'indice de Gini, et donc à augmenter l'homogénéité des nœuds obtenus (un nœud étant parfaitement homogène s'il ne contient que des observations de la même classe).

Il y a beaucoup de critères selon lesquels l'impureté de nœud est minimisée dans un problème de classification, trois métriques de Gini communément utilisée incluent : Gdi, Deviance et Twoing.

La métrique Gdi :
Est l'indice utilisé par défaut dans l'algorithme de CART , il est défini par :

$$\text{Gdi}= 1\text{-}\sum_{i=1}^{k} P^2(i)$$

Où p est la proportion d'observation de la classe i dans le nœud avec la classe i qui atteignent le nœud. Un nœud avec juste une classe (un nœud pur) a l'indice zero Gini ; autrement l'indice de Gini est positif.

La métrique Twoing :
Constatant que l'indice de Gini n'est pas efficace lorsque le nombre de classes est élevé, Breiman propose dans [bre84] la règle *Twoing* qui fonctionne pour les arbres binaires, où le nombre de nœud égal à deux et la partition T se divise en deux noeuds, tG et tD.

Conçu aux problèmes de multiclasse, cette approche préfère la séparation entre les

classes plutôt que la diversité de nœud. On traite chaque division de multiclasse comme un problème binaire. Les divisions qui tiennent des ensembles de classes liées sont préférées. L'approche offre l'avantage de révéler des similarités entre les classes et peut être appliquée aux classes ordonnées aussi.

$$\text{Twoing}= p(t_G)p(t_D)(\sum_{i=1}^{k} |p_i(t_G) - p_i(t_D)|)$$

La règle de Twoing n'est pas une mesure de pureté d'un nœud, mais est une différente mesure pour décider comment diviser un nœud. En laissant $p(t_G)$ dénoter la fraction de membres de classe i dans le nœud gauche fils après la division,et $p(t_D)$ dénote la fraction de membres de classe i dans le nœud fils juste après la division. Choisissez le critère de division maximal.

Où $p(t_G)$ et $p(t_D)$ sont les fractions d'observations qui se divisent vers la gauche et la droite respectivement. Si l'expression est grande, la division a fait chaque nœud fils pure. De même si l'expression est petite, la division a rendu chaque nœud fils semblable l'un à l'autre et dorénavant semblable au nœud parent et donc la division n'a pas augmenté de pureté de nœud.

La métrique de Deviance :
Aussi appelé la trans-entropie ou la mesure de déviance d'impureté, utiliser pour calculer l'impureté de nœud. Un nœud pur a la déviance zero ; autrement, la déviance est positive, définit par :

$$\text{Deviance} = -\sum_{i=1}^{k} p(i)logp(i)$$

3.2 mécanisme de vote :

Une forêt aléatoire est l'agrégation d'une collection d'arbres aléatoires. Cette méthode revient à faire la moyenne des arbres en régression, et à faire un vote majoritaire en classification.

Le vote majoritaire :
Étant donnée une instance (x ; y) et une forêt aléatoire, on fait entrainer chaque arbre de la forêt, ce qui nous donne un ensemble de valeurs de classes prédites. Celles-ci sont alors agrégées en utilisant un vote majoritaire entre les classifieurs.Voir (figure 3.1et Algorithme 3).

Algorithm 3 Algorithme de vote majoritaire

1: Entrainer l'ensemble des classifieurs H sur une X donnée pour prédire les classes C_j
2: Calculer $V_{t,j}$ le vote pour la classe retournée par chaque classifieur

$$V_{t,j}(X) = \begin{cases} 1 & \text{si } h_t\text{donne la classe j} \\ 0 & \text{sinon.} \end{cases}$$

3: Calculer le vote total obtenu par chaque classe $V_j = \sum_{t=1}(V_{t,j})$
4: Choisir la classe qui a la plus grande valeur des votes V_j comme une classe finale pour la donnée X

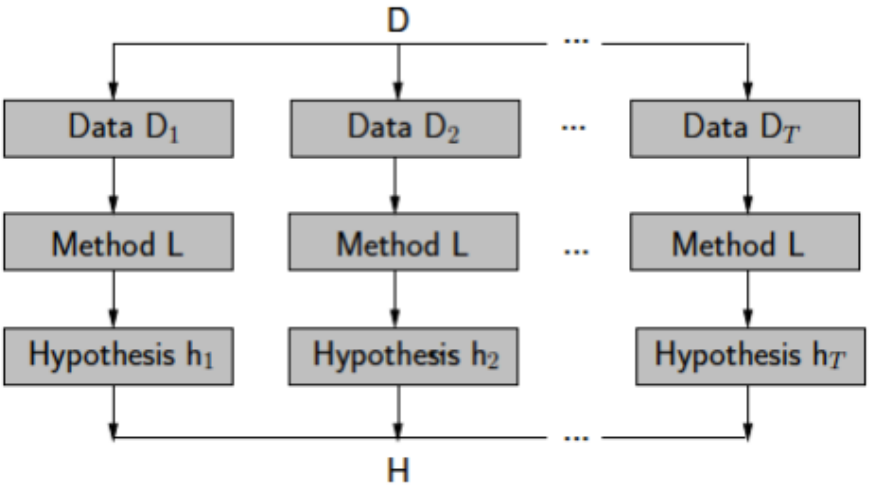

FIGURE 3.1 – Le mécanisme du vote majoritaire.

Le vote pondéré :

Dans ce travail, nous proposons d'étudier une amélioration du système de vote majoritaire classique des forêts aléatoires, dans le but d'obtenir des classifieurs finaux plus performants. L'idée est de pondérer les décisions (la classe attribuée) de chaque arbre de la forêt par sa performance locale. La performance locale de chaque arbre est tout simplement son taux de bonne classification de ses OOB. Enfin on fait la somme de tous les pondérations pour chaque classe et l'élément à classer prend la valeur de celle qui a le plus grand poids.Voir (Algorithme 4)

Algorithm 4 Algorithme de vote pondéré

1: Entrées X : Une instance à classer, T : ensemble de classifieurs, OOB de chaque classifieur.

2: Calculer la performance $Perform_t$ de chaque classifieur « t » de « T »

$$Perform_t = \text{Taux de classification } (OOB_t)$$

3: Calculer $Score_{c,x}$ le score de chaque classe « c » obtenu par les « T » classifieurs pour l'instance « X ».

$$Score_{c,x} = \sum_{t=1}^{T} (Test_{t,c}(x) * Perform_t)$$

$$Test_{t,c}(x) = \begin{cases} 1 & \text{si l'arbre "t" donne la classe "c" pour l'instance "X"} \\ 0 & \text{sinon.} \end{cases}$$

4: La classe c_x avec le score le plus élevé est choisi comme classe finale

$$C_x = Argmax(Score_{c,x})$$

3.3 Critères d'evaluation :

Les performances de classification des données ont été évaluées par le calcul des vrais positifs (VP), vrais négatifs (VN),faux positifs (FP) et faux négatifs (FN), le pourcentage de sensibilité (SE), la spécificité (SP) et le taux de classification(TC), leurs définitions respectives sont les suivantes :

- VP : Vrai Positif : nombre de positifs classés positifs.
- VN : Vrai Négatif : nombre de positifs classés négatifs.
- FP : Faux Positif : nombre de négatifs classés positifs.
- FN : Faux Négatif : nombre de négatifs classés négatifs.

La Sensibilité : c'est la capacité de donner un résultat positif quand la maladie est présente, elle est calculée par :

$$S_e = \frac{VP}{VP+VN}$$

La Spécificité : c'est la capacité de donner un résultat négatif quand la maladie est absente, elle est calculée par :

$$S_p = \frac{VN}{VN+FP}$$

Le Taux de classification : c'est le pourcentage des exemples correctement classés, il est calculé par :

$$T_c = \frac{VP+VN}{VP+VN+FP+FN}$$

3.4 Résultats et interprétation

Dans nos expérimentations nous avons utilisé une forêt aléatoire classique et une deuxième forêt améliorée. Nous utilisons les quatre bases de données suivantes pour chaque expérimentation : Pima, Breast cancer, Bupa Liver, Haberman. Les nombre des attributs à sélectionnés aléatoirement (*mtry*)pour chaque base sont : 2, 3, 2, 1 respectivement. Pour le nombre d'arbre de la forêt nous utilisons quatre groupes de 5, 10, 100 et 200 arbres de décisions (*nbtree*).

Expérimentation 1

	Arbre de décision avec Gini	forêt aléatoire classique
Pima	73.43	82.03
Breast_cancer	94.84	99.14
Bupa_liver	69.56	72 .17
Haberman	69.60	72.54

TABLE 3.13 – Les résultats de l'arbre et de la forêt

Le tableau (Table 3.13) contient les taux de classification d'un seul arbre de décision et d'une forêt aléatoire classique avec vote majoritaire utilisant Gini comme critère de segmentation pour ces deux classifieurs.

En comparant les résultats de l'arbre de décision avec la forêt aléatoire qui contient 100 arbres de décision pour chaque base de données, nous remarquons que la forêt fournit des meilleurs résultats par rapport à un seul arbre de décision pour toutes les bases de données.

Expérimentation 2

Dans le tableau (Table 3.14) nous comparons les performances (taux de classification) des trois différentes forêts aléatoires. La première forêt utilise le critère de Gini(Gdi) pour l'évaluation des attributs et la deuxième et la troisième utilisent les deux diversités de gini (Twoing, Deviance). Nous présentons les résultats des taux de classification. Pour chaque critère d'évaluation nous remarquons une stabilisation des taux de classification pour 100 arbres de décisions. Nous remarquons également que les deux critères (Towing et Deviance) fournissent la plupart du temps de meilleurs résultats par rapport à l'indice de Gini (Gdi), (3 cas sur 4), le 4eme cas les résultats sont presque identiques pour les deux forêts.

Bases de données	Nombre d'arbres	Vote majoritaire		
		GDI	TWOING	DEVIANCE
Pima	10	80.0781	77.3438	79.2969
	50	80.0781	78.5156	82.4219
	100	81.6406	**82.0313**	**82.0313**
	200	80.0781	81.6406	82.0313
Breast Cancer	10	98.7124	98.7124	97.8541
	50	99.5708	99.5708	99.5708
	100	99.5708	99.1416	99.5708
	200	99.5708	99.5708	99.5708
Liver	10	69.5652	70.4348	66.0870
	50	71.3043	66.9565	68.6957
	100	68.6957	**70.4348**	**71.3043**
	200	71.3043	70.4348	69.5652
Haberman	10	71.5686	73.5294	72.5490
	50	72.5490	72.5490	72.5490
	100	72.5490	**73.5294**	**72.5490**
	200	73.5294	72.5490	72.5490

TABLE 3.14 – Les performance des forêts aléatoires utilisant Gini et ses deux varaintes

Expérimentation 3

Dans le but d'améliorer la performance de l'expérimentation précédente nous avons remplacés le vote majoritaire par le vote pondéré, utilisant les mêmes critères pour les deux forêts. Le tableau (Table 3.15) contient les résultats des deux forêts utilisant Gini, en comparant les taux de classification des forêts avec et sans vote pondéré. Nous remarquons que le vote pondéré augmente la performance de la forêt surtout pour les bases Pima et Bupa Liver pour des différents nbtree. Nous remarquons que les meilleurs résultats ont été obtenus avec des forêts qui contiennent 100 arbres de décisions. Aussi à partir de 200 arbres les performances restent constantes.(Voir figure 3.2)

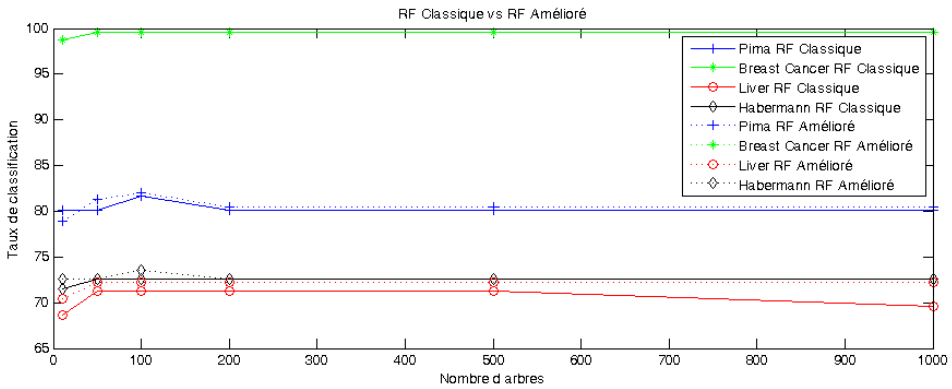

FIGURE 3.2 – RF vs RF amélioré

Base de données	Nombre d'arbre	RF classsique (Gdi et vm)	RF amélioré (Gdi et vp)
Pima	10	80.0781	78.9063
	50	80.0781	81.2500
	100	81.6406	82.0313
	200	80.0781	80.4688
	500	80.0781	80.4688
	1000	80.0781	80.4688
Breast Cancer	10	98.7124	98.7124
	50	99.5708	99.5708
	100	99.5708	99.5708
	200	99.5708	99.5708
	500	99.5708	99.5708
	1000	99.5708	99.5708
Liver	10	68.6957	70.4348
	50	71.3043	72.1739
	100	71.3043	72.1739
	200	71.3043	72.1739
	500	71.3043	72.1739
	1000	69.9056	72.1739
Haberman	10	71.5686	72.5490
	50	72.5490	72.5490
	100	72.5490	73.5294
	200	72.5490	72.5490
	500	72.5490	72.5490
	1000	72.5490	72.5490

TABLE 3.15 – Les résultas des deux forêts aléatoire avec vote majoritaire et pondéré

Expérimentation 4 :

Pour 100 arbres nous avons construit deux forêts aléatoires différentes, la première est une forêt aléatoire classique avec l'indice de Gini(Gdi) et le vote majoritaire pour agréger les 100 classifieurs. La deuxième forêt est amélioré en remplaçant Gini par Towing et un vote pondéré pour l'agrégation. On exécute le programme dix fois pour chaque forêts a fin de comparer les performances des deux méthodes sur les quatre bases de données.

Pima :

	RF classique			**RF amélioré**		
	TC	S	Sp	TC	Se	Sp
Pima	82.03	62.65	91.32	**83.98**	68,67	91,33
	80.46	60.24	90.17	80.86	61.44	90.17
	80.85	60,24	90.75	80.07	59.03	90.17
	82.42	65.06	90.75	82.42	62.65	91.90
	81.64	62.65	90.75	82.03	62.65	91.33
	81.64	62.65	90.75	81.25	66.26	88.43
	81.25	62.65	90.17	83.20	68.67	90.17
	80.07	60.24	89.6	82.81	66.26	90.75
	79.68	59.04	89.6	83.59	67.46	91.32
	80.46	60.24	90.17	82.81	65.06	91.33
Moyenne	**81.05**	61.56	90.4	**82.3**	64.81	90.69

TABLE 3.16 – Les performance de la forêt amélioré pour ma base Pima

Le tableau (Table 3.16) contient les valeurs des taux de classification, la sensibilité et la spécificité pour les deux forêts variantes utilisant la base de données Pima.

Les résultats de cette expérimentation montrent que la spécificité des deux forêts est très élevée ce qui veut dire que les forêts aléatoires ont fait un bon apprentissage pour les données négatives donc lorsqu'un patient est non diabétique elles le détectent avec succès.

Une moyenne sensibilité des deux forêts et un très bon taux de classification pour les deux forêts.

Notre constat immédiat, en comparant les résultats obtenus, est que la forêt aléatoire amélioré fournit la plupart du temps de meilleurs résultats que la forêt aléatoire classique (7 cas sur 10).

Breast Cancer :

	RF classique			**RF amélioré**		
	TC%	Se%	Sp%	TC%	Se%	Sp%
Breast cancer	**99.57**	98.15	100	**99.57**	98.14	100
	99.14	98.15	99.44	99.57	98.14	100
	99.14	98.15	99.44	99.14	98.14	99.44
	99.14	98.15	99.44	99.57	98.14	100
	99.14	98.15	99.44	99.14	98.14	99.44
	99.14	98.15	99.44	99.14	98.14	99.44
	99.57	98.15	100	99.57	98.14	100
	99.14	98.15	99.44	99.14	98.14	99.44
	99.14	98.15	99.44	99.14	98.14	99.44
	99.14	98.15	99.44	99.14	98.14	99.44
Moyenne	**99.28**	98.15	9.62	**99.42**	99.14	99.81

TABLE 3.17 – Les performance de la forêt amélioré pour ma base Breast Cancer

Le Tableau (Table 3.17) résume les résultats obtenus par les deux forêts aléatoires utilisant la base de données Breast Cancer. Les deux forêts aléatoires donnent des valeurs de sensibilité et spécificité très élevés, ces résultats montrent que les forêts aléatoires ont fait un bon apprentissage pour les données négatives et positives, donc elles réalisent une très bonne détections des patients bénins et malins.

En comparant les résultats des deux forêts nous remarquons que les résultats sont presque identique vu la nature de la base de donnée. Notant que des très travaux utilisant les réseaux de neurones ont obtenus 100% de bonne classification de cette base.

Bupa liver :

	RF classique			**RF amélioré**		
	TC%	Se%	Sp%	TC%	Se%	Sp%
Bupa Liver	69.56	85.14	41.46	73.04	89.18	43.90
	71.3	85.14	46.34	72.17	87.83	43.90
	70.43	86.49	41.46	70.43	86.48	41.46
	69.56	83.78	43.90	73.91	87.84	48.78
	72.17	87.84	43.9	71.30	86.48	43.90
	68.69	83.78	41.46	**74.78**	93.24	41.46
	72.17	87.84	43.9	71.30	85.13	46.34
	73.04	90.54	41.46	73.91	89.18	46.34
	70.43	86.49	41.46	69.56	82.43	46.34
	68.69	85.13	39.02	72.17	87.83	43.90
Moyenne	**70.72**	86.33	42.54	**72.17**	87.38	44.71

TABLE 3.18 – Les performance de la forêt amélioré pour ma base Bupa Liver

Dans le tableau (Table 3.18) nous citons les résultats des deux forêts aléatoires pour la base de données Bupa-liver.

La sensibilité des deux méthodes est très élevée, ce la veut dire que les deux forêts ont fait un bon apprentissage pour les données positives. Lorsqu'un patient est malade il a été détecté avec beaucoup de succès. Par contre la spécificité des deux méthodes est faible donc la détection des patients non malades est moins bonne.

En comparant les deux forêts aléatoires, nous remarquons que la forêt amélioré fournit la plupart du temps de meilleurs résultats que la forêt classique (8 cas sur 10).

Haberman :

	RF classique			RF amélioré		
	TC%	Se%	Sp%	TC%	Se%	Sp%
Haberman	72,54	40	83,12	72.54	44	81.81
	73,52	40	84,42	71.56	40	81.81
	74,5	44	84,42	72.54	48	80.51
	71,56	40	81,82	73.52	48	81.81
	73,52	48	81,82	74.50	44	84.41
	70,58	36	81,82	**76.47**	40	88.31
	72,54	48	80,52	73.52	44	83.11
	75,49	44	85,71	71.56	36	83.11
	71,56	40	81,82	73.52	48	81.81
	72,54	36	84,42	72.54	44	81.81
Moyenne	**72,84**	41,6	82,98	**73,23**	43,6	82,85

TABLE 3.19 – Les performance de la forêt amélioré pour ma base Haberman

Le tableau (Table 3.19) contient les résultats obtenus par les deux forêts aléatoires utilisés dans cette expérimentation, utilisant la base de données Haberman.

La sensibilité des deux méthodes est faible, donc on obtient une mauvaise détection des patients qui sont morts pendant cinq ans. Une très bonne valeur de la spécificité des deux forêts cela nous donne une bonne détection des patients survécus pendant les cinq ans.

Pour la comparaison des deux forêts aléatoires classiques et amélioré, on constate que la deuxième méthode donne la plus par de temps des meilleurs résultats par rapport au forêt classique.

4 Etude comparative :

Bases	Auteur/ Résultât	PMC	SVM	N-Flou	RF	Notre RF
Pima	Auteur	[KM10]	[Hua06]	[NK98]	[RS04]	Notre travail
	Résultat	78.21%	81.50%	79.84%	77.3%	82,30%
B.Cancer	Auteur	[ZB12]	[GKS11]	[HG10]	[RS04]	Notre travail
	Résultat	98.97%	97.13 %	97.87%	96.7	99,42%
Liver	Auteur	[BSB12]	[GKS11]	[KKS03]	[RS04]	Notre travail
	Résultat	76.47%	61.16%	56.72%	71.9%	72,17%

TABLE 3.20 – Etude comparative

Après plusieurs recherches sur cette problématique nous pouvons confirmer que notre méthode fournit des meilleurs résultats en comparant avec d'autres techniques de classification. Nous remarquons aussi que notre méthode est plus performante par rapport à la forêt aléatoire amélioré de Robnik,2004 [RS04]

Conclusion

Nous avons présenté une forêt aléatoire amélioré qui a donné une amélioration intéressante de taux de classification de forêt classique, tout en optimisant l'algorithme de la construction de ses classifieurs. Après la comparaison des résultats obtenus avec d'autres travaux qui améliorent les forêt aléatoires et d'autres différents travaux de la littérature, nous avons remarqué que les résultats trouvés sont comparables et la plupart du temps meilleurs par rapport aux autres résultats.

Conclusion

Nous nous sommes intéressés dans ce travail à étudier la performance d'un modèle ensembliste appelé Random Forest sur une tâche de classification reliée au domaine médical.
À cet effet, nous avons procédé en premier temps à une analyse de travaux effectué dans le domaine qui a permis de mettre en évidence plusieurs avantages ainsi que certaines limites des forêts aléatoires employées dans le cadre de la classification des données médicales. Nous avons constaté en conséquence que les classifieurs déjà proposés à base des forets aléatoires font preuve de bonnes performances mais peuvent être encore amélioré pour apporter plus de précisions aux résultats.

Afin de créer une application performante utilisée pour la classification, nous avons implémenté une méthode qui utilise plusieurs variations des forêts aléatoires. Pour cela nous avons utilisé quatre bases de données d'UCI : pima, breast-cancer, Bupa-liver et Heberman pour évaluer notre modèle.

Nous avons en premier lieu ré-implémenté le RF-classique en utilisant l'indice de Gini et le vote majoritaire, puis nous avons procédé au développement de de plusieurs variances du même classifieur en utilisant l'indice de Déviance et Twoing rule au niveau du choix de la variable de division des nœuds des arbres et enfin on a utilisé le vote pondéré comme méthode d'agrégation de l'ensemble d'arbres.

Nous avons évalué et testé les performances de chaque foret en termes de sensibilité (Se), Spécificité (Sp) et le taux de classification correcte (CC).

Le taux de classification obtenu avec notre méthode est parmi les meilleurs résultats obtenus jusqu'à maintenant pour la classification de ces bases de données. Les résultats obtenus sont très compétitif par rapports aux autres versions des forêts aléatoires.

Dans les perspectives de ce travail, nous souhaitons apporter d'autres améliorations aux forets aléatoires en remplaçant l'indice de Gini par ReliF. Nous envisageons également intégrer cette application dans un système d'aide au diagnostic médical pour l'utilisation dans un hôpital ou dans un cabinet médical.

Annexe A : Modélisation de l'application

Notre approche de conception suit le modèle UML (Unified Modelling Language. Plusieurs raisons nous ont conduits à ce choix :

- La première est sa normalisation par l'OMG [OMG03]. L'historique a montré que la profusion des notations est, à moyen terme, préjudiciable aux entreprises comme à leurs fournisseurs.

- La deuxième raison est l'intérêt montré par les informaticiens pour ce langage de modélisation.

- La troisième raison est la possibilité d'utiliser le même atelier de génie logiciel,depuis l'expression des besoins jusqu'à la génération de tout ou partie de l'application.

- La dernière raison, mais non la moindre, est d'utiliser les principes et concepts objet pour enrichir la démarche de conception des systèmes d'aide à la décision. On en attend des améliorations dans le sens tout à la fois de la richesse, d'une modularité, d'une cohérence et d'une rigueur accrues

Définition des besoins de l'utilisateur :

L'utilisateur de notre application est le médecin, il a besoin d'un outil pour l'aider à effectuer un diagnostique des maladies pour un patient, cette application doit assurer les caractéristiques suivantes :

- Une meilleure performance.
- Un minimum d'erreur.
- Demande le minimum d'informations pour faire le diagnostique (les plus pertinentes).
- Interprétable (donne la raison pour laquelle il a trouvé ces résultats).

Les acteurs principaux de l'application :

Un acteur représente un rôle joué par une personne, un groupe de personnes ou par une chose qui interagit avec le modèle. Les acteurs se recrutent parmi les

utilisateurs de l'application et aussi parmi les responsables de sa configuration et de sa maintenance. Dans notre modèle nous avons considéré deux acteurs principaux :

- Dans notre cas l'administrateur est l'informaticien ou le programmeur, il est fortement impliqué dans l'organisation de l'application, il a comme rôle la gestion et le contrôle de l'application.
- Le médecin est l'utilisateur principale de l'application, il l'utilise pour l'aider à faire un diagnostique des maladies pour un patient.

L'étude des cas d'utilisation a pour objectif de déterminer ce que chaque acteur attend du système. La détermination des besoins est basée sur la représentation de l'interaction entre l'acteur et le système. Donc un cas d'utilisation est une abstraction d'une partie de comportement du système, qui s'instancié à chaque utilisation du système par une instance d'un acteur.

Diagramme de cas d'utilisation création d'un classifieur :

Les cas d'utilisation représentent un élément essentiel de la modélisation orientée objets : ils interviennent très tôt dans la conception, et doivent en principe permettre de concevoir, et de construire un système adapté aux besoins de l'utilisateur (Build the right system).

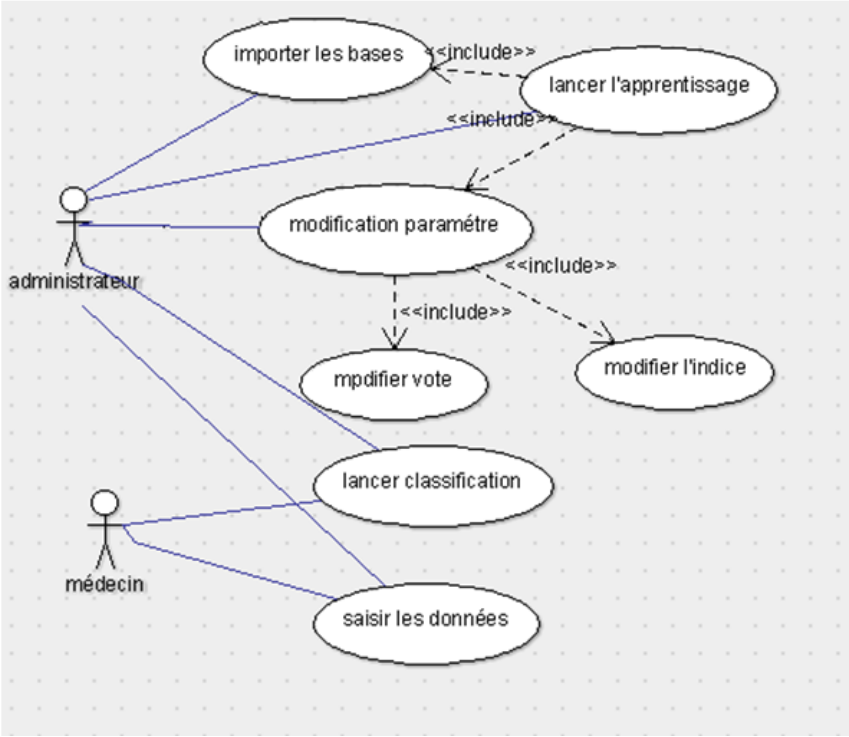

FIGURE 3.3 – Le diagramme de cas d'utilisation

Le diagramme de classes :

Après avoir effectuée une modélisation par les cas d'utilisation qui correspond à une description fonctionnelle, il est nécessaire de passer à une modélisation structurelle, statique du système à réaliser.

Dans la réalisation de notre application nous avons considéré trois grandes classes et une interface voir (Figure 3.9) :

Classe BDD Elle est utilisée pour importer le fichier de la base de données, et analyser les données de cette base, elle contient six méthodes :

- ChargerBase : une méthode publique utilisée pour charger le fichier de la base de données, ce fichier est de type (.txt), elle a comme entrée le nom de la base.
- GetBdd : une méthode publique utilisée pour récupérer la base de données charger du fichier source.
- GetNbrAttr : elle donne le nombre d'attributs ou de variables de la base.
- GetNbrInst : elle donne le nombre d'instances de la base.
- GetNbrPos : elle donne le nombre d'instances positif de la base.
- GetNbrNeg : elle donne le nombre d'instances négatif de la base.

Classe RF classique Elle est utilisée pour la classification des données par les forêts aléatoire classique cette classe contient sept méthodes :

- choixNbrTree() : une méthode publique utilisée pour choisir le nombre(des classifieur) des arbres de foret .

- Apprendre : une méthode publique utilisée pour l'apprentissage classique de la forêt aléatoire par l'indice de GINI ;elle a comme entrées la base d'apprentissage(Bag) et mtry(la racine de nombre d'attribut) , elle retourne l'ensemble d'arbre (structure).

- Tester : une méthode publique utilisée pour le test des arbres après l'apprentissage elle a comme entrée la base de test et elle retourne le tableau de classification des instances pour chaque arbres.

- Vote-maj : une méthode publique utilisée pour faire le vote majoritaire entre les résultats des classifieurs pour chaque instances, elle retourne le vecteur de classification des instances de la foret.

- CalculerTaux() :calculer le taux de classification.

- calculerSensi() :calculer la sensibilité .

- claculerSpeci() :calculer la spécificité.

Classe RF améliorée Elle est utilisée pour la classification des individus par la méthode des forets aléatoire améliorée au niveau de la construction d'arbre (l'indice de division)et au niveau d'agrégation des classifieur (vote pondéré) pour obtenue le résultat final , cette méthode contient toutes les méthodes de la classe précédente avec trois autres méthodes en plus :

- remplacer-indice() : une méthode publique utilisée pour modifier l'indice de division au niveau de construction l'arbre on remplace GDI par TWOING ou DEVIANCE

- calculerScoreAr() : une méthode publique utilisée pour calculer le scores de chaque classifieurs en utilisant les out-of-bag.

- modifier-vote() : une méthode publique utilisée pour améliorer vote majoritaire par le vote pondérer

Interface Elle est utilisée pour faire l'interaction entre les trois premières classes, et utilise des objets de ces classes, elle est utilisé aussi pour récupérer les données saisies par l'utilisateur et afficher les résultats obtenus, pour plus de détails sur le fonctionnement de l'interface voir [AnnexeB, Manuel d'utilisation].

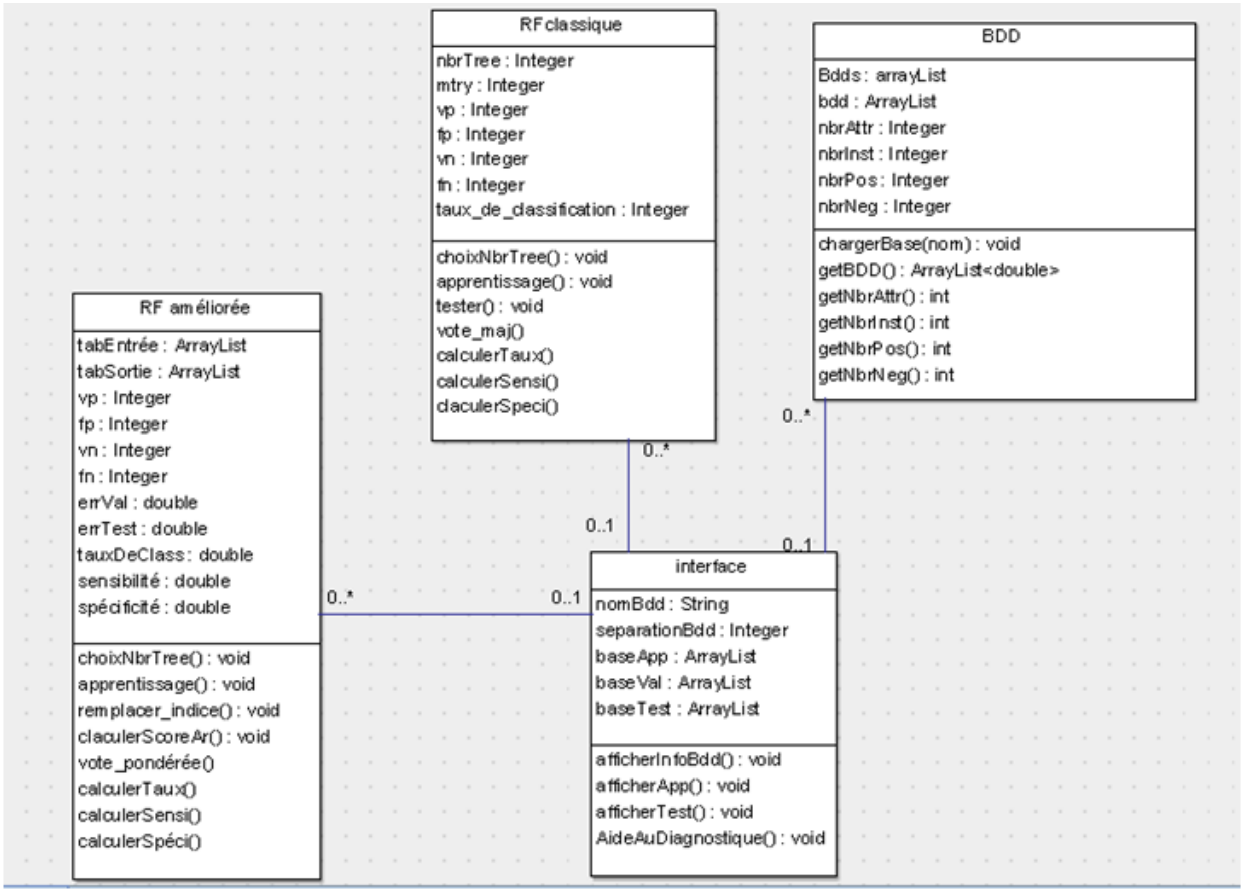

FIGURE 3.4 – Le diagramme de classes

Annexe B : Manuel d'utilisation

Matlab permet d'écrire assez simplement une interface graphique pour faire une application interactive utilisable par des utilisateurs non formés à Matlab. Elle contient des panneau modulaires ,chaque panneau est responsable d'une tache bien définie ;

FIGURE 3.5 – l'interface principale de l'application

La figure 3.5 présente notre interface principale de notre application. Il y a deux boutons un pour l'accès à l'application et l'autre pour la quitter. Une fois accéder à l'application l'utilisateur a une interface qui est divisée essentiellement en trois parties (Apprentissage, classification et informations). La figure suivante est une capture d'écran de cette interface.

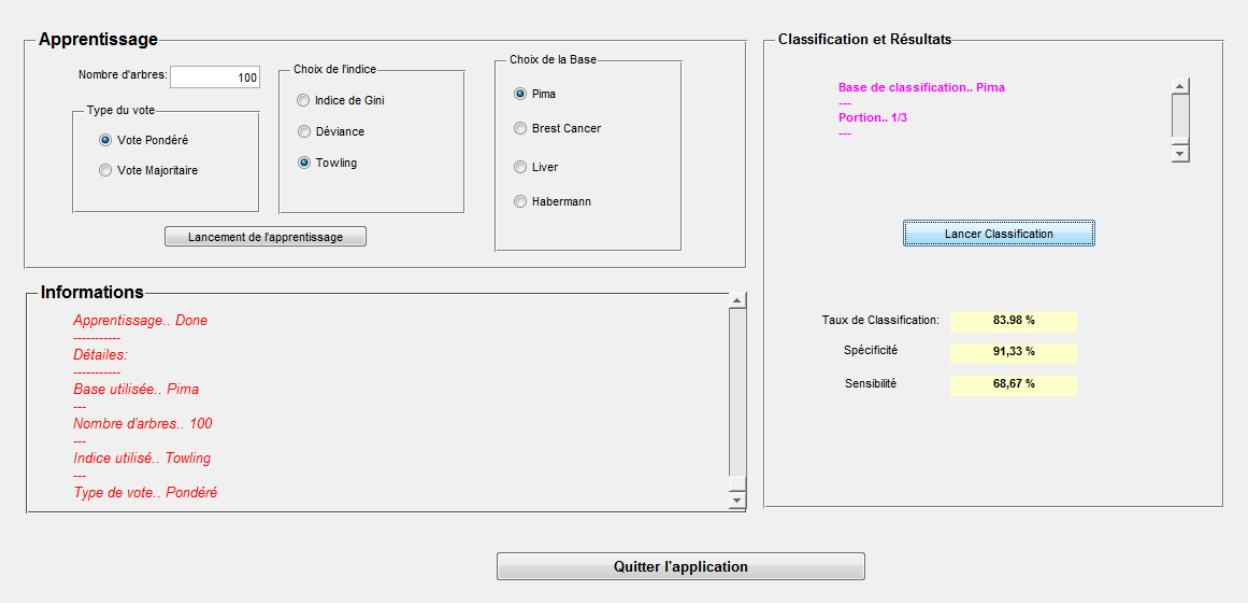

FIGURE 3.6 – l'interface de classification

En ce qui suit nous allons donner plus de détails sur chaque partie de cette deuxième interface.

APPRENTISSAGE :

Dans cette partie ce fait l'apprentissage de notre application, On remarque bien que tous les paramètres des méthodes utilisés y figure. Pour cela l'utilisateur a les possibilités suivantes :

1. choix de la base d'apprentissage
2. choix le nombre d'arbres (classifieurs)
3. choix l'indice utilisée pour construire les arbres
4. Choix le type de classification (vote pondérée ,vote majoritaire)
5. Bouton pour lancer l'apprentissage.

FIGURE 3.7 – panneau du choix des paramètres d'apprentissage

Information :

Dans cette partie tous les choix et les actions de l'utilisateur sont affichés :

1. Affichage de la base de données choisie
2. Nombres d'arbres
3. L'indice utilisé
4. Le Type de vote

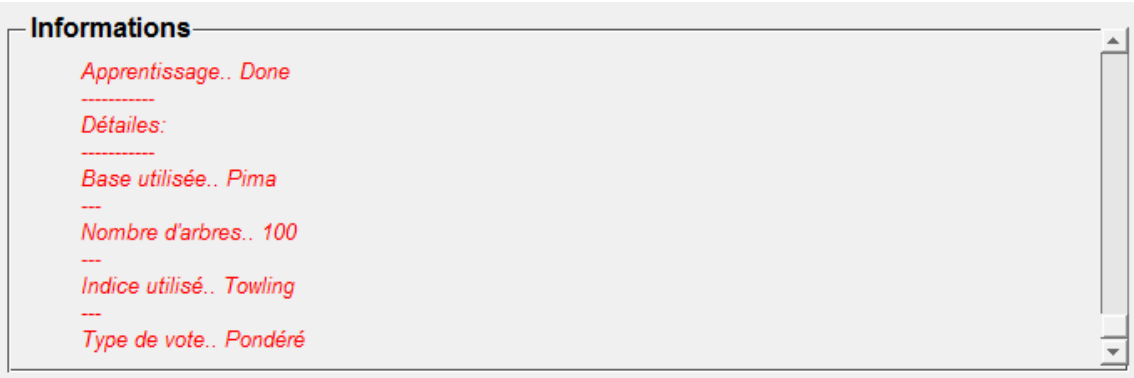

FIGURE 3.8 – Panneau d'affichage des détails d'apprentissages.

classification et résultats :

Cette partie concerne la classification des données ainsi que les résultats obtenus avec le classifieur, elle contient :

1. Zone d'affichage des paramètre du test.
2. Lancement de la classification ;
3. Affichage du taux de classification ;
4. Affichage de la spécificité ;
5. Affichage de la sensibilité.

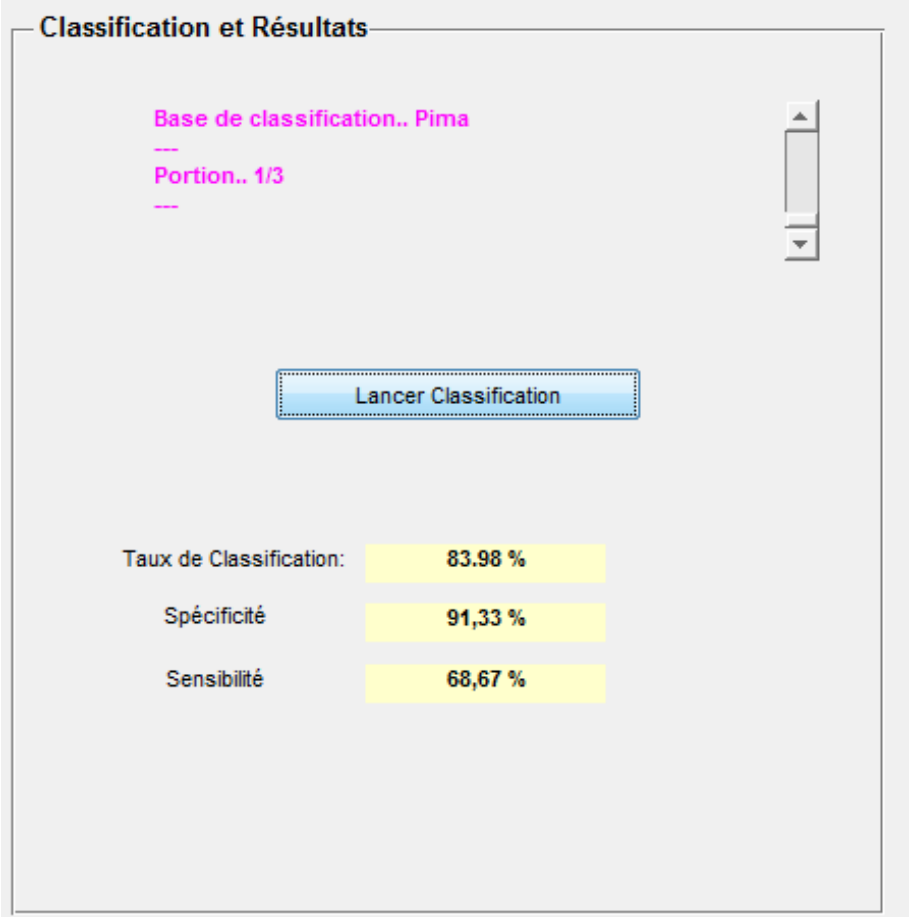

FIGURE 3.9 – panneau d'affichage des résultats de classification

Bibliographie

[AD] http://alzheimers.org.uk/factsheet/407.

[Agm] Agrégation de modèles. http://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-agreg.pdf.

[AM03] J. Abellan and S. Moral. Building classification trees using the total uncertainty criterion. *Int. J. Intell. Syst.*, 18(12) :1215–1225, 2003.

[AM09] Joaquín Abellán and Andrés R. Masegosa. A filter-wrapper method to select variables for the naive bayes classifier based on credal decision trees. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 17*, 6 :833–854, 2009.

[BC05] Leo Breiman and Adele Cutler. Random forests. 2005. http://www.stat.berkeley.edu/users/breiman/RandomForests/.

[Ber09] Simon Bernard. Forêts aléatoires : De l'analyse des mécanismes de fonctionnement à la construction dynamique. Thèse, Université de Rouen, 2009.

[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth Publishing Company, 1984.

[BKR09] Jon Atli Benediktsson, Josef Kittler, and Fabio Roli, editors. *Multiple Classifier Systems, 8th International Workshop,MCS 2009, Reykjavik, Iceland, June 10-12, 2009. Proceedings*, volume 5519. Springer, 2009.

[Bre] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsi.

[bre84] *L. Breiman and J. Friedman and R. Olshen and C. Stone*. Wadsworth and Brooks, Monterey, CA, 1984.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[BSB12] Peiman Mamani Barnaghi, Vahid Alizadeh Sahzabi, and Azuraliza Abu Bakar. A comparative study for various methods of classification. *2012 International Conference on Information and Computer Networks*, 27, 2012.

[CMM09] Andrés Cano, Andrés R. Masegosa, and Serafín Moral. A bayesian random split to build ensembles of classification trees. *CAEPIA 2009*, pages 469–480, 2009.

[Cun08] Padraig Cunningham. A taxonomy of similarity mechanisms for case-based reasoning. *Technical Report UCD-CSI-2008-01*, 2008.

[Die98] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine Learning*, 40 :139–157, 1998.

[ET93] B. Efron and R. Tibshirani. An introduction to the bootstrap. Chapman and Hall, 1993.

[Fri37] M. Friedman. The use of rank to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, page 675–701, 1937.

[Fri40] 86–92 (1940) Friedman, M. : A comparison of alternative tests of significance for the problem of m rankings. Annals of Mathematical Statistics 11. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, (11) :86–92, 1940.

[Gen10] Robin Genuer. Forêts aléatoires : aspects théoriques, sélection de variables et applications. Thèse, Université Paris-Sud11, 2010.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1) :3–42, 2006.

[GKS11] Shelly Gupta, Dharminder Kumar, and Anand Sharma. Performance analysis of various data mining classification techniques on healthcare data. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(4), 2011.

[hab] `http://archive.ics.uci.edu/ml/datasets/Haberman+Survival`.

[HG10] Hazlina Hamdan and Jonathan M. Garibaldi. Adaptive neuro-fuzzy inference system (anfis) in modelling breast cancer survival. *WCCI 2010 IEEE World Congress on Computational Intelligence*, 2010.

[HJH+06] Hu H, Li J, Wang H, Daggard G, and Shi M, editors. *A maximally diversified multiple decision tree algorithm for microarray data classification.* The 2006 workshop on intelligent systems for bioinformatics (WISB2006), 2006.

[HKR07] Michal Haindl, Josef Kittler, and Fabio Roli, editors. *Multiple Classifier Systems, 7th International Workshop, MCS 2007, Prague, Czech Republic, May 23-25, 2007, Proceedings*, volume 4472. Springer, 2007.

[Ho98] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 :832–844, 1998.

[HT01] David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning Journal*, 45 :171–186, 2001.

[Hua06] Chieh-Jen Wang Cheng-Lung Huang. A ga-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31 :231–240, 2006.

[JRP03] Petrella JR, Edward Coleman R, and Murali Doraiswamy P. Neuroimaging and early diagnosis of alzheimer disease. *a look to the future. Radiology*, 226 :315–336, 2003.

[KJ97] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2) :273–324, 1997.

[KKS03] U.V. Kulkarni, B.B.M. KrishnaKanth, and T.R. Sontakke. Detection of liver disorder using fuzzy neural classifiers. *SGGS College of Engineering and Technology, Nanded*, pages pp. 553–557, 2003.

[KM10] M.A.Jayaram Asha Gowda Karegowda and A. S. Manjunath. Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 02(2) :271–277, 2010.

[Kon94] Igor Kononenko. *Estimating attributes : analysis and extensions of Relief.* In Luc De Raedt and Francesco Bergadano, Springer Verlag, Berlin,, 1994.

[Kon95] Igor Kononenko. *On Biases in Estimating Multi-Valued Attributes.* 1995.

[KR00] Josef Kittler and Fabio Roli, editors. *Multiple Classifier Systems, First International Workshop,MCS 2000, Cagliari, Italy, June 21-23, 2000, Proceedings*, volume 1857. Springer, 2000.

[KR01] Josef Kittler and Fabio Roli, editors. *Multiple Classifier Systems, Second International Workshop,MCS 2001 Cambridge, UK, July 2-4, 2001, Proceedings*, volume 2096. Springer, 2001.

[Lec07] Guillaume Lecue. Méthodes d'agrégation : optimalité et vitesses rapides. Thèse, Université Paris VI, 2007.

[liv] `http://archive.ics.uci.edu/ml/datasets/Liver+Disorders`.

[MA95] Patrick M. Murphy and David W. Aha. Uci repository of machine learning databases. 1995. `http://www.ics.uci.edu/mlearn/MLRepository.html`.

[MJ63] Sonquist J.A. Morgan J. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58 :415–435, 1963.

[Nil65] N. Nilsson. *Learning machines.* McGraw-Hill, 1965.

[NK98] D. Nauck and R. Kruse. Nefclass-x - a soft computing tool to build readable fuzzy classifiers. *T Technology Journal*, 16(3) :180–190, 1998.

[OPKR05] Nikunj C. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, editors. *Multiple Classifier Systems, 6th International Workshop, MCS 2005, Seaside, CA, USA, June 13-15, 2005, Proceedings*, volume 3541. Springer, 2005.

[P99] Scheltens P. Early diagnosis of dementia. *neuroimaging. J Neuro*, 246 :16–20, 1999.

[pim] `http://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes`.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, 1986.

[Qui93] J. Ross Quinlan. *C4.5 : Programs for Machine Learning.* Morgan Kaufmann, 1993.

[Rak05] Ricco Rakotomalala. Arbres de décision. 2005. `http://www-rocq.inria.fr/axis/modulad/archives/numero-33/tutorial-rakotomalala-33/rakotomalala-33-tutorial.pdf`.

[RK02] Fabio Roli and Josef Kittler, editors. *Multiple Classifier Systems, Third International Workshop, MCS 2002, Cagliari, Italy, June 24-26, 2002, Proceedings*, volume 2364. Springer, 2002.

[RKW04] Fabio Roli, Josef Kittler, and Terry Windeatt, editors. *Multiple Classifier Systems, 5th International Workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004, Proceedings*, volume 3077. Springer, 2004.

[RS04] Marko Robnik-Sikonja. Improving random forests. *Machine Learning ECML 2004*, pages 359–370, 2004.

[RSCK03] Marko Robnik-Sikonja, David Cukjati, and Igor Kononenko. Comprehensible evaluation of prognostic factors and prediction of wound healing. *Artificial Intelligence in Medicine*, 29 :25–38, 2003.

[SH] Gunter S and Bunke H. Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm. *Electron Lett Comput Vis Image Anal 2004*, page 3 :25–41.

[sls09] simon.bernard, laurent.heutte, and sebastien.adam. On the selection of decision trees in random forests. *international joint conference on Neural Networks*, pages 790–795, 2009.

[TFAM10] Evanthia E. Tripoliti, Dimitrios I. Fotiadis, Maria Argyropoulou, and George Manis. A six stage approach for the diagnosis of the alzheimer's disease based on fmri data. *Journal of Biomedical Informatics*, 43(2) :307–320, 2010.

[TPC06] Alexey Tsymbal, Mykola Pechenizkiy, and Padraig Cunningham. Dynamic integration with random forests. In *ECML*, pages 801–808, 2006.

[WM97] D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *J. Artif. Intell. Res. (JAIR)*, 6 :1–34, 1997.

[WR03] Terry Windeatt and Fabio Roli, editors. *Multiple Classifier Systems, 4th International Workshop, MCS 2003, Guilford, UK, June 11-13, 2003, Proceedings*, volume 2709. Springer, 2003.

[Zar98] Jerrold H. Zar. *Biostatistical Analysis (4th Edition)*. New Jersey, 1998.

[ZB12] Manel Zribi and Younes Boujelbene. Les réseaux de neurones un outil de sélection de variables : Le cas des facteurs de risque de la maladie du cancer du sein. *Faculté des Sciences Economiques et de Gestion, Université de Sfax, Tunisie,Unité de Recherche en Economie Appliqué*, 9(1), 2012.