Table de matière

Remerciements……... I

Résumé……II

Table de matières……...III

Liste des figures……....IV

Liste de tableaux……V

Glossaire……..VII

Introduction générale.....1

CHAPITRE I : Etude des méthodes ensemblistes

I.1. Introduction……3

I.2. Les travaux dans le domaine……...3

I.2.1 Taxonomies des méthodes de combinaison……9

I.2.2 L'apprentissage automatique ……10

I.2.2.1 Principe de la classification…..…….……...10

I.2.2.2 Principe de la classification binaire……..11

I.2.2.3 principe de la classification multi-class…….....11

I.2.2.4 Classification binaire vs Multi-class……..11

I.3. Présentation des méthodes ensemblistes……...11

I.3.1 Définition du Classifieur……11

I.3.2 Les sorties d'un classifieur……...12

I.3.3 Mesures de performances d'un classifieur……...13

I.4 L'intérêt de la combinaison des classifieurs……14

I.4.1 Définition de combinaison……...14

I.4.2 Les types de combinaison des classifieurs……...15

I.5 Amélioration de la précision……16

I.5.1 Le compromis entre le biais et la variance……..17

I.6 La manière de combiner les classifieurs……18

I.6.1 Solution……...18

I.7 Optimisation d'ensembles de classifieurs ……18

I.8 Méthodes ensemblistes……...19

I.8.1 Les types……...20

I.8.1.1 Ensemble de classifieurs homogènes……..20

I.8.1.2 Ensemble de classifieurs hétérogènes…….23

I.9 Conclusion……..24

CHAPITRE II : Principe des méthodes utilisées

II.1 Introduction……..25

II.2 Approche ensembliste médicale……..25

II.3 Ensemble de classifieurs homogènes……..25

II.3.1 Bagging……26

II.3.2 Boosting……...28

II.3.3 Les forêts aléatoires……31

II.3.3.1 Les arbres de décision……31

II.3.3.2 Principe des forêts aléatoires……..32

II.4 La différence entre les trois classifieurs…….34

II.4.1 Étude comparative entre Bagging et Boosting……..36

II.4.2 Avantages et inconvénients……..36

II.5 Ensemble de classifieurs hétérogènes…….37

II.5.1 Vote majoritaire……..37

II.5.2 Vote pondéré…….38

II.5.3 Majorité pondérée…….38

II.6 Conclusion……..38

CHAPITRE III : Expérimentations et Résultats

III.1Introduction…….....39

III.2 Bases de données…….39

III.2.1 Colon ……39

III.2.2 breast cancer-wisconsin…….39

III.2.3 Pendigits ……....40

III.2.4 Heart ……..40

III.2.5 Hepatite……..40

III.3 Matériels et méthodes ……40

III.4 Mesures de performance……41

III.5 Expérimentation 1……42

III.5.1. Les résultats de bagging au niveau des bases de données……42

III.5.1.2 Interprétation des résultats concernant le bagging…….....47

III.5.2 Les résultats de boosting « AdaBoostM1 » au niveau des bases de données.48

III.5.2.1 Interprétation des résultats concernant le boosting « AdaBoostM1 »..51

III.5.3 Les résultats de random forest au niveau des bases de données……52

III. 5.3.1 Interprétation des résultats……...55

III.6. Comparaison des résultats……...56

III.6.1. Interprétation général des résultats……..…57

III.7. L'exécution des trois méthodes d'ensembles sur les 5 bases de données dans le même temps…….57

III.7.1 L'expérimentation 02…….57

III.8. Interprétation des résultats……..58

III.9 Conclusion…….59

Conclusion générale……...60

Bibliographie……...61

Liste des figures

FIGURE I.1 Approche séquentielle……..15

FIGURE I.2 Approche parallèle……16

FIGURE I.3 Approche hybride……...16

FIGURE I.4 illustration graphique de biais et variance……18

FIGURE I.5 Principe général des méthodes d'ensemble……..20

FIGURE I.6 Les type de méthodes ensemblistes -1-……...21

FIGURE I.7 Les type de méthodes ensemblistes -2……....22

FIGURE I.8 Illustration schématique des trois méthodes d'ensemble homogènes populaires……..23

FIGURE II.1 Principe de Bagging……27

FIGURE II.2 Principe de Bootstrapping……...28

FIGURE II.3 Principe de Boosting……...30

FIGURE II.4 Random Forest vs Bagging.....35

FIGURE III.1 résultats obtenus pour la BDD colon pour la méthode bagging ……44

FIGURE III.2 résultats obtenus pour la BDD hepatite pour la méthode bagging……...45

FIGURE III.3 résultats obtenus pour la BDD pendigits pour la méthode bagging…….45

FIGURE III.4 résultats obtenus pour la BDD heart pour la méthode bagging …….46

FIGURE III.5 résultats obtenus pour la BDD Wisconsin pour la méthode bagging…..……46

FIGURE III.6 résultats obtenus pour la BDD Wisconsin pour la méthode boosting……....49

FIGURE III.7 résultats obtenus pour la BDD Colon pour la méthode boosting ……...49

FIGURE III.8 résultats obtenus pour la BDD pendigits pour la méthode boosting……50

FIGURE III.9 résultats obtenus pour la BDD hepatite pour la méthode boosting…….50

FIGURE III.10 résultats obtenus pour la BDD heart pour la méthode boosting……51

FIGURE III.11 résultats obtenus pour la BDD Colon pour la méthode RF……53

FIGURE III.12 résultats obtenus pour la BDD Wisconsin pour la méthode RF …….54

FIGURE III.13 résultats obtenus pour la BDD pendigits pour la méthode RF …….54

FIGURE III.14 résultats obtenus pour la BDD hepatite pour la méthode RF ……...55

FIGURE III.15 résultats obtenus pour la BDD heart pour la méthode RF ……55

FIGURE III.16 résultats obtenus pour la comparaison des résultats pour les trois méthodes…….56

Liste des tableaux

TABLE I.1 Les travaux dans le domaine des méthodes ensemblistes……8

TABLE II.1 Représentation d'état de l'art des méthodes ensembliste-1-…….33

TABLE II.2 Représentation d'état de l'art des méthodes ensembliste-2-…….33

TABLE II.3 comparaison entre les trois classifieurs…….35

TABLE II.4 Bagging vs Boosting.....36

TABLE II.5 étude comparative entre les avantages et les inconvénients des méthodes ensemblistes……...37

TABLE III.1 concernant la description des bases de données……..42

TABLE III.2 Les résultats obtenus pour les cinq BDD pour la méthode Bagging……..44

TABLE III.3 Les résultats obtenus pour les cinq BDD pour la méthode Boosting…….49

TABLE III.4 Les résultats obtenus pour les cinq BDD pour la méthode Random forest……53

TABLE III.5 Comparaison des résultats……...…56

TABLE III.6 Comparaison des résultats……..58

Glossaire

BAG: Bagging

BDD: Base de Donnée

BOOST: Boosting

BT: AdaboostM1

IA : Intelligence Artificiel

ME : Méthodes Ensembliste

RF: Random Forest

Se : Sensibilité

Sp : Spécificité

SVM : Support Vector Machine

Tc : Taux de Classification

TE : taux d'erreur

VN : Vrai Négatif

VP : Vrai Positif

Rapport-Gratuit.com

Introduction générale

Une décision précise et robuste est prise en générale d'une manière consensuelle par un ensemble de décideurs plutôt que de la laisser à une seule personne. Le fait de charger un groupe de décideur plutôt que dans un décideur unique se justifie de deux manières différentes. Une mauvaise décision individuelle se traduit en général par de lourdes conséquences, on cherche donc à obtenir la décision la plus objective possible en ayant recours à des groupes de décideurs pour avoir plusieurs points de vue. La seconde raison pour laquelle on a également recours à un comité de décideurs est que les différents intervenants peuvent n'avoir qu'une vision partielle du problème posé. C'est le cas par exemple d'un groupe de médecins, où les différents membres présents ne sont concernés par q'une ou deux spécialistes. L'accord sur le diagnostic est alors décidé à la majorité parmi les différents intervenants.

Actuellement, l'Apprentissage Automatique tend à s'inspirer de cette approche pour rechercher des algorithmes permettant de construire des ensembles de classifieurs (Ensemble Learning, Commitee Learning). L'objectif est d'obtenir un nouveau classifieur, constitué d'un ensemble de prédicteurs de base, de manière à diminuer le nombre d'exemples mal classés tout en conservant un temps de calcul raisonnable.

La littérature est très riche et diverse. L'idée de combiner les sorties des classifieurs pour créer un système avec une fiabilité élevée n est pas nouvelle ; les chercheurs attribuent la première application de combinaison des réseaux de neurones à Nilsson. Le véritable intérêt pour la combinaison de classifieurs date en fait des années 80, avec en particulier les travaux de [55] qui ont montré l' importance d'avoir des solutions robustes pour les problèmes de reconnaissance de l écriture manuscrite en particulier En 1989, Clemen citait déjà plus de 200 travaux reliés à la combinaison de classifieurs [54]. Cependant, ce n est qu'à partir des années 90 que les systèmes de combinaison de classifieurs ont été concrètement mis en œuvre. L'étude de ces techniques est d'un grand intérêt pour obtenir des performances supérieures à celles d'un seul classifieur.

Parlons de la classification avec 1000, voire 10000 classes alors que les problèmes multi-classes classiques ne dépassent pas habituellement la centaine de classes. Aujourd'hui, il existe même des problèmes dépassant le million de catégories. Il n'est pas difficile d'imaginer des tâches pouvant atteindre le milliard de catégories dans le cas de reconnaissance de visages humains par exemple. Donc l'émergence des bases de données actuelles et leur croissance exponentielle contribue à la construction des masses de données qui dépassent de loin les capacités humaines à les traiter.ces données sont la source d'information et de connaissance qui nécessitent des méthodes automatiques de synthèses et d'interprétation. Cette croissance des données

aujourd'hui génère de nouvelles problématiques pour lesquelles l'apprentissage statistique ne possède pas de réponses adaptées. Ainsi le cadre classique de la classification qui consiste à affecter une ou plusieurs classes à une instance est étendu à des problèmes avec des milliers, voire des millions de classes différentes. Avec ces problèmes viennent de nouveaux axes de recherches comme la réduction de la complexité de classification qui est habituellement linéaire en fonction du nombre de classes du problème, ce qui est problématique lorsque le nombre de classe devient trop important. Plusieurs familles de solutions pour cette problématique ont émergé comme la construction d'une hiérarchie de classifieurs ou bien l'adaptation de méthodes ensemblistes. La minimisation de l'erreur est l'une des principales motivations de la recherche en apprentissage automatique, par un cout induit par une mauvaise classification peut être élevé.ces dernières années un grand nombre des travaux en apprentissage automatique ont porté sur les méthodes ensemblistes .nous allons présenter dans ce travail les trois classifieures bagging ; forêt aléatoire ; boosting.

Organisation de notre travail

- **Chapitre 1 :** nous présentons une étude sur les méthodes d'ensemble et nous exposons l'état de l'art de ces dernières méthodes.

- **Chapitre 2 :** nous citons brièvement les méthodes ensemblistes (boosting,bagging et les forêts aléatoires) utilisées , en précisant leurs avantages et leurs inconvénients.

- **Chapitre 3 :** nous consacrons le dernier chapitre à la discussion des résultats expérimentaux obtenus.

En dernier lieu, une conclusion générale qui résume les points essentiels de ce travail avec des perspectives.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Chapitre I

Etude des méthodes ensemblistes

I.1 Introduction

Un des principaux enjeux de l'apprentissage automatique consiste à concevoir des systèmes de classification performants a partir d'un ensemble d'exemples représentatifs d'une population de données. Parmi les différentes approches permettant d'aborder ce type de problématique, combiner un ensemble de classifieurs individuels faibles pour former un unique système de classification appelé Ensemble de Classifieurs a suscité un intérêt grandissant de la communauté scientifique. Des travaux récents de recherche qui ont montré que certains principes de combinaison de classifieurs sont particulièrement efficaces, tel que le Boosting [1] (ou Arcing [2]), le Bagging [3], le Random Subspaces [4], ou plus récemment les Random Forests [5]. L'efficacité des combinaisons de classifieurs repose principalement sur leur capacité à tirer parti des complémentarités des classifieurs individuels, dans le but d'améliorer autant que possible performances en généralisation de l'ensemble.

Dans ce chapitre, nous présentons quelques notions utilisés à la compréhension de la combinaison des classifieures. Ce chapitre est scindé en 04 sections :

- Les travaux dans le domaine et les taxonomies des méthodes ensemblistes.
- Présentation des méthodes ensemblistes
- L'intérêt de la combinaison des classifieurs
- Les méthodes ensemblistes homogènes et hétérogènes.

I.2 Les travaux dans le domaine

Le tableau ci-dessus résume brièvement quelques travaux des méthodes ensemblistes citées dans la littérature :

Auteur	Titre	Resultants et experiences
Sami FAYE-CHELLALI [6]	Optimisation du processus prédictif de sélection médicale en prévoyance individuelle Application au télémarketing	Les forêts aléatoires semblent proposer le meilleur équilibre entre la performance chiffrée sur les données et la simplicité du modèle. Mortality slippage de 100.5% Taux de faux négatifs de 0.75% alors que le boosting est très bon pour les propriétés théoriques (100.5% pour le traitement des classes plus risqués)

Philippe BESSE, Hélène MILHEM, Olivier MESTRE, Anne DUFOUR, Vincent-Henri PEUCH [7]	Comparaison de techniques de « Data Mining » pour l'adaptation statistique des prévisions d'ozone du modèle de chimie-transport MOCAGE	Cette étude, réalisée sur cinq sites, illustre l'intérêt des techniques d'adaptation statistique qui permettent d'améliorer à moindre coût les prévisions déterministes du modèle MOCAGE. Dans notre étude de cas, on réduit l'écart-type d'erreur de prévision de 36 µg/m3 à 26 µg/m3 pour les meilleures méthodes. L'analyse de covariance avec inter - actions et la méthode des forêts aléatoires donnant d'excellents résultats, elles devront être privilégiées pour réaliser l'adaptation statistique sur l'ensemble des postes.
Thomas François Aspirant FNRS [8]	Modèles statistiques pour l'estimation automatique de la difficulté de textes de FLE	les trois classifieurs présentent des résultats qui ne sont pas significativement différents. Il est donc impossible de conclure à la supériorité de l'un des modèles statistiques, même si la méthode qui semble se comporter le mieux est la RLM. Sachant de plus que le temps nécessaire à l'entraînement d'un tel modèle est considérablement plus réduit que celui pour le bagging et surtout le boosting, il nous semble que la RLM constitue le meilleur choix aussi bien au niveau de l'optimisation du temps de calcul qu'au niveau de l'efficacité du

		modèle. Cette conclusion concorde avec nos résultats précédents (François, 2009).
Mohamed aymen fodda [9]	Detection et reconnaissance de visage	La méthode viola jones est la meilleure pour la reconnaissance faciale ; le seuil pour laquelle cet algorithme détermine la ressemblance d'une zone de l'image aux descripteurs de haar est détermine au cours de l'apprentissage par la méthode adaboost L'algorithme discrets adaboost [freund 1996] permet d'éliminer 50% des fausses detections mais il élimine 0.2 % des modèles frontaux de visage
Ritty Jean-François Elève Mécatronique 2013 [10]	Détection de piétons en milieu routier urbain	Méthodes discriminatives basées sur l'apparence des piétons utilisant : • Descripteurs : HOG, somme de gradient, … • Classificateurs : SVM, AdaBoost, … Base de données Kitti : Description : • Situations réelles et variées • Grand nombre d'objets présents Données fournies : • Calibrage caméra • Labellisation des objets • Résultats relativement faibles qui peuvent provenir des vidéos d'évaluations et qui sont légèrement améliorés par la segmentation. Modèles par partie difficilement

		utilisables en temps réel.
Chesner Desir [10]	Classification Automatique d'Images, Application a l'Imagerie du Poumon Profond	Les résultats des extra-trees sont compares a ceux obtenus par forest-RI dans le tableau 2.10. Pour ces deux méthodes de forets aléatoires, le temps moyen de calcul par arbre est également indique. Nous observons que les deux méthodes présentent des resultats tr es proches, avec un avantage en moyenne pour les extra-trees. La différence essentielle réside dans les couts de calculs : les extra-trees sont deux fois plus rapides que les forest-RI.
Henri-Maxime Suchier [11]	Nouvelles contributions du boosting en apprentissage automatique	Pour tous les niveaux de bruit, et en moyenne sur les 10 bases, le taux de succès d'OraBoost au bout de 1000 itérations est plus élevé que celui d'AdaBoost
Alexis LECHERVY [12]	Apprentissage interactif et multi-classes pour la détection de concepts sémantiques dans des données multimédia	Leur méthode construit donc un nouveau noyau par combinaison de noyaux faibles selon un processus inspiré des méthodes de Boosting Les différentes expériences réalisées ont permis de montrer la pertinence de notre modèle. leur méthode obtient de meilleurs résultats que les méthodes concurrentes de l'état de l'art.
Xin Li [13]	Boosting one of combining models	Adaboost fonctionne avec un modèle unifié CPM; 1) Ce modèle a lissé l'ensemble de données

		formées et réduit l'influence de l'ajustement excessif. 2) L'influence de l'historique est très intéressante. Cela supprimera le bruit et renforcera la direction de renforcement WeakLearn. 3) La longueur d'étape de KNN sélectionnée par Adaboost n'est pas discutée ici. Ceci est également utile pour supprimer le bruit. Adaboost ajuste adaptablement les erreurs des hypothèses faibles par WeakLearn. Contrairement à l'algorithme de stimulation conventionnel, l'erreur précédente ne doit pas être connue à l'avance. La règle de mise à jour réduit la probabilité attribuée aux exemples sur lesquels l'hypothèse fait de bonnes prédictions et augmente la probabilité des exemples sur lesquels la prédiction est médiocre.
Rimah Amami, 2 Dorra Ben Ayed, 3 Noureddine Ellouze [17]	Application de la Méthode Adaboost à la Reconnaissance Automatique de la Parole	Adaboost est une nouvelle méthode de Boosting principalement utilisée pour stimuler les performances des algorithmes d'apprentissage. Depuis son apparition jusqu'à aujourd'hui, il y a eu la naissance de plusieurs générations. Pours nos

		expérimentations, nous avons étudié différentes méthodes d'Adaboost et nous avons proposé d'évaluer dans ce travail l'approche Real Adaboost (RAB), l'approche Gentle Adaboost (GAB) et l'approche Modest Adaboost . Nous avons mis en œuvre un système de reconnaissance en se basant sur l'algorithme d'apprentissage CART. En occurrence, cet algorithme a été fusionné avec les différentes méthodes d'Adaboost dans le but de réduire les taux d'erreurs de la reconnaissance. Nous avons observé, également, que l'approche Modest Adaboost a généré de meilleures performances suivie par le système basé sur l'approche Gentle Adaboost et enfin par le système basé sur l'approche Real Adaboost en appliquant les coefficients MFCC+Delta+Delta-Delta. Les taux trouvés sont de l'ordre de 88,02% pour Modest Adaboost, 85,8% pour Gentle Adaboost et 85% pour Real Adaboost.

TABLE I.1 Les travaux dans le domaine des méthodes ensemblistes

I.2.1 Taxonomies des méthodes de combinaison

a. Différentes taxonomies des méthodes de combinaison ont été proposées dans la littérature [18] et [19]. Suivant les auteurs, les taxonomies mettent en avant le type de sorties des classifieurs combines, la capacité d'apprentissage des méthodes de combinaison ou les stratégies de combinaison choisies.

b. Dans [20], les méthodes qui fusionnent des classifieurs de type mesure sont divisées en deux groupes selon la quantité d'information à exploiter pour produire la réponse finale. Les méthodes appelées class conscious comme la règle du produit ou la moyenne. Les méthodes appelées class-indifférent utilisent toutes les informations des classifieurs dans la prise de décision finale.

c. Dans [21], les méthodes de type rang ont été divisées en deux groupes. Le premier groupe de méthodes est base sur la réduction de l'ensemble des classes. Son objectif est de réduire l'ensemble de classes de départ mais en s'assurant que la vraie classe existe toujours dans le sous-ensemble final réduit. Le deuxième groupe de méthodes consiste à réordonner les classes de telle façon que la vraie classe soit rangée dans les premières propositions de la liste de solutions. Les méthodes de type rang sont divisées selon la taxonomie de Ho [21] ; deux approches sont possibles: celles permettant de réduire l'ensemble des classes et les méthodes permettant de réordonner les classes.

d. Dans [22], une autre taxonomie des methodes de combinaison de classifieurs est proposée. Cette taxonomie est divisée en deux niveaux. Dans le premier niveau on trouve les méthodes de sélection dont le rôle est de choisir les classifieurs en fonction de certains critères (le taux de reconnaissance de l'ensemble, la corrélation entre les sorties des classifieurs,...). Les sorties de ces classifieurs sont fusionnées par l'une des méthodes du deuxième niveau. Ces méthodes de fusion sont classées en fonction du type des sorties de classifieurs.

e. Moobed [23] distingue plutôt les méthodes de combinaison de type rang par le type de résultats qu'elles fournissent (type ensemble, type rang ou type mesure). Alors que les taxonomies se distinguent généralement par le type des sorties des classifieurs à combiner (propriété de classification)

f. Xu [24] distingue les méthodes de combinaison uniquement par le type de sorties des classifieurs (classe, rang, mesure) présentées en entrée de la combinaison. Le type de sorties des classifieurs est un critère qui apparait pratiquement dans toutes les taxonomies présentées dans la littérature. La prise en compte de ces niveaux différents d'information influence directement la complexité des méthodes de combinaison développées.

g. Jain [19] construit une taxonomie suivant trois critères d'égale importance : le type de sorties des classifieurs, l'adaptabilité des méthodes de combinaison (capacité à

exploiter l'expertise des classifieurs) et leur capacité d'apprentissage. Ce dernier critère est aussi utilise par d'autres chercheurs [25] pour séparer les méthodes de fusion.

h. Duin [18] distingue, quant à lui, dans les méthodes de fusion, les méthodes de combinaison de classifieurs hétérogènes (différents) des méthodes de combinaison de classifieurs faibles (homogènes). l'intérêt des méthodes de combinaison de classifieurs faibles réside dans la combinaison de classifieurs ayant la même structure mais entrainés sur des données différentes ou initialises de manière différente.

i. D'autres critères comme la nature des classifieurs sont aussi utilises [26]. Selon [26], les informations utilisées par les méthodes de combinaison parallèle sont liées au type de classifieurs à combiner et à leur réponse individuelle. Cette réponse peut être une classe, une probabilité, une confiance, une connaissance ou une information floue.

j. Dans les méthodes de fusion, on distingue les méthodes dites figées ou non-paramétriques (les sorties des classifieurs sont combinées dans un schéma dont les paramètres sont invariables) des méthodes avec apprentissage qui cherchent à apprendre, sur les données disponibles, les paramètres nécessaires a la combinaison. Enfin, la complexité de ces méthodes peut varier en fonction du niveau d'information associe aux réponses fournies par les classifieurs à combiner (sortie de types classe, rang ou mesure).

I.2.2 L’apprentissage automatique

L’apprentissage Automatique Parmi les nombreuses disciplines représentées en informatique, l’Apprentissage Automatique, ou Machine Learning en anglais, désigne la recherche et l’étude d’algorithmes permettant à une machine de réaliser l’induction automatique de règles à partir d’un ensemble d’exemples. Ce domaine de l’intelligence artificielle est proche des statistiques, de la fouille de données et de la reconnaissance de formes. Pour résoudre des problèmes issus du monde réel, la machine doit apprendre à produire la sortie désirée lorsqu’on lui présente un vecteur d’entrées particulier. On distingue en général les problèmes de régression et les problèmes de classification. Pour les problèmes de régression, la sortie à calculer a une valeur continue. Réaliser une estimation du chiffre d’affaire d’une entreprise à partir d’un ensemble de variables concernant ses activités est une illustration possible de ce type de problèmes. Les problèmes de classification, quant à eux, cherchent à exprimer une sortie correspondant à une étiquette ou à un groupe, à valeur dans un ensemble fini. La reconnaissance de caractères manuscrits, où l’étiquette correspond alors à l’une des vingt-six lettres de l’alphabet, est l’une des nombreuses applications usuelles de ce type de problème.

I.2.2.1 Principe de la classification

La classification est une discipline qui consiste à regrouper divers objets (les individus) d’une série de caractéristiques prédéfinies par apprentissage en sous-ensembles d'objets (les classes).

I.2.2.2 Principe de la classification binaire

Chaque point de formation appartient à l'un des 2 différentes classes. L'objectif est de construire une fonction qui, compte tenu d'un nouveau Point de données, prédit correctement la classe à laquelle le Un nouveau point appartient.

I.2.2.3 Principe de la classification Multi-class

Il existe de nombreux scénarios dans lesquels il existe plusieurs classes aux quels les points appartiennent, ce problème se décompose trivialement en un ensemble de problèmes binaires non liés, qui peut être résolu naturellement en utilisant nos techniques de classification binaire.

I.2.2.4 Classification binaire vs Multi class

Dans la plupart des cours d'initiation à l'apprentissage automatique, les classificateurs binaires sont souvent le foyer en raison de leur présentation plus simple. Cependant, de nombreux problèmes ont en soi plus de deux résultats possibles. Par exemple, on veut peut-être former un système de vérification du visage qui peut détecter l'identité d'une photo d'un groupe de personnes. Ce type de problème est connu comme un problème de classification «multi-class».

La plupart des définitions mathématiques des classiffieurs sont initialement posées comme classificateurs binaires, y compris le SVM. Ainsi, il existe deux approches principales du problème multiclass: Ajoutez directement une extension multiclass à un classiffieur binaire.

I.3 Présentation des méthodes ensemblistes

. Concernant la combinaison de classifieurs, la littérature est très riche et diverse. L'idée de combiner les sorties des classifieurs pour créer un système avec une fiabilité élevée n'est pas nouvelle. Lorsqu'il faut prendre une décision importante, il vaut souvent mieux recueillir plusieurs avis que de se fier à un seul .Utiliser un modèle pour prédire un comportement, c'est un premier pas. Mais agréger des milliers de modèles ayant des avis divergents mais pouvant être chacun spécialisés sur des parties de la data donne le plus souvent de meilleurs résultats. Nous parlons alors de méthodes ensemblistes, dont les plus connues sont le bagging et le boosting. Pour faire une étude sur les méthodes ensemblistes il est bien nécessaire de rappeler ce qu'on entend généralement par classifieur dans le cadre de la combinaison et pourquoi et comment combiner des classifieurs.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

I.3.1 Définition du Classifieur

Le classifieur est défini comme étant tout système de traitement de données qui reçoit une forme x et donne des informations à propos de la classe correspondant à cette forme. Quel que soit le domaine d'application pour lequel il est utilisé, la mise en œuvre de tout classifieur nécessite de choisir tout d'abord une représentation pour décrire les données (caractéristiques), un algorithme de décision et une base d'apprentissage permettant de fixer les paramètres du classifieur. Lorsqu' il est intégré dans un système de reconnaissance, le classifieur atteint certaines performances. Introduire des modifications dans le classifieur que ce soit au niveau des données qu'il traite ou au niveau de ses paramètres (type de sorties, règles de décision,...) modifie ses performances. [27]

Le classifieur a pour mission d'affecter une étiquette de classe à l'objet qui lui est présenté par l'extracteur de caractéristiques. Un avantage pratique de la séparation entre l'extraction de caractéristiques et la classification est que de cette façon, il est possible d'étudier les classifieurs hors de leur domaine d'application spécifique ; ceci grâce au niveau d'abstraction suffisamment élevé produit par la représentation des objets sous forme de vecteurs de caractéristiques. [27]

L'objectif principal du classifieur consiste à séparer l'espace de représentation par des frontières et à assigner des étiquettes de classe aux régions ainsi formées. La région où se trouve un vecteur de caractéristiques détermine donc sa classe d'appartenance. Généralement, il n'est pas possible de bâtir une partition parfaite de l'espace, aussi le rôle du classifieur sera souvent de donner une probabilité d'appartenance d'un objet à une classe.

Dans le cadre de la combinaison, nous proposons de définir un classifieur comme étant un système de reconnaissance qui travaille dans un certain espace de caractéristiques, qui utilise une certaine base pour apprendre ses paramètres, qui prend sa décision à partir d'une certaine règle et qui fournit en sortie un certain type de réponse. D'après cette définition, deux classifieurs qui diffèrent par au moins l'un de ces paramètres (espace de caractéristiques, données d'apprentissage, règle de décision et type de sortie) ne donnent pas les mêmes résultats et sont donc considérés comme différents [27]. Si on fait abstraction de son architecture interne, le classifieur e_j est vu comme une boîte noire qui reçoit en entrée un ensemble de caractéristiques et qui fournit en sortie une réponse à propos de la forme à reconnaître (on se place ici dans l'hypothèse d une classification exclusive avec laquelle un classifieur e(x) sera d' autant meilleur que sa réponse se rapprochera de la vraie classe). Selon Xu [24], cette réponse peut être divisée en trois catégories suivant le niveau d'information apporté par le classifieur, catégorisation également adoptée dans la majorité des travaux [22].

I.3.2 Les sorties d'un classifieur

Si l'on fait abstraction de l'architecture interne d'un classifieur, le classifieur e_j est vu comme une boite noire qui reçoit en entrée un ensemble de caractéristiques et qui fournit en sortie une réponse à propos de la forme à reconnaitre. Cette réponse peut être divisée en trois catégories suivant le niveau d'information apporte par le classifieur :

- **Type classe :**

$$e_j(x) = C_i,\ i \in \{1,\ldots N\} \qquad (I.1)$$

Ci est la classe attribuée par le classifieur e_j à x. Dans ce cas, l'avis du classifieur est binaire. On peut alors représenter la réponse du classifieur par un vecteur binaire dans lequel '1' indique la classe proposée par le classifieur. Un classiffieur peut aussi produire un ensemble de classes. Il considère alors que la forme x appartient à une des classes de cet ensemble sans donner d'autres informations permettant de discriminer les classes. [27]

- **Type rang :**

$$e_j(x) = (r_{1,j}, r_{2,j} \ldots\ldots, r_{N,j}) \qquad (I.2)$$

$r_{i;j}$ est le rang attribue à la classe C_i par le classifieur e_j. Il s'agit d'un classement sur les classes. Le classifieur indique ce classement en fournissant en sortie un vecteur des rangs de taille N. La classe placée au premier rang de la liste proposée par le classifieur est considérée comme la plus probable pour la forme x et la classe du dernier rang est la moins probable. [27]

- **Type mesure :**

$$e_j(x) = (m_{1,j}, m_{2,j}, \ldots\ldots, m_{N,j}) \qquad (I.3)$$

$m_{i;j}$ est la mesure attribuée à la classe i par le classifieur e_j . Elle indique le niveau de confiance du classifieur dans sa proposition. La sortie du classifeur est donc un vecteur de mesures de taille N. Cette mesure, normalisé ou non, peut être une distance, une probabilité a posteriori, une valeur de confiance, un score, une fonction de croyance, une possibilité, une crédibilité, une mesure floue, etc. [27].

I.3.3 Mesures de performances d'un classifieur

Pour une entrée donnée, un classifieur peut générer les réponses suivantes :

- **Un rejet** : pour indiquer que le classifieur n'a pas pu identifier cette entrée.
- **Une reconnaissance** : dans ce cas, il identifie bien l'entrée, et il lui attribue sa classe appropriée.
- **Une substitution** : le classifieur attribue une autre classe à l'entrée.

La performance d'un classifieur peut être mesurée en calculant les trois taux suivant :

- ✓ **taux de rejet** = Nombre de formes rejetées/ Nombre total de formes (1)
- ✓ **taux de reconnaissance** = Nombre de formes reconnues/ Nombre total de formes (2)
- ✓ **taux de substitution** = Nombre de formes males reconnues/Nombre total de formes (3)

I.4 L'intérêt de la combinaison des classifieurs

L'idée principale derrière la combinaison de classifieurs est l'augmentation de la qualité des résultats. En effet, de plus en plus, les chercheurs se rendent compte qu'il n'existe aucune méthode de classification pouvant manifester une supériorité sur les autres méthodes dans tous les problèmes et toutes les situations [17]

Si aucune d'entre elles ne peut satisfaire entièrement les exigences d'une application envisagée, l'utilisation simultanée de plusieurs méthodes en même temps peut, éventuellement, permettre d'en cumuler les avantages sans en cumuler les inconvénients. Différents schémas de combinaison de classifieurs ont été proposés dans la littérature selon les critères de combinaison [27]

Ainsi, on trouve le critère de l'ordre d'exécution des classifieurs où on observe trois types de combinaison : séquentielle, parallèle et série parallèle

I.4.1 Définition de combinaison

La combinaison des classifieurs ne dépend pas seulement de la fonction de fusion, mais aussi de la topologie de ces derniers. Lam a proposé la classification suivante pour décrire les principales topologies utilisées.

– **Conditionnelle** Dans ce type d'organisation, un premier système de classification prend la décision de classer ou de rejeter un objet. Son rôle est de classer les objets les plus faciles à reconnaître et de laisser aux autres étages, plus complexes, les objets difficiles. C'est une façon de minimiser l'utilisation des ressources ou de réduire le temps moyen de reconnaissance d'un système.

– **Sérielle** Il s'agit ici de construire un système où les classifieurs sont utilisés les uns à la suite des autres, chacun réduisant l'ensemble de classes possibles pour un objet à classer.
– **Parallèle** L'organisation parallèle des experts d'un ensemble de classifieurs implique que chaque classifieur individuel produise une sortie simultanée. L'ensemble de ces sorties est ensuite fusionné de façon à produire une décision unique. Avant de rentrer dans les détails de la problématique de la combinaison parallèle de classifieurs, il est nécessaire de rappeler ce qu'on entend généralement par classifieur dans le cadre de la combinaison et de présenter les différentes stratégies de combinaison de classifieurs.

I.4.2 Les type de combinaison des classifieurs

La multiplication des travaux sur la combinaison a entrainé la mise au point de nombreux schémas traitant les données de manières différentes [26]. Trois approches pour la combinaison de classifeurs peuvent être envisagées: parallèle, séquentielle et hybride. D'autres organisations « avec bouclage » ou « avec interaction » sont aussi possibles .Mais malgré la diversité des schémas de combinaison, la détermination de la meilleure organisation reste un problème ouvert.

-Approche séquentielle

La combinaison séquentielle, appelée également combinaison série, est organisé en niveaux successifs de décision permettant de réduire progressivement le nombre de classes possibles. Dans chaque niveau, il existe un seul classifieur qui prend en compte les réponses fournies par le classifieur placé en amont à fin de traiter les rejets où confirmer la décision obtenue sur la forme qui lui est présentée dans cette figure :

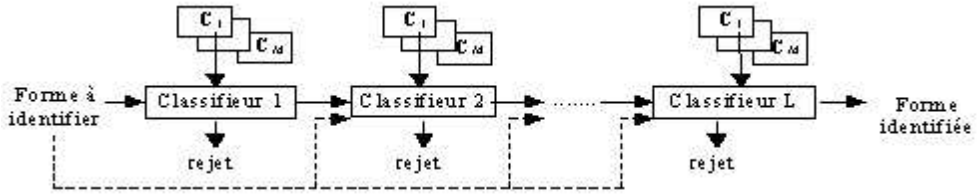


FIGURE I.1 Approche séquentielle

-Approche parallèle

A la différence de l'approche séquentielle, l'approche parallèle laisse dans un premier temps les différents classifieurs opérer indépendamment les uns des autres puis fusionne leurs réponses respectives. Cette fusion est faite soit de manière démocratique, dans le sens où elle ne favorise aucun classifieur par rapport à un autre, soit au contraire dirigé et, dans ce cas, on attribue aux réponses de chaque classifieur un poids en fonction de ses performances. L'ordre d'exécution des classifieurs n'intervient pas dans cette approche.

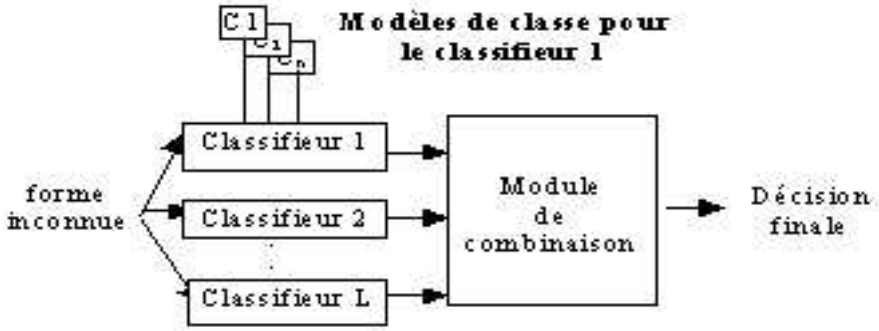


FIGURE I.2 Approche parallèle

Approche hybride

L'approche hybride consiste à combiner à la fois des architectures séquentielles et parallèles à fin de tirer pleinement avantage de chacun des classifieurs utilisés. Ce type d'approche permet de générer de nombreux schémas de coopération qui peuvent rapidement devenir complexes à optimiser.

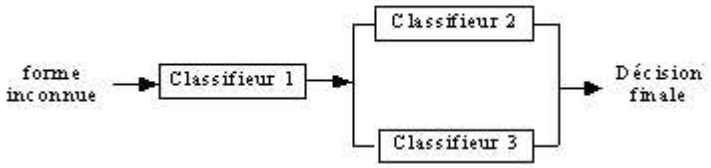


FIGURE I.3 Approche hybride

I.5 Amélioration de la précision

Pourquoi un ensemble de classifieurs est plus précis qu'un classifieur seul ?selon Dietterich il existe trois raisons :

La première est statique : puisque les classifieurs sont divers, ils font des erreurs à des différents endroits et prendre une moyenne et un vote de leurs décisions individuelles c'est à dire diminue le risque de prendre la mauvaise décision. Ceci se produit d'avantage lorsque la proportion d'observation est trop faible par rapport au nombre de variables explicatives.

Pour comprendre les deux autres raisons il faut souligner le fait que tout modèle peut être décrit par une fonction, mais que cette fonction n'est pas toujours facile à trouver. Dans le cas où la fonction est trop difficile à trouver, des hypothèses

approximant la fonction sont utilisées .ces hypothèses sont à la base des deux autres raisons .la première entre elle est informationnelle.

La raison informationnelle :plusieurs algorithmes fonctionnent en optimisant une recherche locale , qui peut se retrouvé coincée dans un optimum local ,entre autres lors de la recherche du meilleur embranchement .dans le cas où le nombre de donné important ,il peut être difficile pour l'algorithme de trouver la meilleur hypothèse .un ensemble dont les différents éléments sont construits en commençant la recherche locale à partir de plusieurs point différents fournira au bout du compte ,une meilleur approximation que tous les classifieurs pris séparément .

La dernière raison est représentationnelle : la fonction qui décrit exactement les données ne peut pas être représentée par les hypothèses fournies par les classifieurs.

L'utilisation des méthodes de combinaison des classifieurs garantie :

- La précision : des 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 divisé pour régner).

I.5.1 Le compromis entre le biais et la variance

La variance est un paramètre qui a été développé par Ronald Fisher en 1918[28] .Il combine toutes les valeurs dans une base de données pour obtenir une mesure de dispersion. Elle est symbolisée par (S2). La variance et l'écart-type sont les plus couramment utilisés.

L'erreur en généralisation peut être décomposée en deux termes : biais et variance. Ces deux types d'erreur sont liés au langage choisi.

- Soit celui-ci est trop pauvre pour contenir le concept cible et, dans ce cas, l'apprentissage revient à chercher la meilleure approximation de la cible dans le langage. L'erreur de l'approximation finalement trouvée sera due au biais (de langage).
- Soit le concept cible peut être représenté dans le langage des hypothèses mais, cette fois, la richesse du langage fait qu'il est probable que d'autres hypothèses aient le même comportement que la cible sur les données disponibles. L'algorithme d'apprentissage doit choisir parmi ces candidats équivalents, au hasard. L'erreur en généralisation de l'hypothèse choisie sera due à la variance.

Dans la pratique des algorithmes, il est important de construire des modèles parcimonieux (optimaux) quelle que soit la méthode utilisée. Toutes les méthodes algorithmiques sont concernées : que ce soit le nombre de variables explicatives, le nombre de feuilles dans un arbre, ou de neurones dans une couche cachée. Plus

un modèle est complexe et donc plus il intègre de paramètres, et plus il est flexible donc capable de s'ajuster aux données engendrant ainsi une erreur faible d'ajustement. En revanche, un tel modèle peut s'avérer défaillant lorsqu'il s'agira de prévoir ou de généraliser, c'est-à-dire de s'appliquer à des données qui n'ont pas pris part à sa construction. L'objectif consiste donc à trouver le meilleur compromis entre les deux situations : le biais et la variance en contrôlant l'ajustement aux données et la complexité du modèle. On acceptera un certain biais afin de diminuer la variance du modèle ou inversement selon le but recherché. L'idée est qu'il n'y a pas de « méthode universelle » pour tous les problèmes, mais chaque méthode est plus ou moins bien adaptée au problème posé, à la nature des données.

Pour illustrer la notion de « biais – variance » nous utiliserons le schéma ci-dessous :

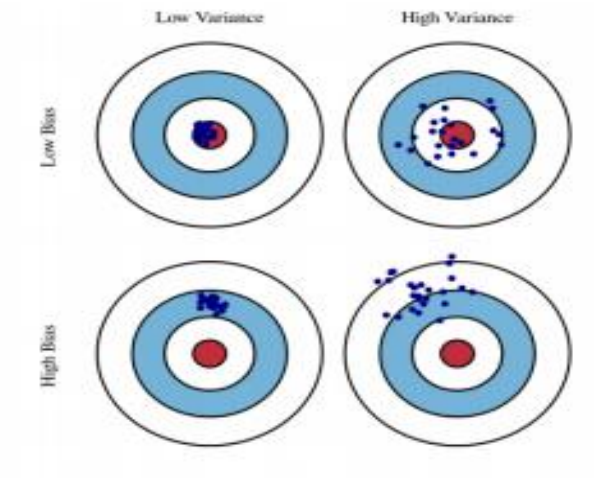


FIGURE I.4 illustration graphique de biais et variance

La cible en haut à gauche est la situation idéale, mais peu réaliste « faire mouche à chaque coup ». La 3e cible représente quant à elle une faible variance (les points sont regroupés) et d'un fort biais (les points sont loin du centre rouge). Sur la 4e cible, les points sont dispersés et loin du centre de la cible, c'est la pire situation. Finalement, c'est la 2e cible que l'on tente toujours d'approcher. Le choix du meilleur modèle pour un algorithme donné est primordial.

I.6 La manière de combiner les classifieurs

Disposant d'un ensemble de L classifieurs, comment les agencer pour fiabiliser une prise de décision?

I.6.1 Solution

Dépend de la façon dont on veut faire interagir les classifieurs

- ✓ Indépendamment les uns des autres (vote)
- ✓ Elimination d'hypothèses (décisions dépendantes)
- ✓ Coopération de classifieurs (chacun résout un problème…)

I.7 Optimisation d'ensembles de classifieurs

La création d'ensemble de classifieurs nécessite la génération des classifieurs individuels différents les uns des autres. Ces classifieurs ne doivent pas nécessairement être le plus performant possible mais doivent, lorsque utilisés en comité, offrir de meilleures performances que le plus performant des classifieurs de l'ensemble. Souvent, la génération d'un grand nombre de classifieurs suffit à atteindre cet objectif. Ajouter des classifieurs à un ensemble jusqu'à l'obtention d'un certain seuil de performance est donc une façon de créer d'ensemble de classifieurs. Par contre, cette méthode a le désavantage d'être sous-optimale. En effet, rien ne garantit l'obtention d'un seuil donné. De plus, certains classifieurs n'aident pas à améliorer la performance de l'ensemble. Au mieux ils ajoutent à la complexité du système. Au pire, et c'est souvent le cas, ils en diminuent la performance. Aucun classifieur individuel n'est directement responsable de cette situation et c'est l'agencement de ces derniers qui en est la cause [27]. C'est là tout le problème de la création d'ensemble de classifieurs car on ne peut prédire complètement la performance d'un ensemble à partir de la seule étude des individus le composant.

I.8 Méthodes ensemblistes

Lorsqu'il faut prendre une décision importante, il vaut souvent mieux recueillir plusieurs avis que de se fier à un seul. Utiliser un modèle de machine learning pour prédire un comportement c'est un premier pas. Mais agréger des milliers de modèles ayant des avis divergents mais pouvant être chacun spécialisés sur des parties de la base de données le plus souvent de meilleurs résultats. Nous parlons alors de méthodes ensemblistes. [29]

Face au très grand nombre de méthodes d'apprentissage statistique présentées dans la littérature, a émergé l'idée de les agréger pour tirer le meilleur parti de leurs avantages respectifs. [29]

Les méthodes d'ensembles sont des méthodes permettent de construire une collection de prédicteurs et agréger l'ensemble de leurs prédictions (voir Figure1.4). Cette

idée suit le comportement de la nature humaine qui tend à rechercher plusieurs avis avant de prendre une décision importante [30]

L'utilisation d'ensembles de classifieurs permet souvent d'augmenter la performance d'un système de reconnaissance de formes par rapport à un système mono-classifieur. En revanche, les ensembles de classifieurs forment des systèmes plus complexes qui demandent plus de ressources à mettre en oeuvre. Bien que le principal objectif de tout système basé sur un ensemble de classifieurs soit d'en maximiser la performance, en limiter la complexité fait souvent partie des contraintes importantes.

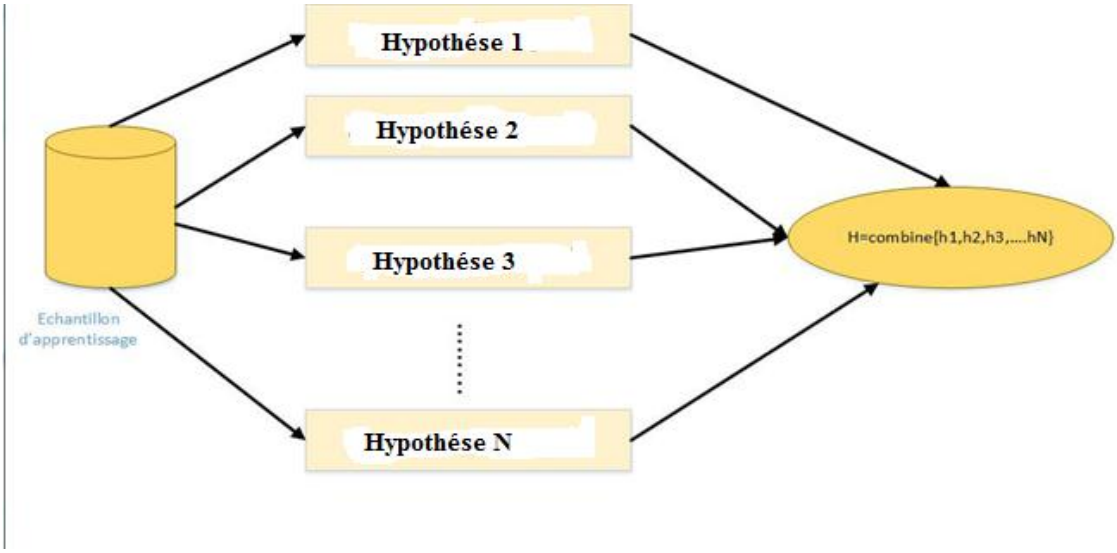


FIGURE I.5 – Principe général des méthodes d'ensemble parallèle

I.8.1 Les types

Les méthodes d'ensemble sont divisées selon leur principe de fonctionnement en deux types : les méthodes homogènes et les méthodes hétérogènes.

I.8.1.1 ENSEMBLE DE CLASSIFIEURS HOMOGÈNES

Les méthodes homogènes son principe de fonctionnement est de traiter plusieurs ensembles d'apprentissage en parallèle utilisant un seul modèle de classification.
Cet axe de recherche a suscité beaucoup d'intérêt dans la littérature :

A. Bagging (Bootstrap aggregating) est une approche de construction d'ensemble qui utilise différents sous-ensembles de données d'apprentissage avec une méthode de classification unique [17]. Étant donné un ensemble d'apprentissage de taille **t**, Bagging attire des instances aléatoires **t** de l'ensemble de données avec remplacement

(à l'aide d'une distribution uniforme). Ces **t** instances sont apprises, et ce processus est répété plusieurs fois. Etant donné que le tirage au sort est effectué avec remplacement, les instances tirées contiendront des doublons et des omissions par rapport à l'ensemble d'apprentissage initial. Chaque cycle à travers le processus aboutit à un classifieur. Après la construction de plusieurs classifieurs, les sorties de chaque classifieur sont combinés pour produire la prédiction finale [17].

B. Une autre approche appelée « **Boosting** » utilise également une méthode d'apprentissage unique avec différents sous-ensembles de données d'apprentissage [reff]. Sa structure globale est similaire à celle de la méthode Bagging, à la différence qu'elle conserve la trace de la performance de l'algorithme d'apprentissage et se concentre sur les cas qui ne sont pas correctement appris. Au lieu de choisir les instances **t** d'apprentissage à l'aide d'une distribution uniforme de manière aléatoire, les exemples d'apprentissage sont sélectionnés en favorisant les instances qui ne sont pas bien classées. Après plusieurs cycles, la prédiction est réalisée selon un vote pondéré des prédictions de chaque classifieur. Ainsi, les poids sont proportionnels à la précision de chaque classifieur sur son ensemble d'apprentissage. L'algorithme le plus connu de l'approche Boosting, appelée « AdaBoost », a été proposée par frieman et al. [27].

C. Les **forêts aléatoires** (plus connus sous **Random Forest**) sont une combinaison d'arbres de décision, où chaque arbre dépend des valeurs d'un vecteur aléatoire indépendamment échantillonné et avec la même distribution pour tous les arbres de la forêt. L'erreur de généralisation d'une forêt d'arbres dépend de la force des arbres individuels dans la forêt et de la corrélation entre eux. L'utilisation d'une sélection aléatoire de caractéristiques pour diviser chaque nœud donne des taux d'erreur qui se comparent favorablement à Adaboost.

Ce schéma résume les types :

-Différent ensembles d'apprentissage
-Même algorithmes ou modèle de classification

-Même ensemble d'apprentissage

-Différent algorithmes ou modèle de classification

FIGURE I.6 Les type de méthodes ensemblistes -1-

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Selon la façon de générer les composants des classificateurs, des algorithmes d'apprentissage ensemble actuel se divisent en deux catégories, à savoir, les algorithmes qui génèrent les composants des classificateurs en parallèle et des algorithmes qui génèrent les composants des classificateurs en séquence.

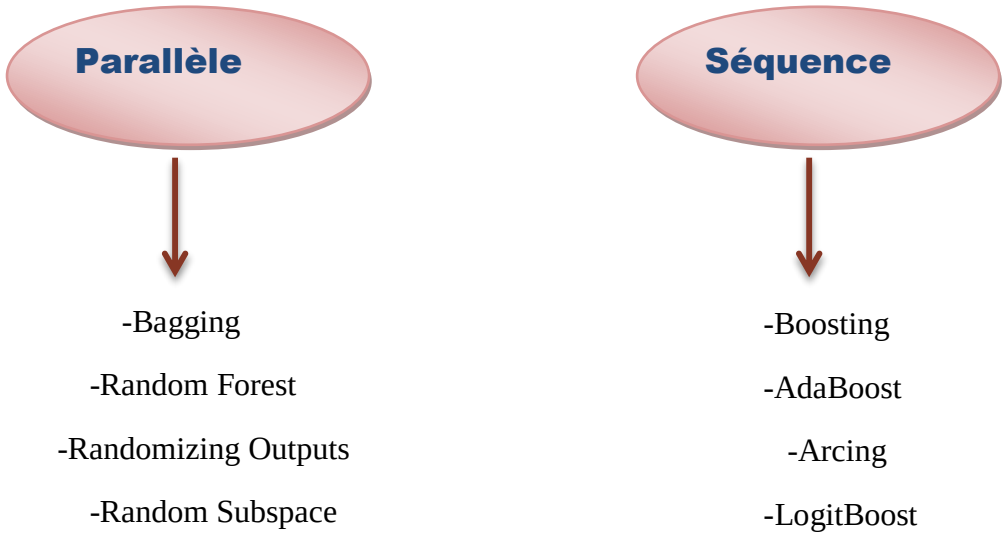


FIGURE I.7 Les type de méthodes ensemblistes -2-

- **Bagging ou Random Forests** : utilise le hasard pour améliorer les performances d'algorithmes et créer la diversité.
- **Randomizing Outputs**: construit des échantillons indépendants dans lesquels les sorties sont randomisées
- **Random Subspace:** considère que l'ensemble des variables, applique la règle de base en ne tenant compte que des variables sélectionnées
- **Boosting** : utilise une stratégie adaptative pour booster des performances, applicable là aussi à tout type d'algorithme (Réseau de neurones, Cart, etc.)

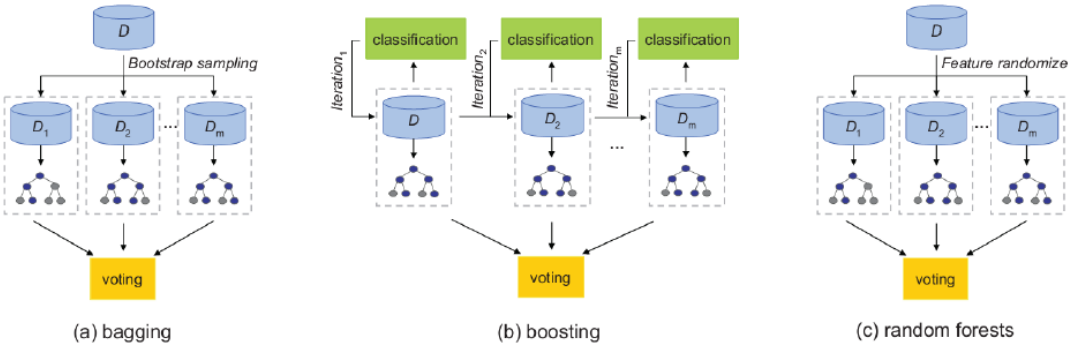


FIGURE I.8 Illustration schématique des trois méthodes d'ensemble homogènes populaires

I.8.1.2 ENSEMBLE DE CLASSIFIEURS HETEROGÈNES

Ce type d'ensemble permet la combinaison d'hypothèses produites à partir de méthodes d'apprentissage différentes (des hypothèses de nature hétérogène), appliquées à un même échantillon d'apprentissage. Les travaux de la littérature dans ce domaine portent soit sur la manière de combiner les classifieurs efficacement, soit sur la sélection des classifieurs. [31]

Le vote est la méthode la plus simple pour combiner les prédictions à partir de classifieurs multiples et en va citer les types de vote établie par Bahler et Navarro (2000) lesquelles :

- ✓ **Le vote à majorité :** C'est une technique simple et intuitive, qui consiste à classer la nouvelle instance selon la prédiction majoritaire des classificateurs de base. L'inconvénient de cette méthode est dans le cas ou plus de la moitié des classificateurs de base obtiennent de faux résultats.

- ✓ **Le vote à majorité pondéré** : C'est un vote basé sur des poids associés aux classificateurs de base. Ces poids peuvent être diminués ou augmentés au fur et à mesure que les classificateurs s'entraînent, suivant qu'ils produisent respectivement une bonne ou une mauvaise prédiction.

- ✓ **Le vote avec seuil :** pour lequel l'´etiquette predite est celle recevant le plus de votes, avec un écart significatif par rapport aux suffrages recueillis par les autres étiquettes. Ce principe peut être également utilise dans un contexte pondéré

- ✓ **L'unanimité** : pour laquelle une étiquette sera associée a un nouvel individu si toutes les hypothèses de l'ensemble concordent sur celle-ci. Il est donc possible qu'un tel procède aboutisse a une situation d'indeterminisation. (on parle alors d'abstention). La encore, il est possible d'adapter ce principe dans un contexte pondéré. Notons ici que ces combinaisons par vote s'utilisent également pour

combiner des hypothèses construites par une seule méthode d'apprentissage à partir d'´echantillons différents.

- ✓ **Sélection dynamique d'hypothèses :** La sélection dynamique d'hypothèses est comparable au principe de sélection adaptative de la meilleure hypothèse au sein d'une famille. Mais la ou certaines méthodes permettent simplement de choisir l'hypothèse jugée la plus pertinente sur l'´echantillon d'apprentissage, les méthodes de sélection dynamique permettent de choisir, pour chaque nouvel individu, l'hypothèse jugée la plus apte à prédire son étiquette. Dans ce contexte, Merz (1996) propose par exemple un mécanisme d'´evaluation, pour différentes zones géométriques de l'espace de représentation, de la performance de chaque hypothèse de base. La classification d'un nouvel individu est alors effectuée par l'hypothèse jugée la plus pertinente dans la zone ou cet

  individu est situe. Giacinto et Roli (1997) [32] proposent une méthode comparable, o`u la sélection est basée sur les performances de chaque classifieur sur les N plus proches voisins de l'exemple considéré dans l'´echantillon d'apprentissage.

- ✓ **Stacking** : C'est une méthode qui permet de combiner plusieurs classificateurs de base. La première phase consiste à induire N classificateurs Cli , à partir de N ensembles de données {S1, S2, ..., SN}. Le test est ensuite fait sur un ensemble d'évaluation T= {t1, t2, ..., tL}, indépendant des ensembles d'apprentissage Si. Dans la deuxième phase, un nouvel ensemble de données M est formé par les valeurs calculées Cli(tj) et la vraie classe de l'instance tj, classe(tj). Chaque instance de M sera de la forme . Dans la dernière étape, un classificateur global est construit à partir de M. Les classificateurs de base peuvent être construits avec des algorithmes différents (arbres de décision, réseaux de neurone..) selon les contraintes du problème [33]

I.9 Conclusion

Dans ce chapitre, nous avons établi un état de l'art non exhaustif des méthodes ensembliste qui abordent le problème de l'Apprentissage Automatique de nombreux travaux montrent que la combinaison de classifieurs (séquentielle, parallèle ou hybride) améliore nettement les performances du système de reconnaissance par rapport a chacun des classifieurs et on a exposé une étude sur ces méthodes puis on a présenté les trois méthodes que l'on va travailler avec et qui seront étudier profondément dans le chapitre suivant.

Chapitre II

Principe des méthodes utilisées

II.1. Introduction

Nous nous sommes intéressés dans ce travail à étudier la performance des modèles ensemblistes sur une tâche de classification reliée au domaine médical.

Les méthodes d'ensembles constituent une famille ou un ensemble d'algorithmes qui génèrent une collection de classifieurs et agrègent leurs prédictions. Dans la littérature, diverses approches sont proposées pour arriver à une classification plus performante parmi elles les méthodes d'ensemble.

Dans ce chapitre, nous présentons toutes les informations nécessaires à l'apprendre pour facilité la compréhension des méthodes utilisées.

II.2 Approche ensembliste

L'objectif visé est que le prédicteur final soit meilleur que chacun des prédicteurs individuels.

Au lieu d'essayer d'optimiser une méthode, les méthodes ensemblistes génèrent plusieurs règles de prédiction et mettent ensuite en commun leurs différentes réponses. L'heuristique de ces méthodes est qu'en générant beaucoup de prédicteurs, on explore grandement l'espace des solutions, et qu'en agrégeant toutes les prédictions, on récupère un prédicteur qui rend compte de toute cette exploration. [34]

Pour qu'une méthode d'ensemble soit performante, elle doit réussir à construire une collection de prédicteur qui vérifie ces deux points :

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

II.3 Ensemble de classifieurs homogènes

Les mécanismes qui sont utilisés pour construire un ensemble de classifieurs homogènes se basent soit sur l'emploie de différents sous-ensembles de données d'apprentissage en utilisant une méthode de classification unique, ou bien sur l'utilisation de différents paramètres d'apprentissage avec une méthode de classification unique [35].

Probablement, bagging et boosting sont les méthodes de construction d'ensembles les plus utilisées dans la littérature. Initialement, ces deux techniques étaient conçues pour améliorer la performance des arbres de décision qui sont des classifieurs faibles (dont les performances ne sont pas stables).

Cependant, il est constaté qu'elles sont aussi efficaces pour d'autres types de classifieurs tels que les classifieurs linéaires [36], les réseaux de neurones [37] et les k-plus proches voisins [38]. Pour les classifieurs linéaires, la performance de bagging et boosting est affectée par la taille de la base d'apprentissage, le choix du classifieur ainsi que la règle de combinaison [36].

II.3.1 Bagging

Bagging est une méthode d'ensemble introduite par Breiman en1996 est basée sur les concepts de Bootstrapping et d'aggregating. Le Bootstrap [39] est un principe de ré-échantillonnage statistique [39] traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. Le bootstrapping est conçu pour générer au hasard et avec remise L copies indépendantes de S objets appelées bootstrap à partir de l'ensemble initial des échantillons d'apprentissage de taille S. Un objet de la base initiale peut être sélectionné plusieurs fois comme il peut être absent dans les copies générées. Le même classifieur est appris sur chacune des copies. On obtient par la suite L classifieurs avec des performances différentes. L'aggrégation consiste à combiner ces classifieurs en utilisant le vote à la majorité comme règle de combinaison. L'algorithme suivant illustre le fonctionnement de bagging.

Algorithme 1 la méthode bagging

Entrée :

B^a : *base d'apprentissage*

S : taille de la base d'apprentissage

X forme à reconnaitre

L classifieur

Début

Pour *j de 1 à L* ***faire***

Début

Générer la sous-base b_j à partir de B^a

Construire le classifieur e_j (x) en utilisant la base b_j

Fin pour

Combiner les L classifieurs construits par le vote à la majorité afin d'obtenir la décision finale de x.

Fin

L'étude simulée de Skurichina [40] a montrée que, généralement, bagging est une technique qui permet d'améliorer la performance des classifieurs linéaires instables. Cette conclusion à était déjà montrée dans [41] La réduction de l'erreur de classification ne peut donc être obtenue que dans le cas d'une combinaison de classifieurs instables. Pour des classifieurs stables, elle détériore la performance de la combinaison.

L'utilisation du bagging est adaptée aux algorithmes à fortes variance qui sont ainsi stabilisés (réseaux neuronaux, arbres de décision pour la classification ou la régression…), mais il peut également dégrader les qualités pour des algorithmes plus stables (k plus proches voisins avec k grand, régression linéaire). [42]

Son principe est donc de tirer uniformément et avec remise les observations pour créer de nouveaux échantillons, sur chacun desquels est construit un classifieur. La prédiction finale est obtenue par vote majoritaire parmi les classifieurs intermédiaires (voir Figure 2.2).

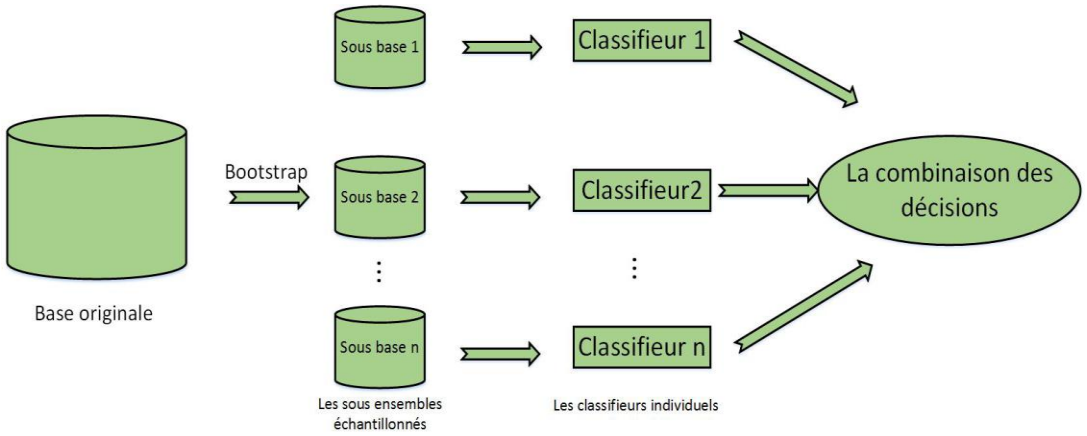


FIGURE II.1 Principe de Bagging

Le terme de ré échantillonnage, ou, en anglais, "bootstrap" désigne un ensemble de méthodes qui consistent à faire de l'inférence statistique sur de nouveaux échantillons tirés à partir d'un échantillon initial (voir Figure 2.3).

L'idée bootstrap est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble des données observées et à l'aide d'un tirage aléatoire avec remis [27]

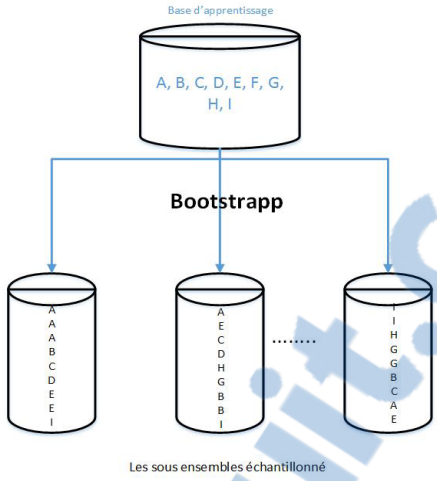


FIGURE II.2 Principe de Bootstrapping

Le Bootstrap est un principe de ré échantillonnage statistique traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. Ainsi chaque classifieur élémentaire de l'ensemble sera entrainé sur un des échantillons bootstrap de sorte qu'ils soient tous entrainés sur un ensemble d'apprentissage différent.L'agrégation de ces classifieurs permet d'obtenir un prédicteur plus performant. [34]

II.3.2 Boosting

Le boosting est une méthode d'ensemble introduite par Schapire, Leur objectif c'est pour améliorer les résultats et obtenir un meilleur classifieur à partir d'un classifieur peu performat . Le principe est issu de la combinaison de classifieurs (appelés également hypothèses). Par itérations successives, la connaissance d'un classifieur faible est ajoutée au classifieur final.

Le mot boosting s'applique à des méthodes générales capables de produire des décisions très précises (au sens d'une fonction de perte) à partir d'un ensemble de règles de décision « faibles », c'est-à-dire dont la seule garantie est qu'elles soient un peu meilleures que le hasard. Ces méthodes s'appliquent aussi bien à l'estimation de densité qu'à la régression ou à la classification. Pour simplifier, nous nous concentrons ici sur la tâche de classification binaire. Dans sa version « par sous-ensembles », cette technique fait produire à l'algorithme trois résultats selon la partie de l'ensemble d'apprentissage sur laquelle il apprend, puis combine les trois apprentissages réalisés pour fournir une règle de classification plus efficace. Examinons d'abord cette technique avant de voir comment la généraliser à l'aide de distributions de probabilités sur les exemples.

Rapport-Gratuit.com

Le boosting est un domaine de l'apprentissage automatique (branche de l'intelligence artificielle). C'est un principe qui regroupe de nombreux algorithmes qui s'appuient sur des ensembles de classifieurs binaires : le boosting optimise leurs performances.

Contrairement au bagging où les ensembles d'apprentissage de bootstrap et les classifieurs sont construits de manière indépendante, dans boosting les échantillons d'apprentissage sont construits incrémentalement par le même classifieur et de manière séquentielle. Initialement, tous les échantillons d'apprentissage ont des poids égaux et le classifieur est construit sur cette base. Ensuite pour chaque étape, les échantillons sont pondères de façon à ce que les objets mal classées (qui se trouvent généralement aux frontières des classes) aient des poids élevées et le classifieur est lancé sur le nouvel ensemble d'apprentissage ainsi pondère. De cette manière, on obtient finalement un ensemble de classifieurs qui sont combines par un vote pondéré pour avoir la décision finale.

Algorithme 2 : la méthode boosting **[27]**

Entrée:

B^a: base d'apprentissage
S: taille de la base d'apprentissage
x: forme à reconnaitre

Début

Initialiser tous les poids w_s^1 (s=1,...,S) des éléments de la base d'apprentissage B^a à $\frac{1}{S}$.
Pour j de 1 à L **faire**
Début

Générer la sous-base b_j à partir de B^a.
Construire le classifieur e_j en utilisant la base b_j
Calculer l'erreur de l'ensemble

$$\epsilon_j = \sum_{s=1}^{S} w_s^j (1 - e_{s,j})$$

$e_{s,j}$ =1 si le classifieur e_j reconnait correctement le s^{ieme} élément et $e_{s,j} = 0$ sinon. Si $\epsilon_j = 0$ ou $\epsilon_j \geq 0.5$ alors réinitialiser w_s^j à $\frac{1}{S}$

Calculer le coefficient de pondération à utiliser dans la règle de combinaison

$$\beta_j = \frac{\epsilon_j}{1 - \epsilon_j}$$

Calculer le coefficient de pondération à utiliser dans la règle de combinaison

$$\beta_j = \frac{\epsilon_j}{1 - \epsilon_j}$$

Calculer le poids de chaque élément de la base suivante

$$w_s^{j+1} = \frac{w_s^j \beta_j}{\sum_{l=1}^{S} w_l^j \beta_j}$$

Fin faire
Combiner les L classifieurs construits par le vote pondéré afin d'obtenir la décision finale de x. Le vote de chaque classifieur e_j est pondéré par un poids de $log(\beta_j)$.

Fin

Le Boosting s'appuie sur le même principe que le Bagging (voir Figure II.3) : il construit un ensemble de classifieurs qui sont ensuite agrégés par une moyenne pondérée des résultats. Cependant, dans le cas du Boosting, cet ensemble de classifieurs est construit d'une façon récurrente et itérative. plus précisément, chaque classifieur est une version adaptative du précédent en donnant plus de poids aux observations mal prédites [43].

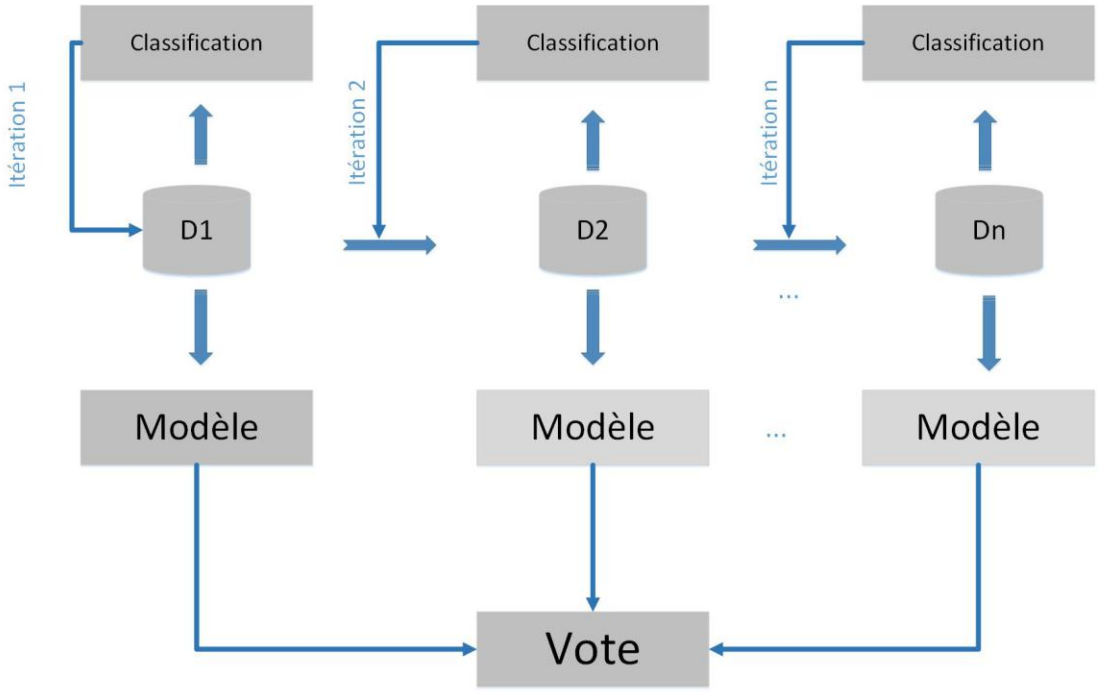


FIGURE II.3 Principe de Boosting

Adaboost repose sur la sélection itérative de classifieur faible en fonction d'une distribution des exemples d'apprentissage.Chaque exemple est pondéré en fonction de sa difficulté avec le classifieur courant. dû à sa simplicité et à ses performances et dont le pseudo-code est détaillé par l'algorithme ci-dessous. L'expression d'Adaboost est une concaténation de "Adaptative Boosting".Il construit un vote de majorité itérativement de manière adaptative.

À la fin de l'algorithme, chaque classifieur ht est pondéré par une valeur *(at)* calculée dans la boucle de l'algorithme. La classification d'un nouvel exemple se fait en utilisant un vote de majorité pondéré [44].

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Algorithme 3 approche d'Adaboost

Algorithme : Adaboost

Entrées : Une base d'exemple $(x_1,y_1)\ldots (x_m , y_m)$, avec des labels $y_i \epsilon\{1,1\}$

Pour *t=1,…..T faire*

-On trouve *le classifieur faible minimisant un critére d'erreur*
-On calcule son poids
-On calcule une distribution D(i) des exemples

FinPour

Sorties : On construit le classifieur fort en effectuant une somme pondérée des classifieurs séléctionnés

II.3.3 principe des forêts aléatoires

Les forêts d'arbres décisionnels (ou forêts aléatoires de l'anglais random forest classifier) ont été formellement proposées en 2001 par Leo Breiman et Adèle Cutler. Elles font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de bagging. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

II.3.3.1 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.

L'apprentissage par arbre de décision est une méthode classique en apprentissage automatique. Son but est de créer un modèle qui prédit la valeur d'une variable-cible depuis la valeur de plusieurs variables d'entrée. Une des variables d'entrée est sélectionnée à chaque nœud intérieur (ou interne, nœud qui n'est pas terminal) de l'arbre selon une méthode qui dépend de l'algorithme et qui sera discutée plus loin. Chaque arête vers un nœud-fils correspond à un ensemble de valeurs d'une variable d'entrée, de manière à ce que l'ensemble des arêtes vers les nœuds-fils couvrent toutes les valeurs possibles de la variable d'entrée. Chaque feuille (ou nœud terminal de l'arbre) représente soit une valeur de la variable-cible, soit une

distribution de probabilité des diverses valeurs possibles de la variable-cible. La combinaison des valeurs des variables d'entrée est représentée par le chemin de la racine jusqu'à la feuille.

L'arbre est en général construit en séparant l'ensemble des données en sous-ensembles en fonction de la valeur d'une caractéristique d'entrée. Ce processus est répété sur chaque sous-ensemble obtenu de manière récursive, il s'agit donc d'un partitionnement récursif. La récursion est achevée à un nœud soit lorsque tous les sous-ensembles ont la même valeur de la caractéristique-cible, ou lorsque la séparation n'améliore plus la prédiction. Ce processus est appelé induction descendante d'arbres de décision.

II.3.3.2 Les forêts aléatoires

Les forêts aléatoires sont composées (comme le terme "forêt" l'indique) d'un ensemble d'arbres décisionnels. Ces arbres se distinguent les uns des autres par le sous-échantillon de données sur lequel ils sont entraînés. Ces sous-échantillons sont tirés au hasard (d'où le terme "aléatoire") dans le jeu de données médical.

Les forêts aléatoires sont des méthodes d'ensembles qui se basent sur l'injection d'aléatoire. Le principe général des méthodes d'ensemble est de construire une collection de prédicteurs, pour ensuite 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 prédicteurs.

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 [5]. Breiman a proposé le formalisme de cet algorithme et le package est codé avec Fortran 77 [45].

Après ces définitions et les principes de chaque méthode en va passer sur l'état de l'art de ces derniers

Le tableau suivant représente la modification au niveau algorithme ont proposé dans la littérature : [34]

Auteurs	Titres	Méthodes et expériences	Résultats
Hoens & Chawla 2010 [54]	Generating Diverse Ensembles to Counter The Problem of Class Imbalence	Dans le but d'améliorer la classification sur ce genre de données les auteurs ont travaillés sur plusieurs base de données et plusieurs méthodes individuellement comme : l'AdaboostM1(BT), Bagging(BG), Random Forest(RF), et Random subspace method(RSM), ainsi que deux autres méthodes en combinaison RSM+SMOTE, RSM+Undersumpling. avec un seul classifieur qui est l'arbre de décision C4.5.	Les expérimentations montrent que le meilleur classifieur est RSM+SMOTE, dans 12 des 21 bases de données.

TABLE II.1 Représentation d'état de l'art des méthodes ensembliste-1-

Robin Genuer1 et Jean-Michel Poggi2 p17, 34 [16]	Arbres CART et Forêts aléatoires Importance et sélection de variables	Un jeu de données fil rouge, l'application du bagging et rf sur les très classiques données spam l'erreur test atteinte par le Bagging (6%) randomForest atteint un taux d'erreur test de 5, 2%

Philippe Foucher [15]	Classification automatique de défauts sur des images de tunnels par forêts d'arbres aléatoires	L'algorithme Random Forest permet ensuite de classer chaque échantillon dans la catégorie « avec défaut » ou « sans défaut ». les résultats de deux expérimentations. Pour chaque expérience, nous ont sélectionné une partie des images pour effectuer l'apprentissage et l'évaluation. Environ 500 échantillons par classe

TABLE II.2 Représentation d'état de l'art des méthodes ensembliste-2-

II.4 La différence entre les trois classifieurs

La différence entre le bagging et le boosting	La différence entre le bagging et le random forest
Bagging • Génération de k échantillons « indépendants » par tirage avec remise • Pour chaque échantillon, apprentissage d'un classifieur en utilisant le même algorithme d'apprentissage • La prédiction finale pour un nouvelle exemple est obtenue par : – Vote majoritaire (classification) – Moyenne des predictions (regression) ***Intérêt*** • Le bagging réduit la variance quand les prédicteurs sont instables [Breiman 96] • Estimation de l'erreur de prédiction par Bootstrap out of bag : prévient le sur-apprentissage **Boosting :** La prédiction finale est issue d'une combinaison (vote pondéré) de plusieurs prédictions ! Méthode : – Itérative – Chaque classifieur dépend des précédents (les classifieurs ne sont donc pas indépendants comme dans d'autres méthodes de vote) – Les exemples sont pondérés différemment	**Bagging et arbre de décision :** • Pour des arbres de décision, plusieurs solutions : – Construire des arbres complets sans élagage – Construire des arbres d'au plus q feuilles – Construire des arbres complets et élaguer par validation croisée Quelle est la meilleure stratégie ? Bagging et arbres de décision • Pour des arbres de décision, plusieurs solutions : – Construire des arbres complets sans élagage – Construire des arbres d'au plus q feuilles – Construire des arbres complets et élaguer par validation croisée • Quelle est la meilleure stratégie ? – En pratique, on garde souvent la première stratégie, chacun des arbres a un faible biais mais une grande variance – La moyenne réduit la variance **Random Forests :** • Problème lié au vote des classifieurs : sont-ils réellement indépendants entre eux ? – Est-ce bien d'avoir des classifieurs très corrélés entre eux? • Amélioration du Bagging dans le cas spécifique des arbres de décision • L'objectif est de rendre les modèles (arbres) construits plus indépendants entre eux • Cette indépendance va permettre de rendre le vote des experts plus efficace Principe des random forests • Jeux de données différents – C'est du bagging) • Apprentissage des arbres par tirage aléatoire des attributs – Pour la construction de chaque noeud de chaque arbre, on tire uniformément q attribut parmi p pour former la décision associée au nœud – On se limite généralement à des arbres de faible profondeur (pas le problème du bagging car les arbres sont très peu corrélés) ➢ **donc Différence entre random forest and bagging :** La différence fondamentale est que, dans les forêts aléatoires, seul un sous-ensemble de caractéristiques est sélectionné au hasard hors du total et la meilleure caractéristique de partage du sous-ensemble est utilisée pour diviser chaque nœud dans un arbre, contrairement à l'ensachage où toutes les fonctionnalités sont considérées pour le fractionnement Un nœud.

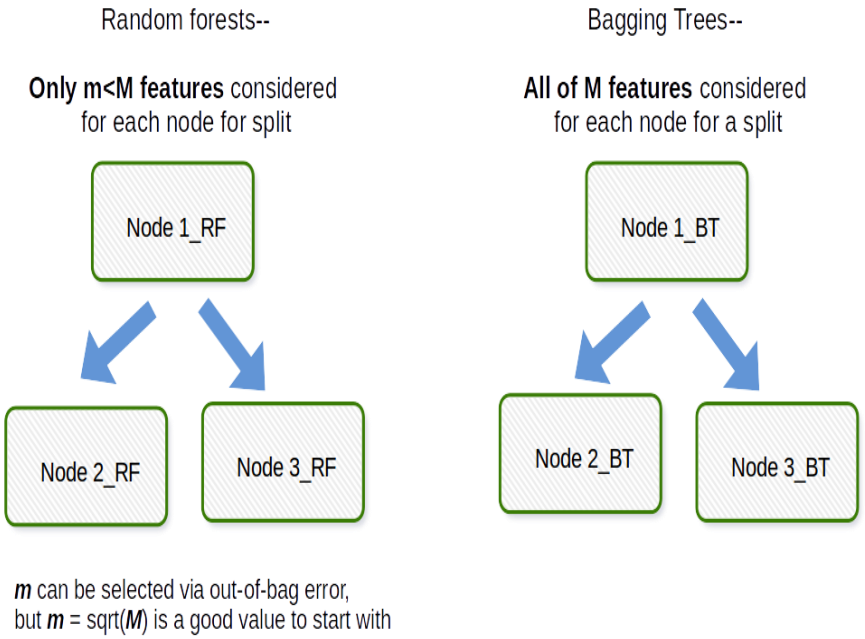
– Le poids des exemples reflète la difficulté des classifieurs précédents à les apprendre	

FIGURE II.4 Random Forest vs Bagging

Les forêts aléatoires améliorent la variance en réduisant la corrélation entre les arbres, ce qui se fait par sélection aléatoire du sous-ensemble de fonctionnalités pour la division à chaque nœud. mtry : le nombre de variables tirées aléatoirement `a chaque nœud

Alors que Bagging améliore la variance en faisant la moyenne / sélection de la majorité des résultats à partir de multiples arbres pleinement développés sur les variantes de l'ensemble de formation. Il utilise Bootstrap avec remplacement pour générer plusieurs ensembles de formation.
La méthode Bagging exploite le principe de la diversité en utilisant les différents ensembles de données perturbées dans la base d'apprentissage des classifieurs. Autrement dit, chaque classifieur de base est entrainé sur un sous-ensemble d'échantillons pour obtenir une hypothèse de classification légèrement différente, puis il est combiné avec les autres classifieurs pour former l'ensemble. D'un autre coté, la méthode Random Forest utilise le même principe mais avec différents ensembles d'attributs. En ce qui concerne la méthode boosting, la diversité est obtenue en augmentant les poids des échantillons mal classés de manière itérative. Généralement, ces trois méthodes utilisent des arbres de décision pour 74 leur sensibilité aux petits changements sur l'ensemble d'apprentissage, et sont donc adaptées à la procédure de perturbation appliquée aux données d'apprentissage [17]

TABLE II.3 comparaison entre les trois classifieurs

II.4.1 Étude comparative entre Bagging et Boosting

L'étude comparative de Dietterich [46], montre que dans le cas des données déséquilibré, Adaboost est plus performant que le bagging et qu'il est moins risqué au cas du sur-apprentissage, car Adaboost essaye directement d'optimiser les votes pondérés. Cette théorie est prouvée par :

1. La diminution exponentielle de l'erreur empirique d'adaboost sur l'échantillon d'apprentissage avec le nombre d'itération.
2. la diminution de l'erreur en généralisation qui continue à baisser même lorsque l'erreur empirique à atteint son minimum.

Contrairement au Bagging qui demande un grand nombre d'itération pour que l'erreur en généralisation se stabilise, l'erreur de généralisation du Boosting ne diminue pas lorsque l'erreur en apprentissage a atteint son minimum. C'est ainsi que le Bagging est plus gourmand que le Boosting du point de vue d'espace mémoire puisque chaque classifieurs doit être stocké pour la prédiction d'un nouvel exemple [47].

Bagging	**Boosting**
Aléatoire	Adaptatif et généralement déterministe
Utilise des échantillons Bootstrap	Utilise échantillon initial au complet
Les modèles ont le même poids	Les modèles pondérés selon leur qualité d'ajustement

TABLE II.4 bagging vs boosting

II.4.2 Avantages et inconvénients

Classifieur	Avantages	Inconvénients
Adaboost	• Rapide • Simple et facile à implémenter • Applicable à de nombreux domaines par un bon choix de classifieur faible • Pas de sur-apprentissage par la maximisation de la marge	• Boosting inadapté quand : – Pas assez de données – Comité de classifieurs faibles trop restreints – Apprenants faibles trop stables – Apprenants faibles trop forts • Risque de sur-apprentissage par eux même – Données bruitées • Mais permet de trouver les exemples aberrants (outliers)
Bagging	• Simple à mettre en place. S'adapte facilement à n'importe quelles	• Calcul important pour évaluer un nombre important de classifieur.

	méthodes d'apprentissage. • Permet de réduire l'impacte du choix de l'ensemble d'apprentissage sur les résultats de classification.	• Nécessite de stocker tous les modèles de la combinaison pour évaluer un nouvel exemple. • Le modèle final n'est pas facilement interprétable. Approche de type boite noire.
Random forest	– Bien adaptées aux très grandes dimensions – Très simples à mettre en œuvre – Nombreux succès applicatifs ces dernières années Elles sont en général plus efficaces que les simples arbres de décision.	- possède l'inconvénient d'être plus difficilement interprétables - Apprentissage souvent long Valeurs extrêmes souvent mal estimées dans cas de régression

TABLE II.5 étude comparative entre les avantages et les inconvénients des méthodes ensemblistes

II.5 Ensemble de classifieurs hétérogènes

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

II.5.1 Vote majoritaire

Dans les méthodes ensemblistes, 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 [31] 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".

Egalement appelé vote simple qui consiste à affecter un poids équivalent à chaque classifieur et revient donc à construire un méta-classifieur correspondant à prédire la classe la plus fréquemment votée par les T modèles.

II. 5.2 Vote pondéré

Chaque classifieurs reçoit un poids *(a t)*proportionnel à sa performance estimée sur un échantillon test. Ce système de vote donne une meilleure performance par rapport au vote simple. [17]

L'objectif principal des systèmes pondérés est d'améliorer la sérénité des votes. Le fait de s'exprimer sur chaque choix, ou candidat, est la seule possibilité qui permet cela. De plus, le fait de s'intéresser à tous les choix permet d'avoir une plus grande maturité puisqu'il faut connaître plus ou moins tous les candidats.

II. 5.3 Majorité pondérée

L'algorithme de Majorité pondérée est une généralisation du HALVING .il est semblable au vote pondéré, à la différence que les poids *(a t)* attribués aux différents classifieurs formant le comité sont appris eux -même par un algorithme d'apprentissage.

-Parmis les méthodes qui permettent la combinaison des classifieurs hétérogènes, nous trouvons la méthode « ***Stacking*** ». Cette méthode vise à améliorer l'efficacité en exécutant un certain nombre de processus d'apprentissage et en combinant les résultats collectifs. La principale différence entre la méthode du vote et la méthode Stacking est que ce dernier combine les classifieurs de base d'une manière non linéaire. La tâche de combinaison intègre les classifieurs de base calculés indépendamment dans un classifieur de niveau supérieur, un méta-classifieur, en réapprenant l'ensemble d'apprentissage au niveau superieur (méta-niveau). Cet ensemble est créé en utilisant les prédictions des classifieurs de base sur l'ensemble de validation comme valeurs d'attribut et la classe désirée comme cible. Les auteurs ont montré que la généralisation réussie nécessite l'utilisation de distributions de classes de sortie plutôt que de prédictions de classe. Dans leurs expérimentations, seul l'algorithme MLR (un discriminant linéaire) était approprié pour être utilisé comme un classifieur de niveau 1[31].

6 Conclusion

En résumé, il existe plusieurs façons de construire des classifieurs dont les sorties sont différentes. On peut obtenir ces classifieurs en introduisant des modifications sur les bases d'apprentissage, l'espace de caractéristiques, les structures des classifieurs, etc.

Dans ce chapitre nous avons présenté des méthodes ensemblistes pour trouver un ensemble d'hypothèses qui sont différents dans leurs prises de décision à fin qu'ils puissent se compléter mutuellement. Dans le chapitre suivant nous allons montrer les résultats obtenus pour chaque méthode.

Chapitre III

Expérimentations et Résultats

III.1 Introduction

Nous présentons dans ce chapitre, notre étude expérimentale répartie en 3 parties appliquant l'adaboostM1, le bagging, et le random forest sur les bases de données 3 bases sont binaires (colon, breast cancer et hepatite) et 2 multiclasses (heart et pendigits) puis on surmonte le problème de déséquilibre de classes

a. Expérimentation : études comparative entre les trois algorithmes appliquées sur cinq bases de données

III.2 Bases de données

Pour la réalisation de notre travail, nous appliquons 5 bases de données médicales extraites du dépôt d'UCI [48] : colon, Brest cancer-Wisconsin, pendigits, heart, hepatite.

III.2.1 Colon

Maladie est colon cancer Échantillon est 62, Nombre de caractéristiques est 2000 et Nombre de Classes est 2

III.2.2 breast cancer-wisconsin

Les caractéristiques sont calculées à partir d'une image numérisée d'une aspiration à l'aiguille fine (FNA) d'une masse du sein. Ils décrivent les caractéristiques des noyaux cellulaires présents dans l'image.

- ✓ **Le numéro d'identification**
  - Diagnostic (M=malin, B=bénigne)
  - 10 fonctions à valeurs réelles sont calculées pour chaque noyau cellulaire
  - Rayon (moyenne des distances de centre à des points situés sur le périmètre)
  - Texture (écart-type de valeurs d'échelle de gris)
  - Périmètre
  - La zone
  - Lissé (variation locale dans les longueurs de rayon)
  - Compacité (périmètre∧2/zone-1,0)
  - Concative (gravité des portions concaves du contour)
  - Des points concaves (nombre de portion concaves du contour)
  - Symétrie
  - La dimension fractale (« approximation du littoral »-1

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

III.2.3 Pendigits

- Pour chaque attribut: Tous les attributs d'entrée sont des nombres entiers de 0..100.Le nombre des instances : 7494,numéro de classe : 9,nombre d'attributs : 16 entrées + 1 attribut de classe,le dernier attribut est le code de classe 0..9,valeurs d'attributs manquantes : Non

III.2.4 Heart

La base heart diseases représente les maladies de cœur, 303 instances et 76 attributs Mais la majorité des expériences utilisent la version qui contient 14 attributs. La base est regroupée en 5 classes, la classe 0 représente l'absence de la maladie, et 1, 2, 3,4 la présence d'une maladie. [48]

III.2.5 Hepatite

La base heart diseases représente les maladie de coeur , 155 instances et 19 attribut Mais la majorité des expériences utilisent la version qui contient 14 attributs. La base est regroupée en 5 classes, la classe 0 représente l'absence de la maladie, et 1, 2, 3,4 la présence d'une maladie. [48]

III.3 Matériels et méthodes

Les méthodes ensemblistes ont permis d'améliorer de façon spectaculaire des classifieurs standard en apprentissage supervisé et en particulier le problème de données déséquilibrées, pour cela dans notre étude expérimentale nous avons testé sur des données réelles médicales les différentes méthodes d'ensemble :

b. Adaboost [49] : c'est une méthode d'ensemble séquentiel la plus utilisée, son principe est de lancer l'apprentissage sur toute la base et à chaque itération d'apprentissage il donne l'importance aux instances mal classé.

b. Bagging [50] : c'est une méthode d'ensemble parallèle son principe est de ré-échantillonné la base d'apprentissage en sous ensembles aléatoirement avec remise (Bootstrapping) et pour chaque sous ensemble on affecte un classifieur.

c. Foréts aléatoires : sous le principe : l'union fait la force
• Une « forêt » = un ensemble d'arbres

- apprendre grand nombre T (~ qlq 10aines ou 100aines) d'arbres simples
- utilisation par vote des arbres si classification, ou moyenne des arbres si régression Algorithme proposé en 2001 par Breiman & Cutter

But = obtenir des arbres les plus décarrelés possible

- chaque arbre appris sur un sous-ensemble (~2/3) aléatoire différent des exemples d'apprentissage
- chaque nœud de chaque arbre choisi comme « split » optimal parmi k variables tirées aléatoirement dans les entrées (avec k<< d la dimension des entrés)
- Chaque arbre appris avec algorithme CART sans élagage, On limite fortement la profondeur p des arbres (~ 2 à 5) [51]

Le langage de programmation utilisé c'est le MATLAB et Prtools toolbox

III.4 Mesures de performance

Le critère d'évaluation est un facteur clé à la fois dans l'évaluation de la performance de classification et guidance de la modélisation de classificateur. Pour comparer de façon synthétique la performance des différentes méthodes retenues pour nos expériences nous avons calculé : Sensibilité, Spécificité, [51] :

$$Sen = \frac{\mathrm{VP}}{\mathrm{VP+FN}} \qquad \text{(III.1)}$$

$$Spe = \frac{\mathrm{VN}}{\mathrm{VN+FP}} \qquad \text{(III.2)}$$

$$TC = \frac{\text{nombre des cas bien classée}}{\text{nombre total des cas présents}} \qquad \text{(III.3)}$$

$$TE = \frac{\text{nombre des cas mal classée}}{\text{nombre total des cas présents}} \qquad \text{(III.4)}$$

Vrai positive(VP) : les cas positifs classé positives
Vrai négative(VN) : les cas négatifs classé négatives
Faux positive(FP) : les cas positifs classé négatives
Faux négative(FN) : la négative classée positive
Sensibilité(Sen) : le pourcentage des exemples positive classé correctement
Spécificité(Spe) : le pourcentage des instances négative classé correctement

III.5 Expérimentation 1

Cette partie relate une étude comparative entre les méthodes les plus récentes utilisées pour la classification de données.

- ✓ La question que nous nous posons est ! quelle est la meilleure méthode ?

Les expérimentations ont été réalisées sur 5 bases de données médicales 3 bases de données sont binaires (wisconsin ; colon ;hepatite) ; et deux bases de données sont multi-classe (heart ; pendigits) UCI Machine Learning [48]. Le tableau (Table III.2) résume les caractéristiques de chaque base).

Bases de données	Taille d'apprentissage	Taille de test	Nombre de classes
Colon	2/3	1/3	2
Wisconsin	2/3	1/3	2
Pendigits	2/3	1/3	10
Hepatites	2/3	1/3	2
Heart	2/3	1/3	4

TABLE III.1 concernant la description des bases de données.

Nous avons divisé les bases originales en deux sous bases d'apprentissage et de test. 75% de la base originale pour l'apprentissage et le reste pour le test (25%).

III.5.1. Les résultats de bagging au niveau des bases de données

BDD	Nombre de classifieur	SP	SE	TC	TE
Colon	10	0.1667	0.9167	0.6667	0.3333
	30	0.1667	0.9167	0.6667	0.3333
	50	0.1768	0.9060	0.6767	0.3233
	70	0.1768	0.9060	0.6767	0.3233
	100	0.3333	**1**	**0.7778**	0.2222
	130	0.4883	0.7708	0.7530	0.2470
	150	0.1667	1	0.7222	0.2778

	200	0.5000	0.8333	0.7222	0.2778
	1000	0.5000	0.8333	0.7222	0.2778
Wisconsin	10	0.9252	0.9206	0.9235	0.0765
	30	0.9626	0.9206	0.9471	0.0529
	50	0.9626	0.9206	0.9471	0.0529
	70	0.9810	0.9524	0.9706	0.0294
	100	0.9720	0.9841	0.9765	0.0235
	130	0.9813	0.9048	0.9529	0.0471
	150	0.9813	0.9048	0.9529	0.0471
	200	0.9813	0.9048	0.9529	0.0471
	1000	0.9813	0.9048	0.9529	0.0471
Pendigits	10	0.9990	0.9701	0.9665	0.0325
	30	0.9990	0.9701	0.9665	0.0325
	50	0.9985	0.9915	0.9759	0.0241
	70	0.9985	0.9915	0.9759	0.0241
	100	0.9995	0.9573	0.9875	0.0125
	130	0.9995	0.9829	0.9835	0.0165
	150	0.9990	0.9744	0.9831	0.0169
	200	0.9995	0.9829	0.9849	0.0151
	1000	0.9995	0.9829	0.9849	0.0151
Hepatite	10	0.3333	0.9722	0.8444	0.1556
	30	0.4444	0.8889	0.8000	0.2000
	50	0.6667	0.9444	0.8889	0.1111
	70	0.6667	0.8889	0.8444	0.1556
	100	0.6967	0.9722	0.9111	0.0889
	130	0.6867	0.9167	0.8667	0.1333
	150	0.5444	0.9722	0.8667	0.1333
	200	0.3333	0.9444	0.8222	0.1778
	1000	0.5556	0.8056	0.7556	0.2444
Heart	10	0.4615	0.9796	0.5909	0.4091
	30	0.5641	0.9388	0.5455	0.4545
	50	0.5128	0.8776	0.5341	0.4659
	70	0.5128	0.9184	0.5568	0.4432

	100	0.8980	1	0.6136	0.3864
	130	0.5228	0.8980	0.5568	0.4432
	150	0.5128	0.8980	0.5568	0.4432
	200	0.5897	0.9388	0.5682	0.4318
	1000	0.4872	0.9388	0.5795	0.4205

TABLE III.2 Les résultats obtenus pour les cinq BDD pour la méthode Bagging

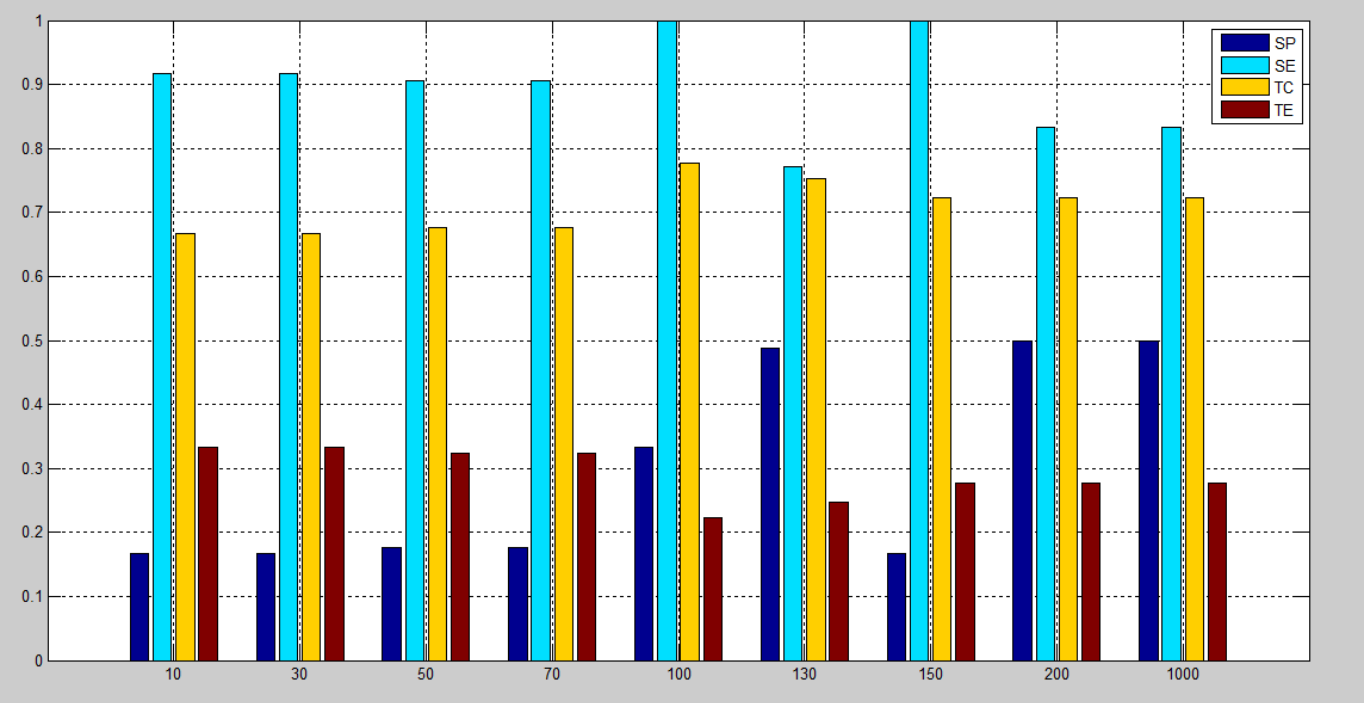


FIGURE III.1 résultats obtenus pour la BDD colon pour la méthode bagging

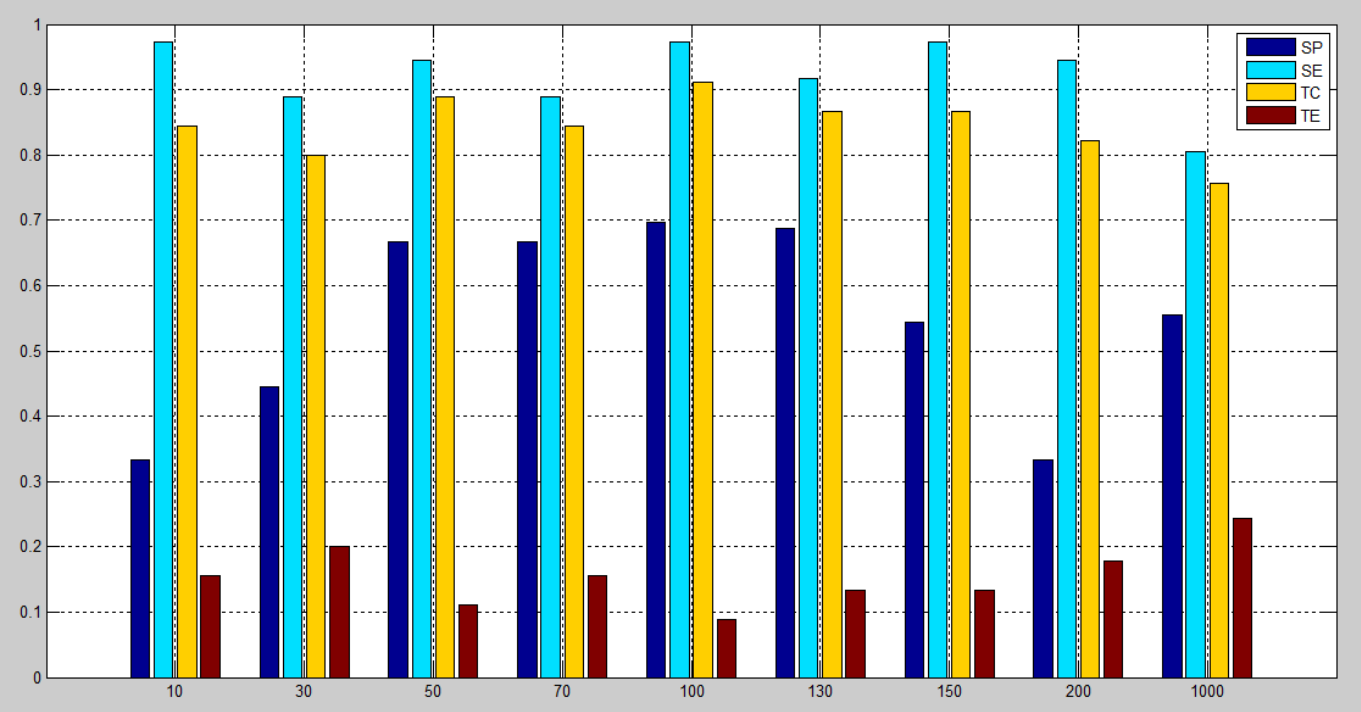


FIGURE III.2 résultats obtenus pour la BDD hepatite pour la méthode bagging

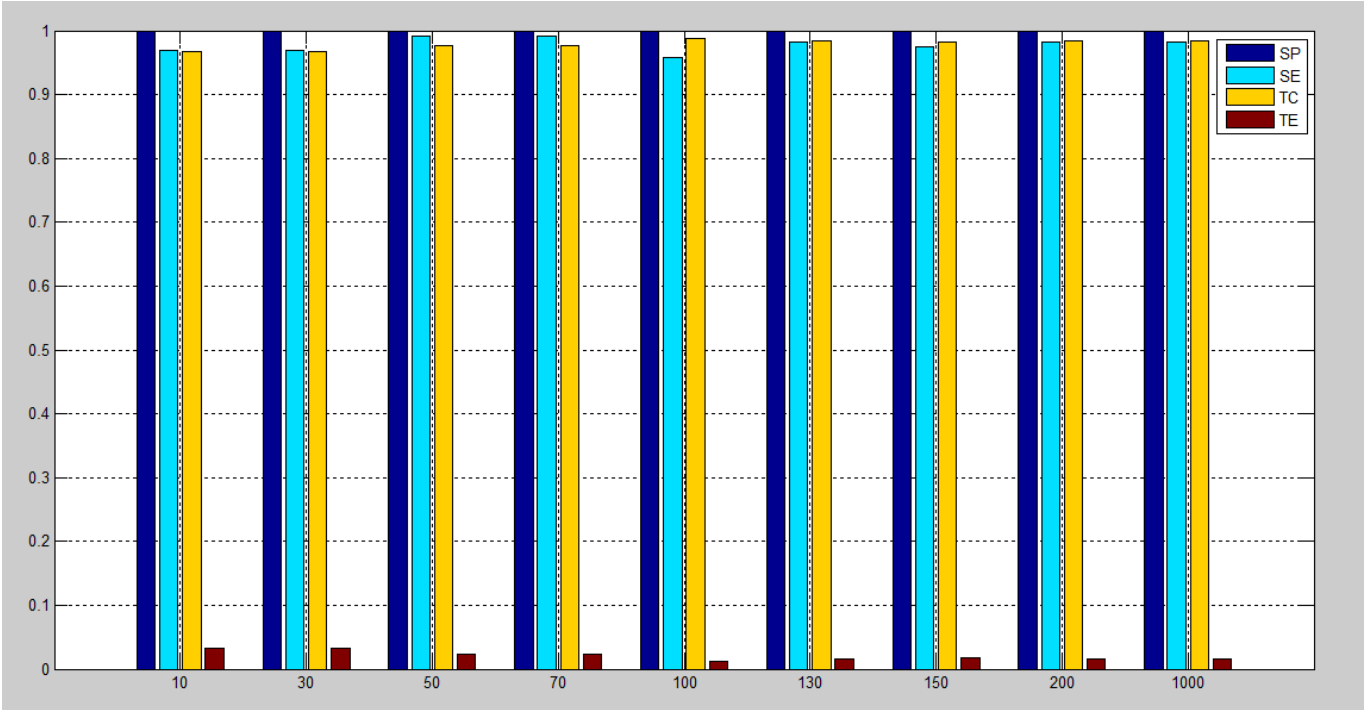


FIGURE III.3 résultats obtenus pour la BDD pendigits pour la méthode bagging

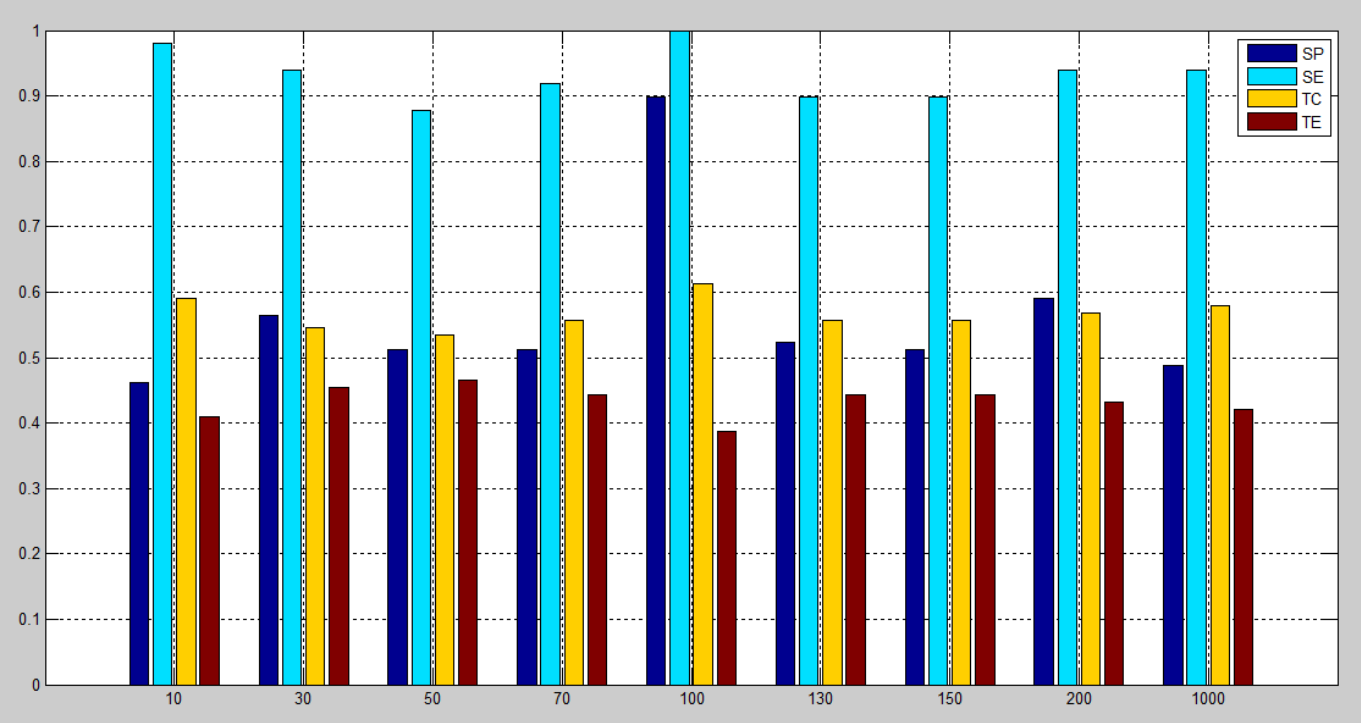


FIGURE III.4 résultats obtenus pour la BDD heart pour la méthode bagging

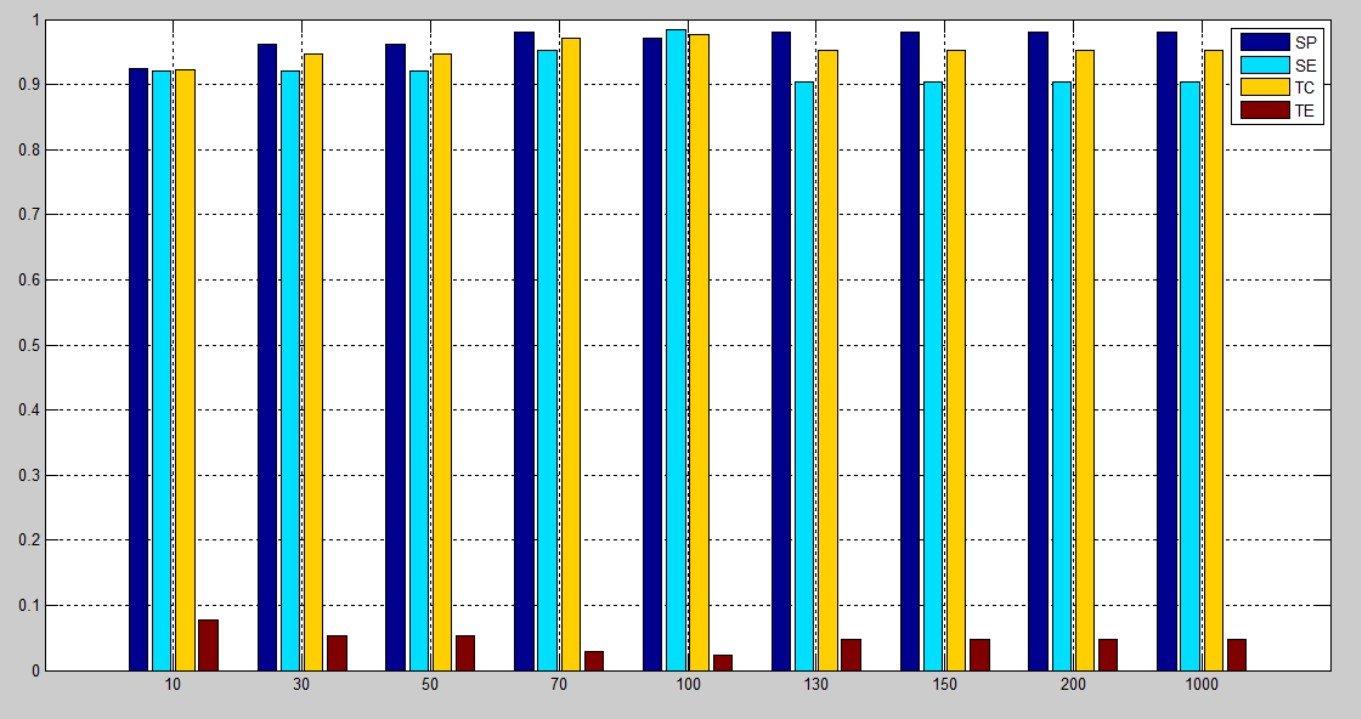


FIGURE III.5 résultats obtenus pour la BDD Wisconsin pour la méthode bagging

III.5.1.2 Interprétation des résultats concernant le bagging

a. Pour les résultats concernant la base de donnée colon on remarque que lorsque le nombre des bags est 100 on a le meilleur taux de classification 77.78%, la sensibilité est élevée (100%) ce qui vaut dire que le système donne une bonne connaissance pour les données positives par contre il donne un faible apprentissage pour les données négatives car la spécificité est diminué mais elle augmente (50%) lorsque on atteint 200 bags.
La sensibilité ; la spécificité et le taux de classification se stabilisent et restent constants dés les 200 bags.

b. Pour les résultats concernant la base de donnée Wisconsin on remarque que lorsque le nombre des bag est 100 on a le meilleur taux de classification 97.65%, la sensibilité est élevée (98.41%) ce qui vaut dire que le système donne une bonne reconnaissance pour les données positives, une bonne spécificité alors il donne un bon apprentissage pour les données négatives, elle est élevée (98.13) au niveau de 130 bag. la sensibilité ; la spécificité et le taux de classification se stabilisent et restent constants dés les 130 bags.

c. Pour les résultats concernant la base de donnée Pendigits on remarque que lorsque le nombre des bags est 100 on a le meilleur taux de classification 98.75%, la sensibilité est élevé (95.73%) ce qui vaut dire que le système donne une bonne reconnaissance pour les données positives , une bonne spécificité alors il donne un bon apprentissage pour les données négatives ,elle est élevée (99.95) au niveau de 100 bags; la spécificité et la sensibilité et le taux de classification se stabilisent dés les 200 bags

d. Pour les résultats concernant la base de donnée heart on remarque que lorsque le nombre des bags est 100 on a le meilleur taux de classification 61.36%, la sensibilité est élevé (100%) ce qui vaut dire que le système donne la meilleure reconnaissance pour les données positives , une bonne spécificité alors il donne un bon apprentissage pour les données négatives ,elle est élevée (89.80) au niveau de 100 bags; la spécificité et la sensibilité et le taux de classification restent toujours instables.

e. Pour les résultats concernant la base de donnée hepatitet on remarque que lorsque le nombre des bags est 100 on a le meilleur taux de classification 91.11%, la sensibilité est élevé (97.22%) ce qui vaut dire que le système donne la meilleure reconnaissance pour les données positives , une bonne spécificité alors il donne un bon apprentissage pour les données négatives ,elle est élevée (69.67) au niveau de 100 bags; la spécificité et la sensibilité et le taux de classification restent toujours instables.

Rapport-Gratuit.com

III.5.2 Les résultats de boosting « AdaBoostM1 » au niveau des bases de données

BDD	Nombre d'itération	SP	SE	TC	TE
Wisconsin	**10**	0,9813	0,9206	0,9588	0,0412
	20	0,9907	0,9683	0,9822	0,0176
	30	0,9813	0,9524	0,9706	0,0294
	40	1	0,9524	0,9824	0,0176
	50	0,9813	0,9206	0,9588	0,0412
	100	0,9813	0,9206	0,9706	0,0294
	150	0,9813	0,9206	0,9588	0,0412
Colon	**10**	0.3333	0.9167	0.7222	0.2778
	20	0.6667	0.7500	0.7222	0.2778
	30	0.5000	0.9167	0.7778	0.2222
	40	0.6250	0.9375	0.8333	0.1667
	50	0.1667	0.8363	0.6667	0.3333
	100	0.1667	0.9167	0.6667	0.3333
	150	0.5000	0.7500	0.6667	0.3333
Pendigits	**10**	80.43	81.23	0.8855	0.1145
	20	81.00	82.45	0.8864	0.1106
	30	91.00	91.45	0.8811	0.1189
	40	96.22	98.20	0.8886	0.1114
	50	95.22	96.02	0.8846	0.1154
	100	90.44	90.50	0.8855	0.1145
	150	91.88	92.13	0.8811	0.1106
Hepatite	**10**	0.6667	0.8333	0.8000	0.2000
	20	0.4444	0.9444	0.8444	0.1556
	30	0.7778	1	0.9556	0.0444
	40	0.6667	0.9444	0.8889	0.1111
	50	0.6667	0.9167	0.8667	0.1333
	100	0.6667	0.9722	0.9111	0.0889
	150	0.4444	0.8333	0.7556	0.2444
heart	**10**	80.02	79.22	0.8295	0.1705
	20	79.02	79.00	0.8523	0.1477
	30	80.20	81.22	0.8409	0.1591

	40	81.23	81.13	0.8750	0.1250
	50	90.00	90.17	0.8977	0.1023
	100	77.98	78.79	0.7955	0.2045
	150	85.45	88.15	0.8523	0.1477

TABLE III.3Les résultats obtenus pour les cinq BDD pour la méthode Boosting.

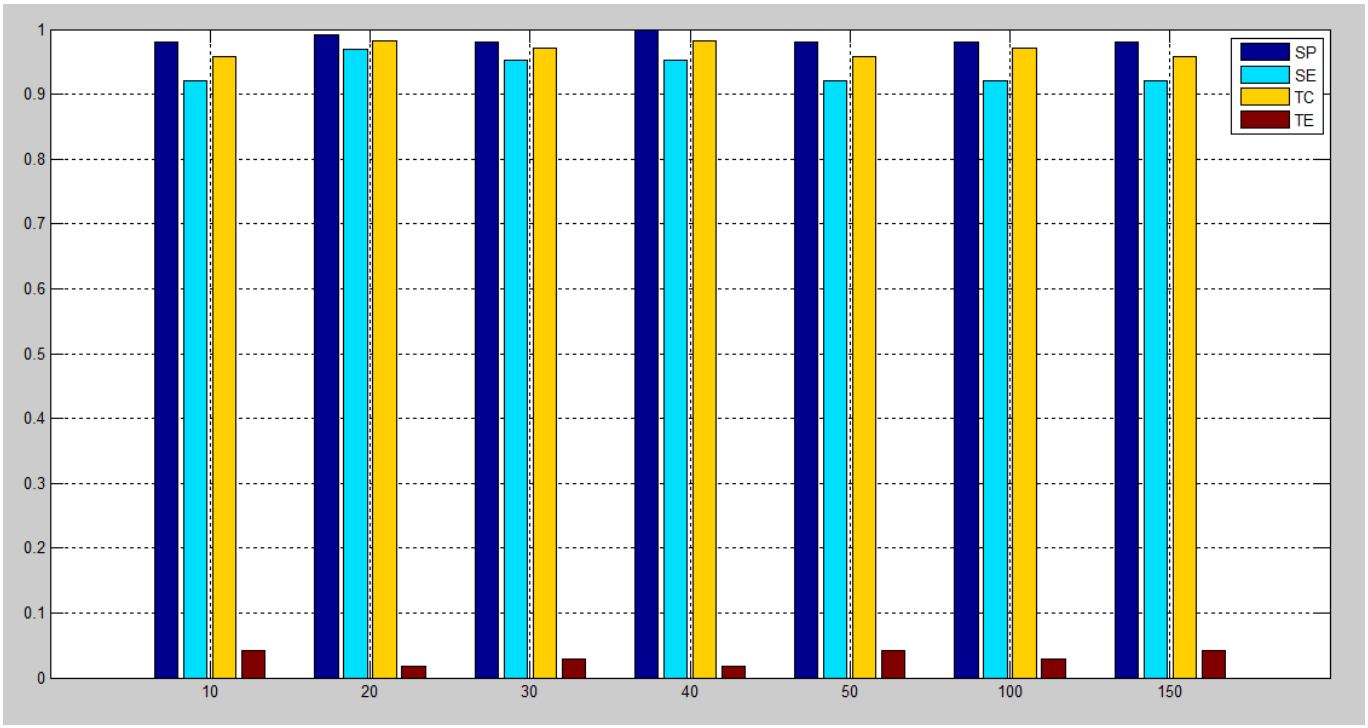


FIGURE III.6 résultats obtenus pour la BDD Wisconsin pour la méthode boosting

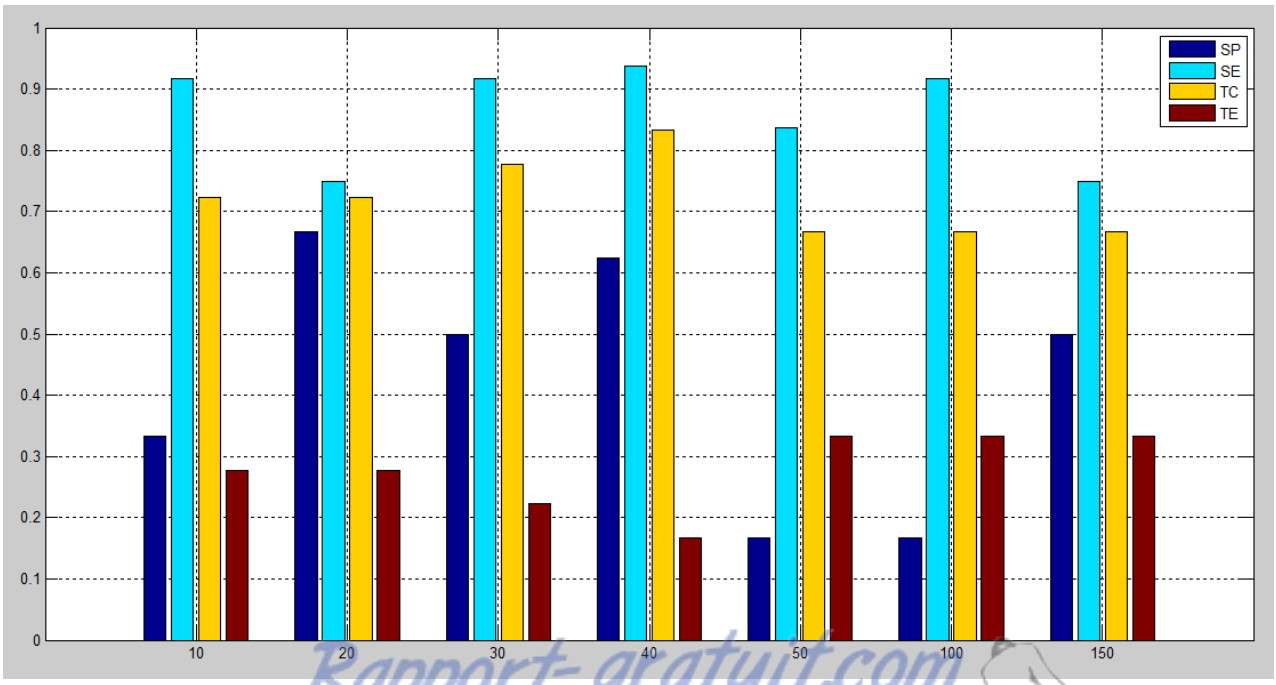


FIGURE III.7 résultats obtenus pour la BDD Colon pour la méthode boosting

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

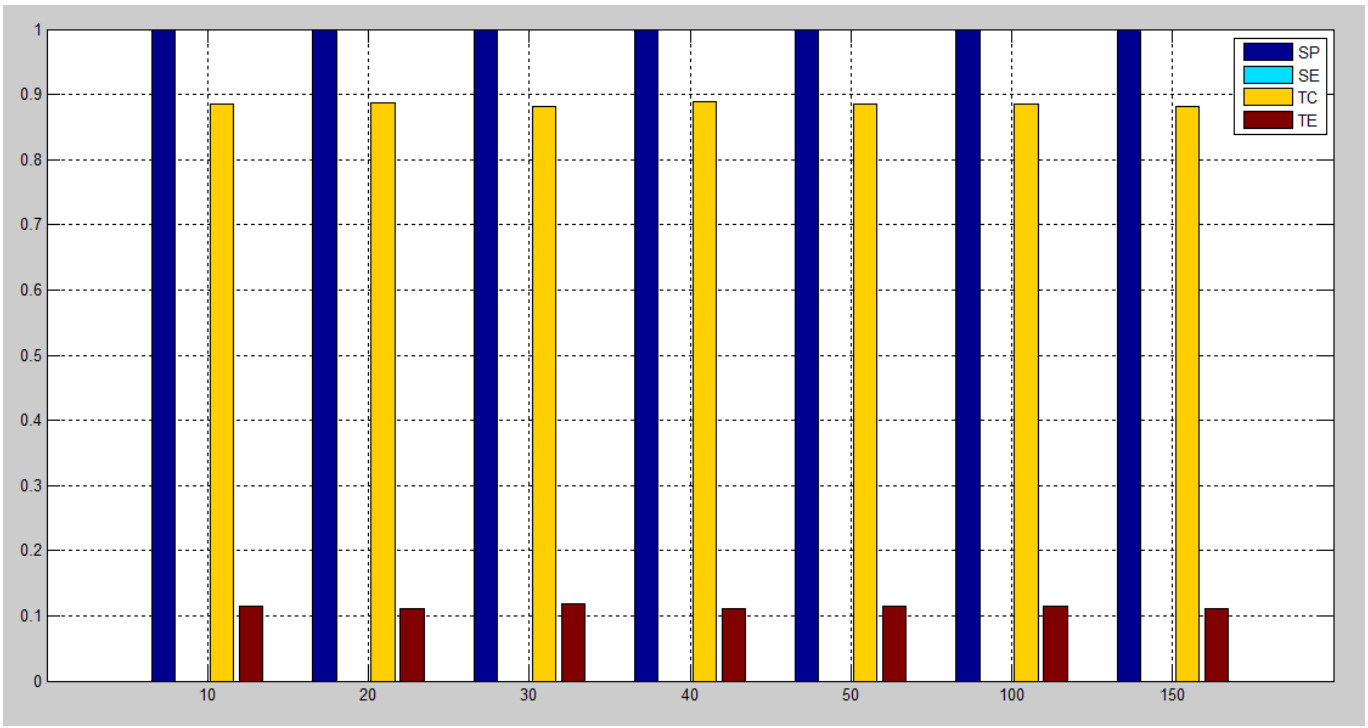


FIGURE III.8 résultats obtenus pour la BDD pendigits pour la méthode boosting

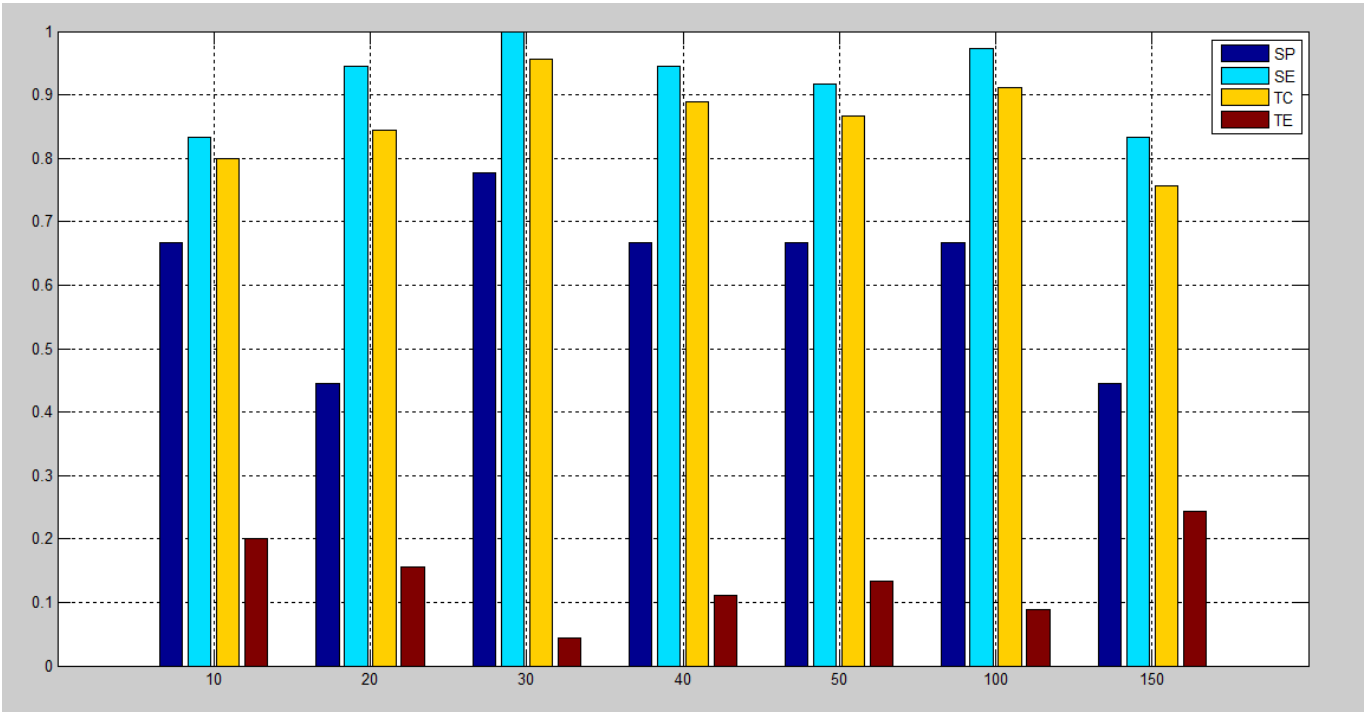


FIGURE III.9 résultats obtenus pour la BDD hepatite pour la méthode boosting

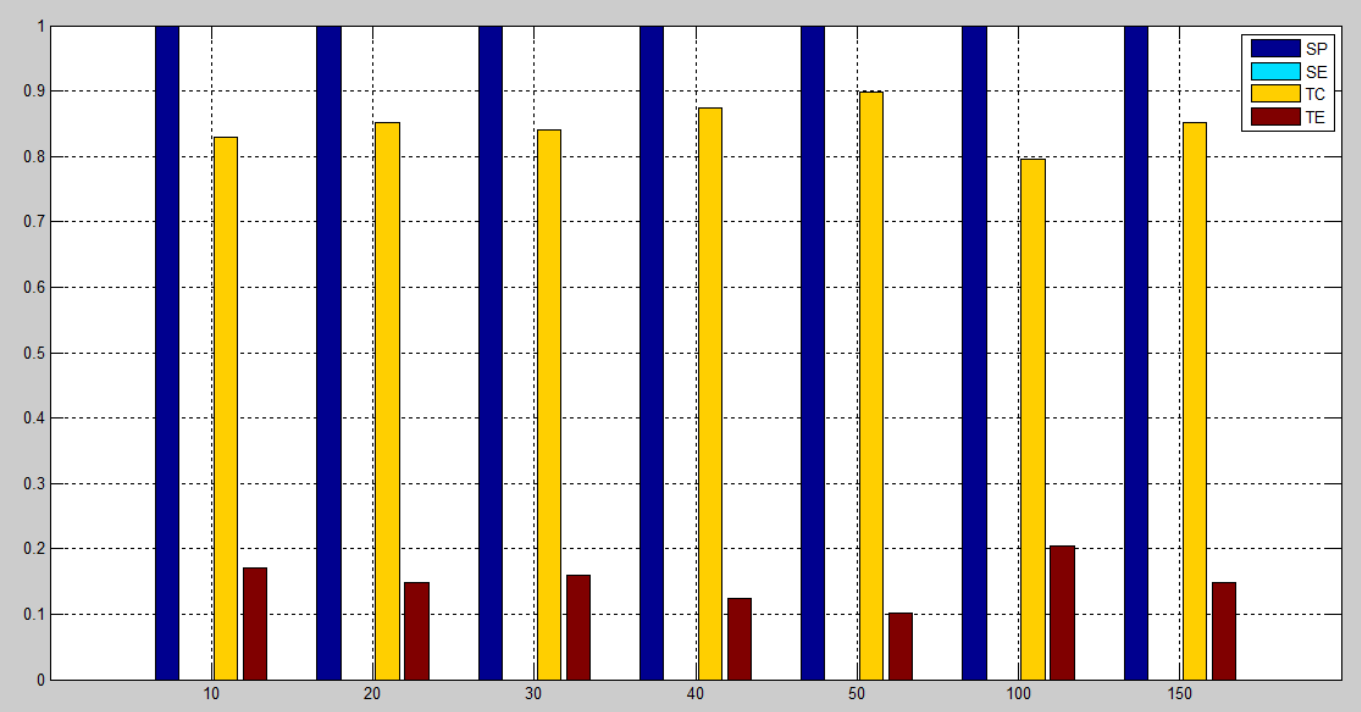


FIGURE III.10 résultats obtenus pour la BDD heart pour la méthode boosting

III.5.2.1 Interprétation des résultats concernant le boosting « AdaBoostM1 »

a. Pour les résultats concernant la base de donnée Colon on remarque que lorsque le nombre des itérations est 40 on a le meilleur taux de classification 83.33%, la sensibilité est élevé (83,2%) ce qui vaut dire que le système donne une bonne reconnaissance pour les données positives la même remarque pour la spécificité elle donne un meilleur apprentissage pour les données négatives (62.50%). Le taux de classification se stabilise et reste constants dés les 50 itérations par contre la sensibilité, la spécificité sont toujours instables.

b. Pour les résultats concernant la base de donnée Wisconsin on remarque que lorsque le nombre des itérations est 40 on a le meilleur taux de classification 98.24%, la sensibilité est élevée (93,75%) ce qui vaut dire que le système donne une bonne reconnaissance pour les données positives la même remarque pour la spécificité elle donne un bon apprentissage pour les données négatives (100%). la sensibilité ; la spécificité se stabilisent et restent constants dés les 50 itérations par contre le taux de classification est toujours instable.

c. Pour les résultats concernant la base de donnée pendigits on remarque que lorsque le nombre des itérations est 40 on a le meilleur taux de classification 88.86%, la spécificité elle donne meilleur apprentissage pour les données négatives la sensibilité donne une bonne reconnaissance pour les données positives ; la spécificité et la sensibilité ne se stabilisent pas et restent toujours instables au niveau de toutes les itérations et même pour le taux de classification est toujours instable.

d. Pour les résultats concernant la base de donnée hépatite on remarque que lorsque le nombre des itérations est 50 on a le meilleur taux de classification 95.56%, la sensibilité est bonne (100%) ce qui vaut dire que le système donne une reconnaissance pour les données positives et on remarque que pour la spécificité elle donne meilleur apprentissage pour les données négatives (77.78%). la sensibilité ; la spécificité se stabilisent et restent constants au niveau de toutes les itérations par contre le taux de classification est toujours instable.

e. Pour les résultats concernant la base de donnée heart on remarque que lorsque le nombre des itérations est 50 on a le meilleur taux de classification 89.77%, la sensibilité est élevée ce qui vaut dire que le système donne une reconnaissance pour les données positives la même remarque pour la spécificité elle donne un meilleur apprentissage pour les données négatives . Le taux de classification la sensibilité et la spécificité sont toujours instables et après les 50 itérations

III.5.3 Les résultats de random forest au niveau des bases de données

BDD	Nombre d'arbre	SP	SE	TC	TE
Colon	**10**	1	0.8182	0.8462	0.1538
	30	0.5000	0.8571	0.6923	0.3077
	50	0.4444	1	0.6154	0.3846
	70	1	0.8003	0.6262	0.3738
	150	0.4444	1	0.6154	0.3846
	170	1	0.8182	0.8462	0.1538
Wisconsin	**10**	0.8857	0.8448	0.8672	0.1328
	30	0.9412	0.8833	0.9141	0.0859
	50	0.9706	0.9167	0.9453	0.0547
	70	0.8987	1	0.9375	0.0625
	90	0.9692	0.8730	0.9219	0.0781
	100	0.9286	0.8966	0.9141	0.0859
	150	0.8875	1	0.9297	0.0703
	200	0.9545	0.8710	0.9141	0.0859
	250	0.7889	1	0.8516	0.1484
Pendigits	**10**	0.8660	1	0.8415	0.1585
	30	0.7779	0.7580	0.8073	0.1927
	50	**0.9647**	0.9399	**0.9518**	0.1471
	100	0.9625	**0.9499**	0.9499	0.2485
	150	0.7960	0.8330	0.6748	0.3252
	170	0.8700	0.7000	0.6685	0.3315
	10	0.4286	0.9630	0.8529	0.1471

Hepatite	**30**	0.1429	0.8889	0.7353	0.2647
	50	1	0.9677	0.9706	0.0294
	100	0.8500	0.9667	0.9412	0.0588
	150	0.7500	0.8750	0.8235	0.1765
	170	0.8330	0.9333	0.5556	0.4444
Heart	**10**	0.6923	0.7568	0.7302	0.2698
	30	0.8095	0.7500	0.7692	0.2308
	50	0.6875	0.6667	0.6719	0.3281
	100	0.8000	0.7500	0.7656	0.2344
	150	0.8095	0.7674	0.7813	0.2187
	170	0.6667	0.6739	0.6719	0.3281

TABLE III.4 Les résultats obtenus pour les cinq BDD pour la méthode Random forest.

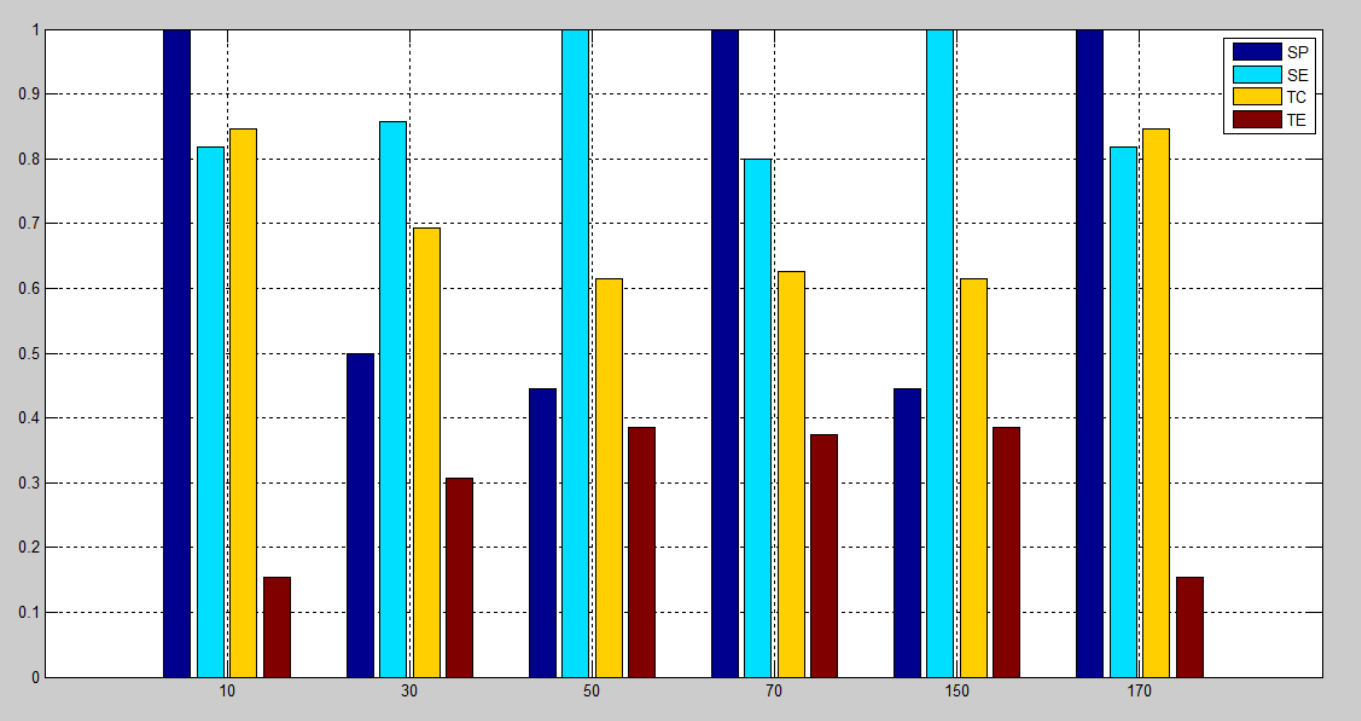


FIGURE III.11 résultats obtenus pour la BDD Colon pour la méthode RF

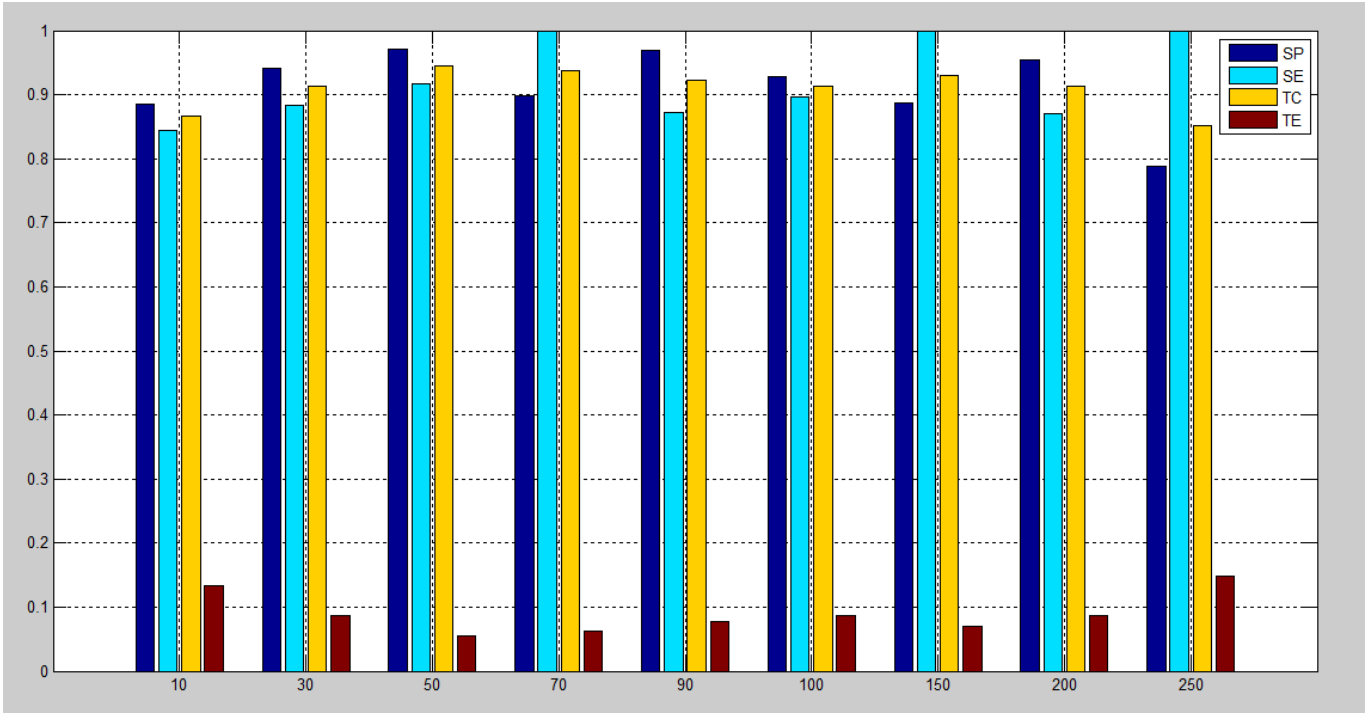


FIGURE III.12 résultats obtenus pour la BDD Wisconsin pour la méthode RF

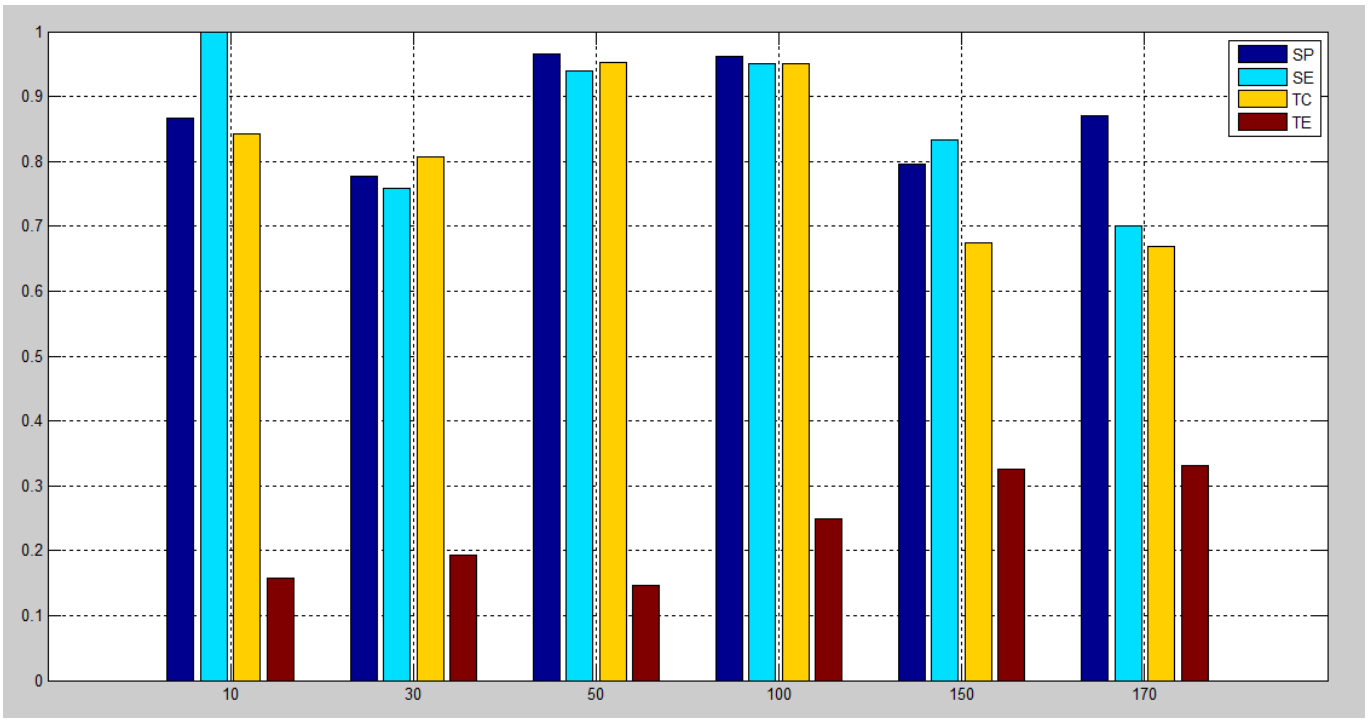


FIGURE III.13 résultats obtenus pour la BDD pendigits pour la méthode RF

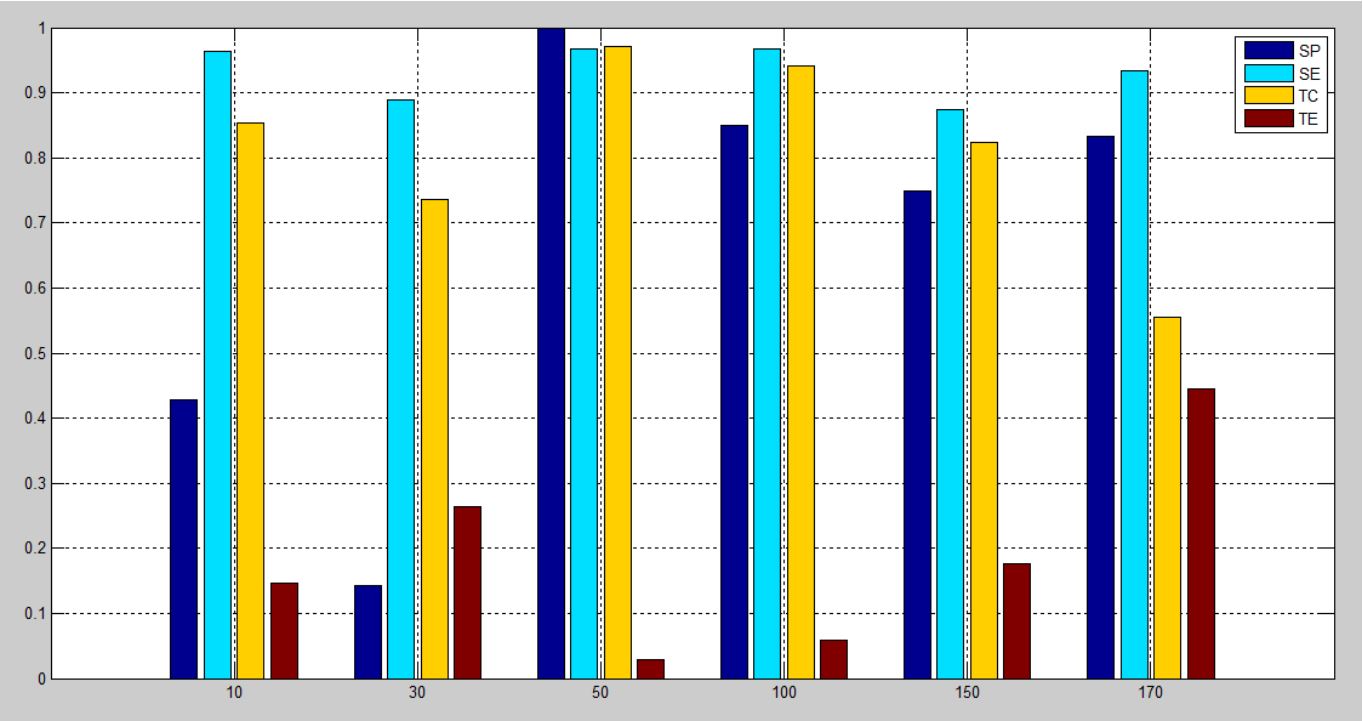


FIGURE III.14 résultats obtenus pour la BDD hepatite pour la méthode RF

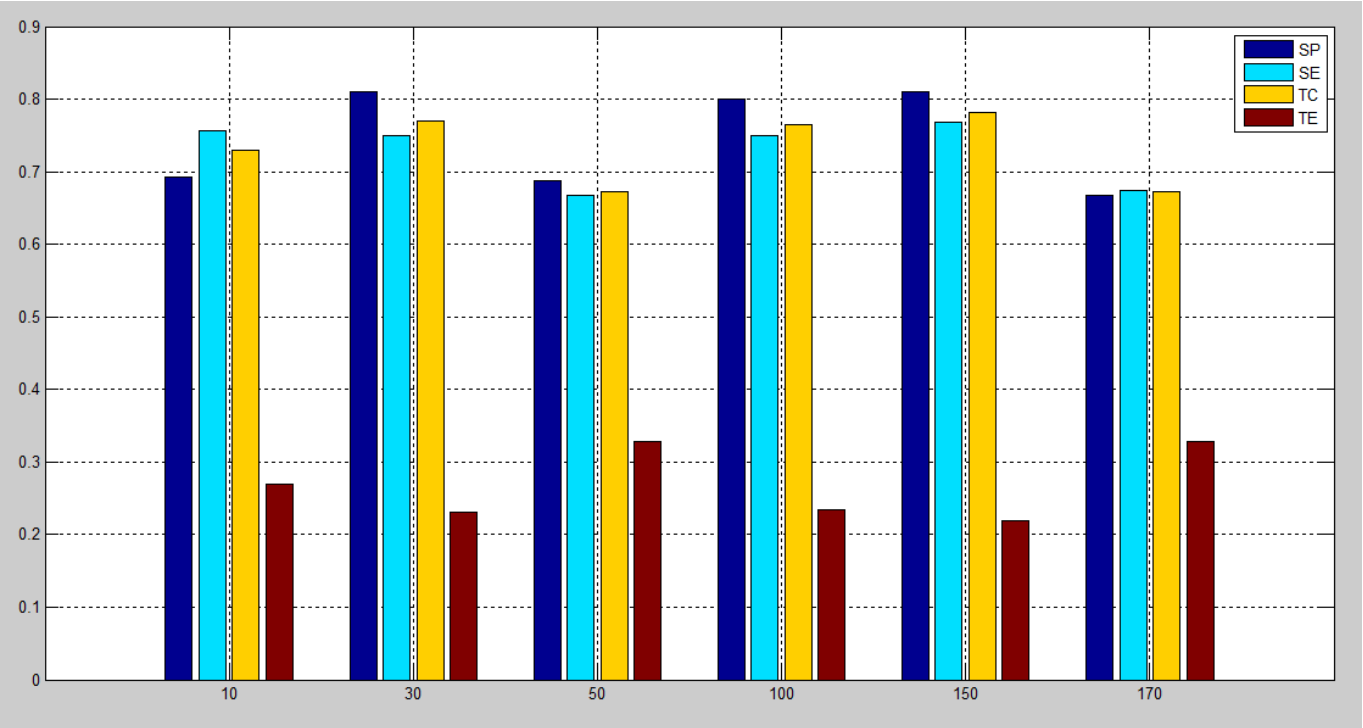


FIGURE III.15 résultats obtenus pour la BDD heart pour la méthode RF

III. 5.3.1 Interprétation des résultats

On remarque que les deux bases de données ont un bon de taux de classification au niveau de 50 arbres (Pendigits ; tc=85.29%), (colon ; tc=75%), (hepatites ; tc=97.06%), (Wisconsin ; tc=99.43%), (heart ; tc=78.13%) au niveau de 150 arbres.

III.6. Comparaison des résultats

Bases de données	Meilleur taux de classification bagging	Meilleur taux de classification AdaBoostM1	Meilleur taux de classification Rf
Wisconsin	97,65	98,24	99,43
Pendigits	98,75	88,86	85,29
Colon	77,78	83,33	85.71
Heart	61.36	89.77	78.13
Hepatite	91.11	97.22	97.06

TABLE III.5 Comparaison des résultats

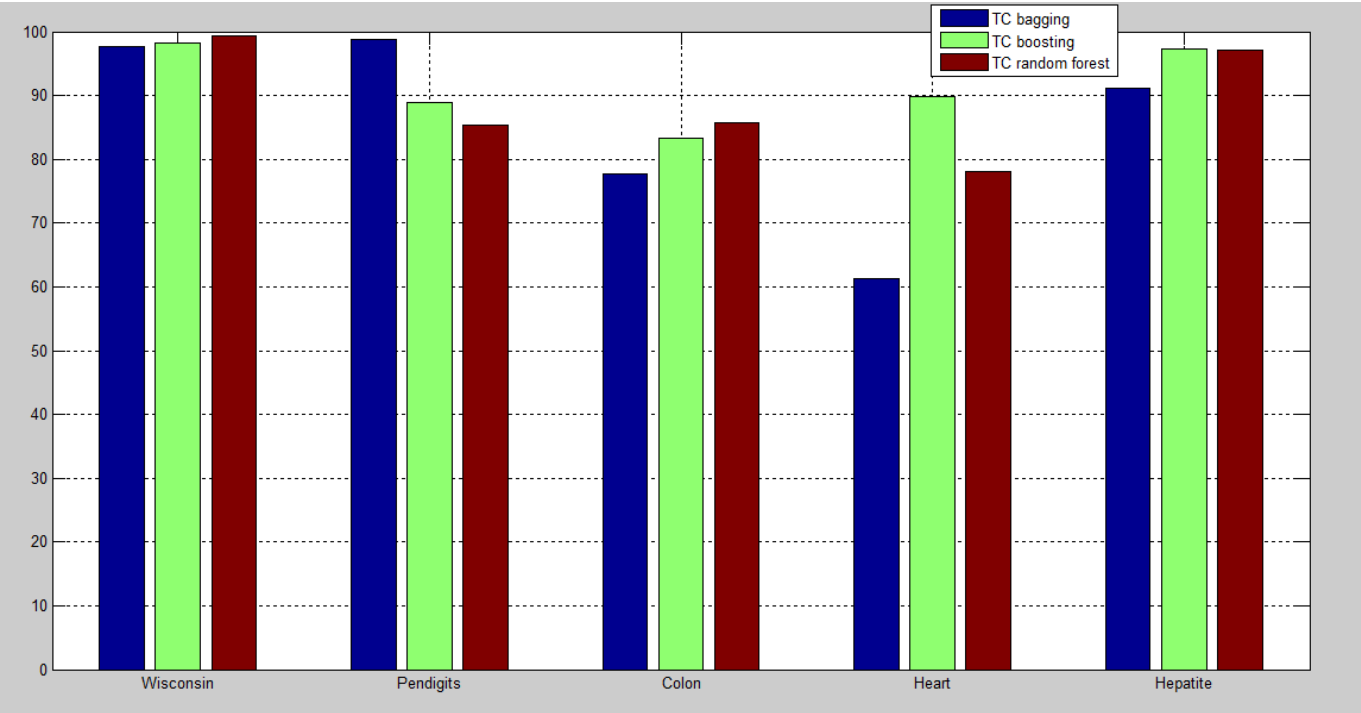


FIGURE III.16 résultats obtenus pour la comparaison des résultats pour les trois méthodes

III.6.1. Interprétation général des résultats

Adaboost est plus performant que le bagging et qu'il est moins risqué au cas du sur-apprentissage, car Adaboost essaye directement d'optimiser les votes pondérés Contrairement au Bagging qui demande un grand nombre de bags pour que l'erreur en généralisation se stabilise,

- ✓ Les deux mesures de taux de classification et le taux d'erreur montrent que pour les 2 bases (Wisconsin, colon) l'adaboost est plus performant que le bagging. Cela du a son principe même qui La sensibilité est élevée avec l'utilisation d'adaboost (avec la base de données Colon), ce qui confirme sa performance sur les instances minoritaire.
- ✓ La spécificité a donnée des bons résultats pour le bagging avec 2 bases (wisconsin, Pendigits), ce qui démontre que le bagging donne l'importance à la classe majoritaire.
- ✓ Les résultats de bagging et de boosting est meilleur que celles de random forest sur les trois bases de données (Pendigits, Colon), alors que le meilleur taux de classification pour la base de donnée Wisconsin est pour randon forest.

III.7. L'exécution des trois méthodes d'ensembles sur les 5 bases de données dans le même temps

III.7.1 L'expérimentation 02 : l'exécution des trois méthodes d'ensembles au niveau de Wisconsin et avec les conditions pour obtenir le meilleur taux de classification pour chaque base de donnée(le nombre d'itération=le nombre des bags= le nombre des arbres)

La base de données	adaBoostM1	Bagging	RandomForest
Wisconsin	Tc=96 ,47 Se=90,48 Sp=100	Tc=96,47 Se=93,65 Sp=98,13	Tc=99.1484 Se=100 Sp=97,06
Pendigits	Tc=88,15 Se=0 Sp=100	Tc=98,66 Se=99,15 Sp=100	Tc=95,18 Se=94,99 Sp=96,47
Colon	Tc=72,22 Se=100 Sp=16,67	Tc=66,67 Se=83,33 Sp=33,33	TC=84,62 Sp=100 Se=100
Heart	Tc=88.07 Se=100 Sp=0	Tc=84.67 Se=90.75 Sp=68.11	TC=76,56 Se=82,32 Sp=77,89

Rapport-Gratuit.com

Hepatite	Tc=95.49 Se=98.11 Sp=82.07	Tc=90.97 Se=85.63 Sp=70.12	TC=97.06 Se=100 Sp=96.02

TABLE III.6 Comparaison des résultats

III.8. Interprétation des résultats

- **Les résultats restent les mêmes** Les résultats de bagging et de boosting est meilleur que celles de random forest sur les 4 bases de données (Pendigits, Colon,,heart, hepatite,), alors que au paravent le meilleur taux de classification pour la base de donnée Wisconsin est par randon forest.la sensibilité et la spécificité et même le taux de classification est instables par rapport aux résultats précédentes appliquées dans les mêmes conditions.

- Les comparaisons qui suivent ont toutes été réalisés sur des jeux de données réeles ou synthétiques le bagging et le boostinge qui ont les méthodes les plus comparées .tous les auteures s'entendent sur le fait que le boosting est plus performant que le bagging (Frend and shapire[1996],quinlan[1996],breiman[1998],bauer and Kohavi[1999],Opitz and richard[1999]. breiman[1998] soutint que ceci est du au fait que le boosting réduit d'avantage la variance par contre la performance de bagging est constante alors que les résultats varient beaucoup plus pour le boosting qui peut faire diminuer radicalement l'erreur sur un ensemble.

- L'application des trois méthodes sur chaque base de données dans les memes conditions on remarque que le temps d'éxécution se differt d' un classifieur à un autre .la spécificité et la sensibilite se changent meme si on avaite le meme taux de classification obtenu dans les mems conditions premièrement.

- Il y avait un certain nombre de bags(arbre de décision C4.5) au niveau de bagging et un certain nombres des arbres au niveau de random forest et un certain nombre d'itération au niveau de AdabootM1 considérées que ce sont les meilleures nombres à choisir car au niveau de ces dernieres on obtient le meilleur taux de classification et meme une bonne sensibilité et spécificité et si on dépasse ces nombres les résultats seront totalement instables.

III.9 Conclusion

Ce chapitre, présente l'application de trois méthodes existantes dans la classification ensembliste sur les différentes bases de données qui ont deux classes et au plus. Des expériences réalisés pour justifier la performance de ces algorithmes ainsi que Adaboost qui a été bien adaptée aux bases données, elle a donné des meilleures performances pour les deux classes (minoritaire, majoritaire).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Conclusion générale

Nous avons réalisé dans ce projet de fin d'étude une étude comparative entre les différentes méthodes d'ensemble (bagging, boosting, forets aléatoires.).ces trois méthodes améliorent de manière significative les performances obtenues en classification par rapport à l'utilisation d'un classifieur unique. Par opposition à l'apprentissage "classique" où un seul classifieur est entrainé, Nous sommes intéressés dans ce travail à étudier la performance des modèles ensemblistes sur une tâche de classification reliée au domaine médical Pour cela nous avons utilisé 5 bases de données d'UCI : breast-cancer, hepatite, colon(bases a 2 classes) heart, pendigits(plusieures classes.)Pour évaluer nos trois modèles.

Les résultats obtenus, nous ont montré que la combinaison des classifieurs différents peut améliorer d'une manière significative les performances.

Comme perspective, nous proposons d'utiliser le SMOTE (***Synthetic Minority Oversampling Technique)*** pour le prétraitement des données.

Bibliographies

[1] FREUND Y., SCHAPIRE R., Experiments with a New Boosting Algorithm, *International Conference on Machine Learning*, pp. 148–156, 1996.

[2] BREIMAN L., Arcing classifiers, *The Annals of Statistics*, vol. 26, no 3, pp. 801–849, 1998.

[3] BREIMAN L., Bagging Predictors, *Machine Learning*, vol. 24, no 2, pp. 123–140, 1996.

[4] HO T., The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no 8, pp. 832–844, 1998.

[**5**] BREIMAN L., Random Forests, *Machine Learning*, vol. 45, no 1, pp. 5–32, 2001

[6] : Sami FAYE-CHELLALI, Optimisation du processus prédictif de sélection médicale en prévoyance individuelle Application au télémarketing, p15, Soutenance de mémoire d'actuariat, 07 novembre 2016.

[7] : Philippe BESSE, Hélène MILHEM, Olivier MESTRE, Anne DUFOUR, Vincent-Henri PEUCH, Comparaison de techniques de « Data Mining » pour l'adaptation statistique des prévisions d'ozone du modèle de chimie-transport MOCAGE, p292
.
[8] : Thomas François Aspirant FNRS, Modèles statistiques pour l'estimation automatique de la difficulté de textes de FLE, p7, article, Senlis, 24–26 juin 2009.

[9] : Ritte Jean-François, Détection de piétons en milieu routier urbain, poster ;

[10] : Thèse par Chesner Desir , Classification Automatique d'Images, Application à l'Imagerie du Poumon Profond ,p54-59-69-169.

[11] : Henri-Maxime Suchier , Nouvelles contributions du boosting en apprentissage automatique p 51,52,53,54,55,79.

[12]: Alexis LECHERVY, Apprentissage interactif et multi-classes pour la détection de concepts sémantiques dans des données multimédia, p153-135, T H È S E pour obtenir le titre de Docteur en Sciences

[13]: Xin Li, one of combining models,p47-48-50-51-52-53.

[14]: Philippe Foucher, Classification automatique de défauts sur des images de tunnels par forêts d'arbres aléatoires , p02, article

[15]: Robin Genuer1et Jean-Michel, Arbres CART et Forêts aléatoires Importance et sélection de variables,p17-34.

[16]: Rimah Amami, 2 Dorra Ben Ayed, 3 Noureddine Ellouze, Application de la Méthode Adaboost à la Reconnaissance Automatique de la Parole, p5, 6,8 article extrait de livre(Grard), volume 8, N1, pp53-60, e-STA, copyright 2011 by see.

[17] : Héla Khoufi Zouari, Contribution a l'évaluation des méthodes de combinaison parallèle de classifieurs [27-28-29-30-31-32-33-34]

[18] Duin, R. and Tax, D. (2000). Experiments with classifier combining rules. In J. Kittler, F. R., editor, 1st International Workshop, Multiple Classifier Systems (MCS), Lecture Notes in Computer Science, volume 1857, pages 16-29, Cagliari, Italy. Springer, Berlin.

[19] Jain, A., Duin, R., and Mao, J. (2000). Statistical pattern recognition : a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4-37

[20] Kuncheva, L., Bezdek, J., and Duin, R. (2001). Decision templates for multiple classifier fusion : an experimental comparison. Pattern Recognition, 34(2):299-314.
Moobed, 1996] Moobed, B. (1996). Combinaison de classifieurs, une nouvelle approche. Informatique, Universite Paris sud, UFR Scientifique d'Orsay.

[21] Ho, T. (1992). A theory of multiple classifier systems and its application to visual word recognition. PhD thesis, Department of Computer Science, Suny at Buffalo, New York.

[22] Ruta, D. and Gabrys, B. (2000). An overview of classifier fusion methods. Computing and Information Systems, 7:1-10.

[23] Methods for combining heterogeneous sets of classifiers. 17th National Conference on Artificial Intelligence (AAAI 2000).

[24] Xu, L., Krzyzak, A., and Suen, C. (1992). Methods of combining
multiple classifiers and their applications to handwriting recognition.IEEE Transactions on Systems, Man, and Cybernetics, 22(3):418-435.

[25] Roli, F., Fumera, G., and Kittler, J. (2002a). Fixed and
trained combiners for fusion of imbalanced pattern classifiers. In 5th International Conference on Information Fusion, pages 278{284, Annapolis (Washington) USA.

[26] Rahman, A. and Fairhurst, M (2003). Multiple classifier decision combination strategies for character recognition: a review. Journal Document Analysis and Recognition JDAR, pages 166-194.

[27] Zouari, H. K. (2004). Contributiona l'évaluation des méthodes de combinaison parallele de classifieurs par simulation (Doctoral dissertation, Université de Rennes I).

[28] Ronald Fisher en 1918 Effective voting of heterogeneous classifiers. Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings (pp. 465– 476).

[29] Gama, J., & Brazdil, P. (2000). Cascade generalization. Machine Learning, p 41, 315–343.

[30] Amélioration des Forêts Aléatoires pour une Meilleure Prédiction Mostafa
El HABIB DAHO ET SETTOUTI Nesma Biomedical Engineering Laboratory Tlemcen University, Algeria conference

[31] doctorat informatique par BOUBLENZA Amina Coopération entre classifieurs hétérogènes pour la reconnaissance des données médicales 2017

[32] Giacinto, G., & Roli, F. (1997). Adaptive selection of image classifiers. ICIAP '97: Proc. of the 9th International Conference on Image Analysis and ProcessingVolume I (pp. 38–45). London, UK: Springer-Verlag.

[33]: David H. Wolpert: Stacked generalization. Neural Networks, 5(2):241-259 (1992).

[34] Hadji Zahra Apprentissage en distribution déséquilibrée par les méthodes d'ensemble encadré par HABIB DAHO Mostafa et Settouti Nesma

[35] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," Artif. Intell. Rev., vol. 26, no. 3, pp. 159–190, 2006.

[36] Skurichina, M. (2001). Stabilizing weak classi_ers. PhD thesis, Delft University of Technology, Delft, The Netherlands…. Page 2 star 2

[37] Avnimelech, R. and Intrator, N. (1999). Boosted mixture of experts: an ensemble learning scheme. Neural Compu-tation, 11:475{490.

[38] Breiman, L. (1996b). Bias, variance, and arcing classi_ers. Technical Report 460, UC-Berkeley, Berkeley, CA.

[39] Efron, B. and Tibshirani, R. (1993). An intro- duction to the bootstrap. Chaman and Hall.

[40] Skurichina, M. and Duin, R. (1998). Bagging for linear classi_ers. Pattern Recognition, 31(7):909{930.

[41] Breiman, L. (1996a). Bagging predictors. Machine Lear-ning, 24(2):123{140.

[42] Leo Breiman and Leo Breiman. Bagging predictors. In Machine Learning, pages 123–140, 1996.

[43] Magali CHAMPION. Contribution à la modélisation et l'inférence
de réseau de régulation de gènes. PhD thesis, Université Toulouse 3 Paul Sabatier, 2014.

[44] Henri-Maxime Suchier. Nouvelles Contributions du Boosting en Apprentissage Automatique. PhD thesis, université Jean Monnet de Saint-Etienne, 2006.

[45] HAMMYANI, A. (2014). Amélioration des forêts aléatoires: Application au diagnostic médical

[46], Thomas G Dietterich. Ensemble methods in machine learning. In Multiple classifier systems, pages 1–15. Springer, 2000.

[47] Emna BAHRI. Amélioration des méthodes adaptatives pour l'apprentissage
supervisé des données réelles. PhD thesis, Université lyon lumière 2, Faculté des Sciences Economiques et de Gestion, 2010.

[48] M. Lichman. UCI machine learning repository, 2013.

[49] Yoav Freund and Robert Schapire. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14(771-780) :1612, 1999.

[50] Leo Breiman and Leo Breiman. Bagging predictors. In Machine Learning, pages 123–140, 1996

[51] ; Pr. Fabien Moutarde ; Arbres de décision et Forêts aléatoires ; Centre de Robotique (CAOR) MINES ParisTech (ENSMP) PSL Research University ; Fév.2017 ;pages ; 15 ; 16 ; 17.

[52] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost : A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 40(1) :185–197, 2010.

[53] Professeurs Quentin Louveaux, Olivier Bruls, et Frédéric Nguyen, "INTRODUCTION A MATLAB", Université Liége, 2008.

[54] Hoens &Chawla2010. A hybrid algorithm applied to classify unbalanced data. In Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on, pages 618–621. IEEE, 2010.

.