Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Abou Bekr Belkaid
Faculté de Technologie
Département de Génie Biomédical

MÉMOIRE DE FIN D'ÉTUDE

pour obtenir le grade de

Master en Génie Biomédical

Spécialité : **Informatique Biomédicale**

présenté et soutenu publiquement
par

Melle Douibi khalida

le 27 Mai 2015

Titre:

Classification Multi-labels des données médicales

Jury

Président du jury.	Pr. Chikh Mohamed Amine,	PROF UABB Tlemcen
Examinateur.	Dr. Benali Radhwane,	MCB UABB Tlemcen
Invité d'honneur.	Mr. Abderrahim Mohamed Amine,	MAA UABB Tlemcen
Encadreur.	Melle. Settouti Nesma,	MAA UABB Tlemcen
Co-Encadreur.	Mr. El Habib Daho Mostafa,	UABB Tlemcen

A cœur vaillant, rien d'impossible
A conscience tranquille, le tout est accessible
Quand il y a la soif d'apprendre, tout vient à point à qui sait attendre
Quand il y a le souci de réaliser un dessein, tout devient facile pour arriver à nos fins
Malgré les obstacles qui s'opposent, en dépit des difficultés qui s'interposent
Les études sont avant tout, notre unique et seul atout
Ils représentent la lumière de notre existence, l'étoile brillante de notre réjouissance
Espérant des lendemains épiques, un avenir glorieux et magique
Souhaitant que le fruit de nos efforts fournis Jour et nuit, nous mènera vers le bonheur fleuri
C'est à vous que je dédie ces mots, mes plus chers parents :

A Ma très chère mère affable, honorable, aimable : tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement, toi qui n'a pas cessé de m'encourager et de prier pour moi.

A Mon très cher père, aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit des sacrifices que vous avez consentis pour mon éducation et ma formation. Je vous dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A Mon cher frère, les mots ne suffisent pas pour exprimer l'attachement, l'amour et l'affection que je te porte. Mon fidèle compagnon dans les moments les plus délicats de cette vie mystérieuse.

A Ma très chère sœur, tu es la sœur que chacun rêverait d'avoir.Une pensée remplie de tendresse pour marquer notre belle complicité, je te dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

A Tous les membres de ma famille, petits et grands, mes chers grand parents, mes chers tantes et oncles à qui je souhaite le parfait bonheur, Veuillez trouver dans ce modeste travail l'expression de mon affection.

A Toutes personnes que j'ai rencontrées ici à Tlemcen depuis ma première année, à vous qui m'avez encourager :
Ma très chère Melle. Hammouni souad et sa famille, Mr. Saidi fodil et sa sœur souad, mon amie Amina et sa famille, je vous considère comme ma deuxième famille avec laquelle j'ai partagé tous mes soucis. Mon amie Asma et sa famille, tous les membres du groupe GDG et WTM Tlemcen, mes chers collègues du Master IBM surtout Soumia et El Batoul. *Grâce* à vous j'ai vécu des moments inoubliables qui resteront gravés dans ma mémoire.

A Ma tres chère enseignante Madame SAHRAOUI

A Ma très chère encadrante Settouti Nesma, vous *êtes* ma source d'inspiration. Votre aide, Votre générosité et votre soutien ont été pour moi une source de courage et de confiance.

Khalida

Remerciements

Au terme de ce travail, je saisis cette occasion pour exprimer mes vifs remerciements à toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail.

Je tiens à remercier notre chère et dynamique encadreur ***Melle. SETTOUTI Nesma*** en tant que Directeur de ce mémoire de fin d'études de Master, qui s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce projet, ainsi que pour son inspiration, son aide et le temps qu'elle a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Mes remerciements s'adressent également à ***Mr. El HABIB DAHO Mostafa***, notre aimable Co-encadreur pour sa présence et son soutien. Que ce travail soit un témoignage de ma gratitude et de mon profond respect.

J'exprime également ma gratitude aux membres du jury, ***Mr. CHIKH Mohamed Amine***, Professeur à l'Université de Tlemcen et ***Mr. BENALI Radhwane***, Maître de conférences à l'Université de Tlemcen qui nous honorent en acceptant de juger ce modeste travail.

Je désire adresser mes remerciements à l'ensemble du corps enseignant de ***la Filière de Génie Biomédical***, et un remerciement particulier à toute l'équipe CREDOM, qui m'a accueillie et partagé avec moi tous les bons moments pendant la réalisation de ce projet.

Je souhaite adresser ici mes sincères remerciements à nos professeurs de la spécialité ***Informatique biomédicale*** de nous avoir incités à travailler en mettant à notre disposition leurs expériences et leurs compétences.

Mes remerciements à : ***Mr. BECHAR Amine*** et ***Mr. TABERKIT Amine***, qui m'ont toujours soutenue et encouragée.

Résumé

Dans la classification supervisée traditionnelle, nous distinguons deux problématiques souvent rencontrées : la classification mono-labels qui associe à chaque exemple de l'apprentissage une classe et la classification multi-classes ayant un *même* but, mais à la différence plusieurs classes sont possibles dans la sortie et chaque exemple est associé à une seule classe.

Récemment, un nouvel axe de recherche a émergé qui attribue à chaque exemple de l'apprentissage une ou plusieurs classes simultanément, il s'agit de la ***Classification Multi-labels***. Ce nouveau domaine a attiré l'attention de nombreux chercheurs.

Dans le domaine médical, l'application de la classification multi-labels s'avère très intéressante puisque sur le terrain, les patients peuvent être atteints de plusieurs pathologies simultanément. Cette problématique nécessite des méthodes de la classification multi-labels pour pouvoir la résoudre, puisque les méthodes de classification mono-labels se limitent à affirmer ou infirmer la présence d'une seule pathologie chez un patient. Il existe deux familles de méthodes : les méthodes de transformation et d'adaptation.

Dans ce projet de fin d'études Master IBM nous présentons les notions de bases de ce nouveau domaine, nous élaborons également une étude comparative entre deux algorithmes très intéressants appartenant à ces deux familles de méthodes en proposant des contributions se basant sur les méthodes d'ensembles par la création d'une méthode appelée *Bagged Mlknn* ainsi que *RAkEL Random Forest*.

Mots clés : Classification multi-labels, méthodes de transformation, méthodes d'adaptation, méthodes d'ensembles, *RAkEL*, *MLknn*, *RAkEL Random Forest*, *Bagged MLknn*.

Abstract

A large body of research in supervised learning deals with the analysis of single label data, where training examples are associated with a single label, we also distinguish another task called Multi-class classification where training examples may belong also to a single label despite there exist various labels in output vector. However, training examples in several application domains are often associated with a set of labels. Such data are called multi-label.

Recently, the issue of learning from multi-label data has attracted significant attention from a lot of researchers.

In the medical context, the application of multi label classification is very interesting since patients can have several pathologies simultaneously. In order to deal to this problem, multi-label classification methods are required, since the single labels classification methods are limited to affirm or deny the presence of only a single pathology in a patient.

In recent years, many different approaches have been developed to solving multi-label learning problems, and are regrouped into two main categories : adaptation methods and transformation methods.

In this work we present the basic concepts of this new field, we are developing a comparative study between two very interesting algorithms belonging to these two families of methods called : *Bagged Mlknn* and *RAkEL RF*.

Keywords : Multi-label classification, transformation methods, adaptation methods, Ensemble methods, *RAkEL*, *MLknn*, *RAkEL Random Forest*, *Bagged MLknn*.

Table des matières

Table des figures

Liste des tableaux

Glossaire

ACP : Analyse en Composantes Principales
Bagged MLKnn : Multi classifieurs MLknn
Bagging : Bootstrap et Aggregating
BDD : Banque de Données
BPMLL : Back Propagation Multi Label
BR : Binary Relevance
BRRF : Binary relevance Random forest
CC : Classifer chains
CLR : Calibrated Label Ranking
Dinstinct : Label-set Distinct
DMl knn : Dependent Multi-Label k-Nearest Neighbors
ECC : Ensembles of Classifier chains
EML : Ensemble of multi-label classifiers
EnML : Multi-Label Ensemble Learning
H2PC : Hybrid algorithm for Bayesian network structure learning
HOMER : Hierarchy Of Multilabel classifiers
IBLR : Instance Based Logistic Regression
KNN : K Nearest Neighbours
LC : Label Cardinality
LD : Label Density
Lift : Multi-label learning with label-specific features
LP : Label Powerset
LPRF : Label powerset Random forest
MAP : global maximum a posteriori
MLKNN : Multi Label K Nearest Neighbours
ML-Loc : Multi-Label learning using Local Correlation
MLNB : Multi-labels Naives Bayes
PCT : Predictive Clustering Trees
PCT RF : Predictive Clustering Tree Random forest
PCTs : Ensembles of Predictive Clustering Trees
PFE : Projet de Fin d'études
PPT : Pruned Problem Transformation
$Rakel_d$: Random k labelsets disjoint
$Rakel_o$: Random k labelsets overlapping
Rakel : RAndom k-labELsets
RF : Random Forest
RPC : Ranking By Pairwise Comparison
SADM : Systèmes d'Aide à la Décision Médicale
SVM : Support Vector Machine
TSA : Two Stage Architecture
TSCCM : Two Stage Classifier Chain Method

TSPCCM : Two Stage Pruned Classifier Chain Method
TSVM : Two Stage Voting Method

Introduction générale

Les systèmes d'aide à la décision médicale (SADM) sont des applications informatiques dont le but est de fournir aux cliniciens les informations décrivant la situation clinique d'un patient ainsi que les connaissances appropriées à cette situation, correctement filtrées et présentées afin d'améliorer la qualité des soins et la santé des patients [1].

Les SADMs sont des outils permettant d'assister le praticien dans sa démarche de diagnostic en fournissant une meilleure évaluation de l'état du patient. De plus, ils visent à diminuer l'incertitude sur la situation du patient.

Ces dernières années, grâce aux nouvelles technologies d'acquisition de données médicales et avec l'apparition de techniques avancées pour l'exploitation et le traitement de ces données, plusieurs approches et algorithmes automatiques pour l'aide au diagnostic médical sont apparus permettant ainsi de construire des classifieurs dits intelligents basés sur des méthodes statistiques et probabilistes.

L'utilisation de ces classifieurs automatiques permet au praticien d'établir facilement et rapidement le diagnostic de son patient, la réponse obtenue grâce aux systèmes automatiques utilisant les méthodes classiques issues de l'intelligence artificielle dite (mono-label) permet simplement de confirmer ou d'infirmer la présence d'une seule pathologie chez un sujet.

Or en réalité, les patients peuvent *être* atteints de plusieurs maladies simultanément, nous illustrons cela par une situation très souvent rencontrée par les praticiens ; l'association chez un même patient d'hypertension, du diabète, d'insuffisance rénale et d'accidents cardio-vasculaires.

Selon l'équipe Médicale Hypertension Online [2] l'hypertension artérielle et le diabète constituent deux facteurs de risques cardiovasculaires, responsables d'atteintes des artères. Le risque cardiovasculaire est considérablement augmenté du fait de l'association de ces deux pathologies. Justifiant que chacune doit être diagnostiquer et prise en charge de façon vigoureuse [3]. La découverte de l'une incite à systématiquement rechercher l'autre puisque 90% des diabétiques sont des hypertendus et 39% des diabétiques récemment diagnostiquée sont déja hypertendus.

Les techniques classiques disponibles dans l'axe de recherche actuel ont pour but la détection automatique des cas avec une seule pathologie. De ce fait, elles permettent de confirmer la présence ou l'absence de pathologie mais ne permettent pas de détecter plusieurs pathologies à la fois, alors que le cas de polypathologies pose une problématique fréquente qui doit être impérativement abordée et résolue.

En utilisant un système d'aide au diagnostic médical, le clinicien espère donner un aperçu complet sur l'état du patient, détecter une seule maladie en utilisant un système automatique c'est intéressant ! mais diagnostiquer les différentes maladies (comme le diabète, l'hypertension, problème cardiovasculaire, insuffisance rénale) à la fois en utilisant un même système automatique et avec précision c'est le cas idéal ! car un diagnostic précoce de pathologies implique un traitement efficace et avec un minium de complications.

Les questions posées à ce niveau sont : *comment peut-on résoudre ce problème ?* et *quels sont les nouvelles solutions pour cerner cette lacune des méthodes mono-label ?*

Ce travail a pour objectif d'introduire cette problématique en détails, présenter un nouvel axe de recherche qui est la classification multi-labels. Un sujet dont l'intérêt a été étudié dans différents domaines. Nous comptons aussi tester ces approches et apporter notre contribution dans le domaine médical.

Ce projet de fin d'études s'articule en quarte chapitres comme suit :

Chapitre 1 présente les notions fondamentales de la classification multi-labels ainsi que les différentes familles de méthodes de ce domaine.

Chapitre 2 concerne l'état de l'art et les différents travaux abordés dans la littérature.

Chapitre 3 présente en détails les approches proposées dans le but d' améliorer deux méthodes de classification multi-labels de la littérature.

Chapitre 4 concerne l'expérimentation, il présente dans la première partie : les banques de données utilisées et les mesures d'évaluation. Dans la seconde partie expose, les expérimentations réalisées, les résultats obtenus avec leurs interprétations et une étude comparative entre les deux approches proposées .

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

Chapitre 1

La classification multi-labels

1 Contexte

Dans les problèmes standards de la classification mono label, un exemple est caractérisé par un ensemble d'attributs et est associé à un seul label ayant un sens unique, exemple : patient malade ou non !

Malgré l'efficacité de ces algorithmes d'apprentissages supervisés permettant de résoudre plusieurs problèmes, la complexité des problèmes récents leurs constituent un handicap. En effet, un objet est généralement assigné à plusieurs labels ayant des sémantiques différentes. Citons par exemple les applicationss réelles suivantes [4], [5] :

- **Catégorisation du texte :** une page d'un journal ou une page web couvre plusieurs sujets et peut être libellée par plusieurs labels selon les thèmes abordés : sports, annonces, politique, art, actualités, santé ...
- **Médical :** un patient peut avoir plusieurs pathologies simultanément.
- **La génétique :** un génome peut avoir plusieurs fonctions telles que : le métabolisme, la transcription,la synthèse de protéines ...
- **Classification de scènes :** une seule scène a une multitude de labels par exemple : montagne, mer, coucher du soleil, ciel...
- **Classification de chansons :** chaque morceau peut évoquer plusieurs émotions simultanément [6].

Récemment, la question de l'apprentissage à partir de données multi-labels a attiré l'attention de beaucoup de chercheurs, motivés par un nombre croissant de nouvelles applications et par son efficacité. Selon Tsoumakas et al. [7] ce type d'apprentissage a été appliqué dans plusieurs domaines tels que :

- l'annotation sémantique d'images et vidéos .
- La bio-informatique.
- La fouille de données sur le web.
- La recherche d'information ...

2 Notions fondamentales sur la classification multi-labels

Dans la classification mono-label un exemple est associé à un seul label, si le nombre de classes est égal à 2 la classification est appelée alors *"classification binaire"* et si le nombre de classes est supérieur à 2, on parle alors de la *"classification multi-classes"*.

Par contre, si l'exemple est associé à plusieurs classes en même temps la classification est appelée *"classification multi-labels"*.

Il existe 2 types de problèmes d'apprentissage multi-labels selon [8] :

1. **Classification multi-labels** : dont le but est de construire un modèle pour la prédiction de labels et d'identifier une liste de labels pertinents.
2. **Ordonnancement de labels (Ranking)** : celui-ci construit un modèle d'apprentissage qui permet d'identifier une liste de labels classés par ordre de pertinence parmi tous les labels possibles pour l'exemple d'entrée.

Ces deux problèmes surgissent dans de nombreuses applications. Le problème de la **classification multi-labels** est rencontré par exemple, dans la caractérisation d'une image (qui peut à la fois contenir une ville, une montagne, une plage,. . .), d'une musique (qui peut *être* à la fois douce, calmante, enthousiasmante) ou d'un film (qui peut appartenir à plusieurs genres). Le problème **d'ordonnancement** est rencontré, par exemple dans la recommandation d'articles selon les préférences des consommateurs.

2.1 Classification multi-labels vs. Classification multi-classes

Dans le cas de la classification multi-classes, chaque exemple est associé à seulement une classe malgré l'existence de plusieurs classes dans la sortie (comme l'illustre clairement la figure 1.1).

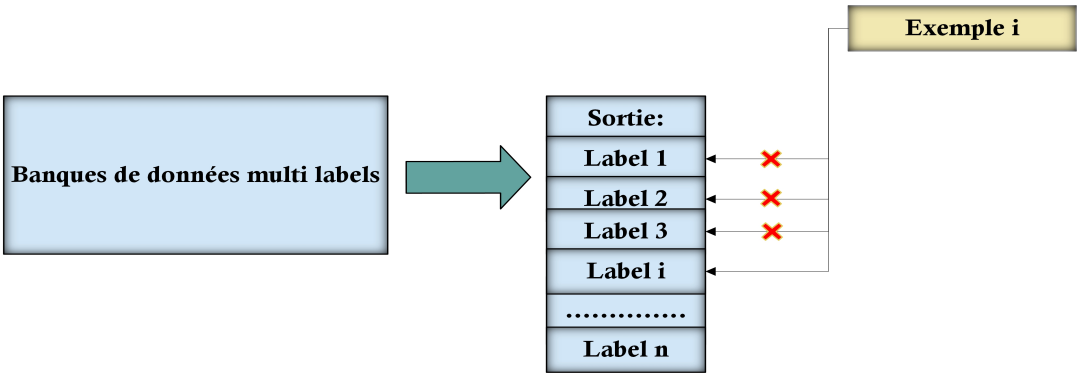


FIGURE 1.1 – La sortie d'un exemple dans le cas de la classification multi-classes

Mais dans la classification multi-labels, un exemple peut être associé à plusieurs labels simultanément (figure 1.2)

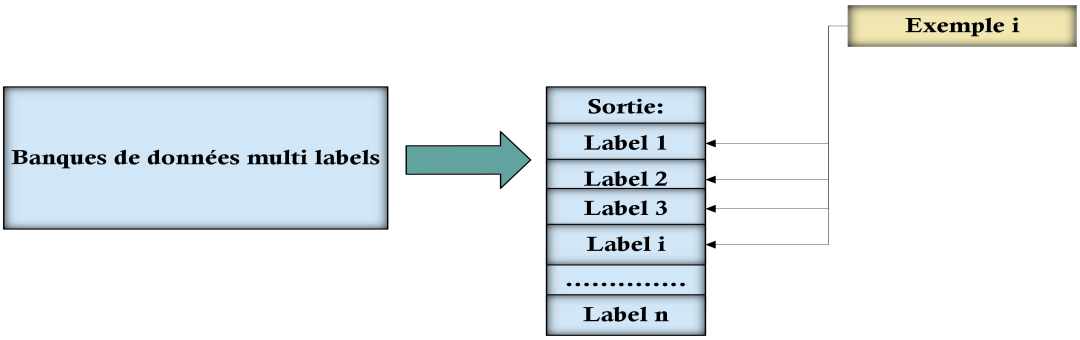


FIGURE 1.2 – La sortie d'un exemple dans le cas de la classification multi-labels

3 Les méthodes de la classification multi-labels

Les méthodes de la classification multi-labels se divisent en trois grandes familles selon Madjarov & DragiKocev [8] :

– *Les Méthodes d'adaptation* : qui se basent sur l'adaptation directe des algorithmes de la classification mono-label pour l'apprentissage multi-labels.

– *Les méthodes de transformation* : elles transforment le problème d'apprentissage multi-labels en plusieurs problèmes de classification ou régression mono-label,

Exemple : utiliser plusieurs classifieurs chacun pour prédire un seul label et combiner les résultats à la fin, dans ce cas on dit qu'on a appliqué une transformation du problème de la classification multi-labels en plusieurs classification binaires. Il existe aussi d'autres types de transformation tels que la classification multi-classes.
Dans cette famille de méthodes "on adapte les données à l'algorithme par une transformation".

– *Les méthodes d'ensembles* : elles utilisent des ensembles de classifieurs issus de la première ou la deuxième famille d'approches.

Dans la partie qui suit nous allons présenter brièvement les méthodes les plus utilisées dans la littérature dans chacune de ces 3 familles.

3.1 Les méthodes d'adaptation

Dans la littérature, certains auteurs ont proposé des méthodes adaptées à la classification multi-labels qui sont basées principalement sur des algorithmes d'apprentissages connus comme : Support Vector Machine(SVM) , K Neaarest Neighbours (KNN), les arbres de décision, réseaux de neurones . . .Nous citerons les approches adaptatives les plus connues dans cette catégorie.

K plus proches voisins : l'adaptation de cet algorithme pour les données multi-labels est appelée MLKNN (multi-labels K Nearest Neighbours), l'algorithme qui a été proposé dans [5, 9]. Cet algorithme utilise le même principe de base de KNN pour la recherche des k plus proches voisins, la différence se constitue dans le fait d'utiliser le calcul de probabilité de l'hypothèse (prior probability) pour déterminer les labels d'un exemple de test

(posterior probability).

MLKNN utilise l'inférence bayésienne qui est une méthode d'inférence permettant de déduire la probabilité d'un événement à partir de celles d'autres événements déjà évalués, elle s'appuie principalement sur le théorème de Bayes.

Arbres de décisions : Clare & king [10] ont adapté l'algorithme C4.5 aux données multi-labels (ML -C4.5) en modifiant la formule de calcul d'entropie.

$$Entropie(D) = -\sum_{j=1}^{q}(p(_j * log_p(_j) + q(_j * log_q(_j)))$$

où : $p(_j)$ =la fréquence relative de la classe $_j$
et $q(_j)$=$1 - p(_j)$

Réseaux de neurones : ont été aussi adaptés pour l'apprentissage des données multi-labels [11,12]. BP-MLL [12] est une adaptation du célèbre algorithme "back-propagation", la modification faite se résume en l'introduction d'une nouvelle formule pour le calcul de l'erreur qui prend en considération le cas multi-labels.

Support à Vaste Marges : Elisseeff and Weston [13] ont proposé une approche basée sur SVM pour l'ordonnancement de labels 'Label Ranking' appelé RankSVM pour les données multi-labels.

3.2 Les méthodes de transformation

Ce type de méthodes transforme le problème d'apprentissage multi-labels en un ou plusieurs problèmes d'apprentissages mono-label, pour l'adapter aux différents algorithmes déjà présents dans la classification supervisée mono-label. Citons les 3 célèbres approches de transformation :

Binary Relevance : Cette méthode [14] transforme le problème d'apprentissage initial en Q problèmes de classification binaires (Q désignant le nombre de classes possibles). Chaque classifieur binaire est utilisé pour séparer une classe des autres. La méthode BR a l'avantage d'être simple et peu *coûteuse* en termes de temps de calcul, mais son problème majeur réside dans le fait qu'elle ne tiennne pas compte des corrélations éventuelles entre les classes comme le montre la figure 1.3 :

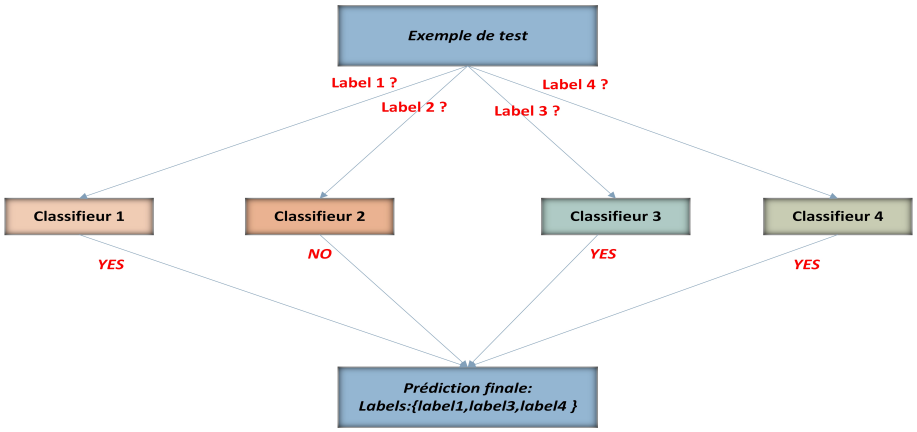


FIGURE 1.3 – Exemple de classification par la méthode Binary Relevance

BR a été amélioré dans [15], l'approche proposée dans cet article se base sur l'utilisation d'une chaine de classifieurs(CC), chaque classifieur permet la prédiction d'un label et prend en considération la dépendance avec les autres labels. Cette méthode donne les mêmes résultats en terme de complexité algorithmique que le BR. Son principal avantage est le fait qu'elle tienne compte des corrélations entre les labels, son inconvénient majeur est la propagation de l'erreur tout au long de la chaine en cas d'erreur.

Label powerset (LP) : appartient à la même catégorie [14], cette méthode a été étudiée récemment par un ensemble de chercheurs [16–18]. Pour cette méthode, chaque ensemble de labels existant dans l'ensemble d'apprentissage de taille N est considéré comme une nouvelle classe. Le problème d'apprentissage multi-labels est ensuite transformé en un problème de classification mono-label où le nombre de classes est au plus égal à min(N,2^Q) avec Q est le nombre de labels avant la transformation . Pour un nouvel exemple à classifier, la classe (multi-labels) la plus probable est prédite. La méthode LP tienne compte des corrélations entre les classes, mais elle souffre en général des problèmes de complexité algorithmique. La figure 1.4 explique ce principe.

exemples/ labels	Label 1	Label 2	Label 3	Label 4
Exemple 1	0	1	1	0
Exemple 2	1	0	0	0
Exemple 3	0	1	1	0
Exemple 4	1	0	0	1

Transformation par LP

exemples/ labels	Nouvelles classes
Exemple 1	0110
Exemple 2	1000
Exemple 3	0110
Exemple 4	1001

FIGURE 1.4 – Exemple de transformation par LP

La méthode *LP* souffre de 3 principaux problèmes :

– Plus le nombre de labels est élevé, la classe à prédire par LP est potentiellement complexe. Pour résoudre cette lacune, Read [19] a proposé la méthode "pruned problem transformation (PPT)" pour sélectionner seulement les labels transformés qui apparaissent souvent dans la phase d'apprentissage par rapport à un seuil défini par l'utilisateur.

– Dans la phase d'apprentissage, on ne trouve pas suffisamment d'exemples ayant la même combinaison de labels surtout pour un grand nombre de labels.

– La prédiction d'un nouvel exemple de test est vraiment très difficile surtout si l'exemple en question a une combinaison de labels totalement différente des labels des exemples de l'apprentissage.

Ranking by Pairwise Comparison : cette méthode [20] est utilisée dans le Ranking, son principe est le suivant :

Soit une banque d'apprentissage contenant : 4 exemples x(1), x(2), x(3), x(4), chaque exemple est associé à 4 labels comme le montre la figure 3.5

Exemples	L1	L2	L3	L4	Y
X(1)	1	0	0	1	{L1,L4}
X(2)	0	0	1	1	{L3,L4}
X(3)	1	0	0	0	{L1}
X(4)	0	1	1	1	{L2,L3,L4}

FIGURE 1.5 – Exemple de banque d'apprentissage

La méthode Pairwise utilise $L(L-1)/2$ classifieurs binaires chacun pour une paire de labels($\lambda i, \lambda j$),$1 \leq i < j \leq L$

Le principe de cette méthode est expliqué dans la figure 1.6, chaque apprenant fait l'apprentissage sur les exemples qui sont annotés par au moins un et un seul label du label-set (mais pas les deux).

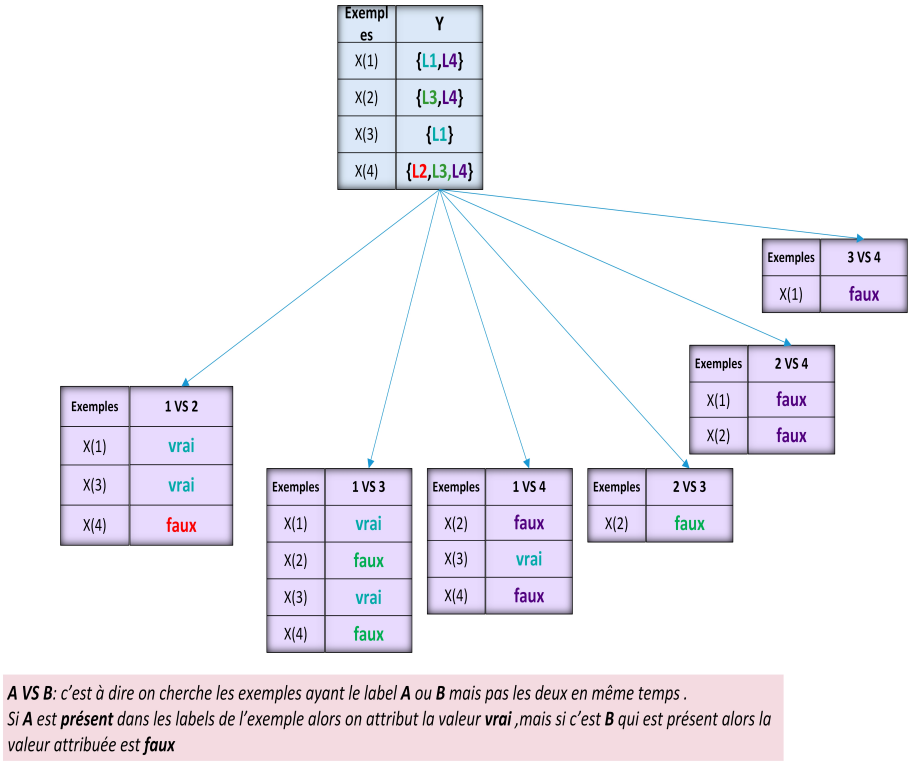


FIGURE 1.6 – Principe du Ranking By Pairwise Comparison

Pour une nouvelle instance, tous les modèles sont évoqués dans la prédiction de labels, et un ordonnancement de labels (Ranking)est obtenu par rapport à un vote donné par chaque apprenant pour chaque label comme montré dans la figure 1.7.

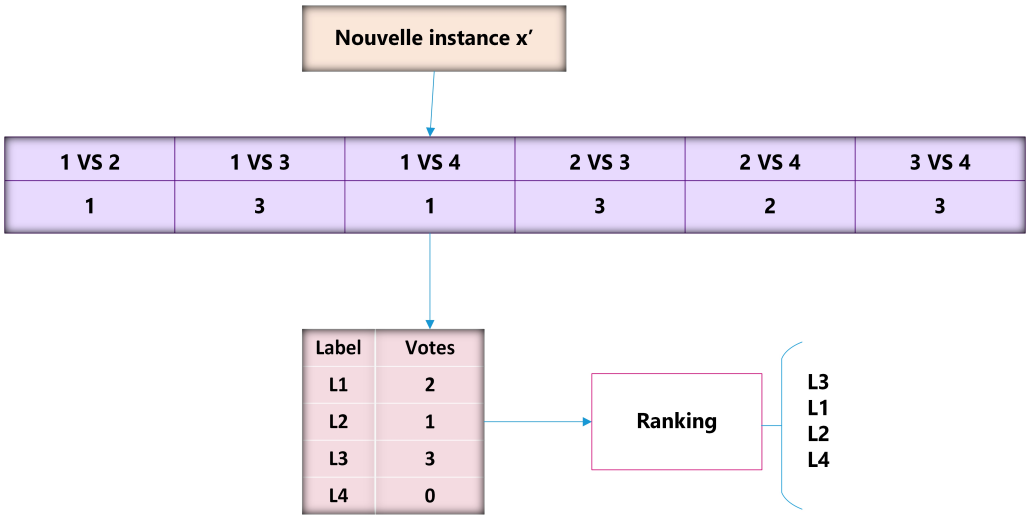


FIGURE 1.7 – Principe du Ranking By Pairwise Comparison(2)

3.3 Les méthodes d'ensembles

Les méthodes d'ensembles de transformations les plus connues sont :

***RAkEL* (RAndom k-labELsets) :** la méthode *RAkEL* [14] est l'une des améliorations de la méthode LP qui apporte des solutions aux limites de LP cités précédemment. Son principe de base est de générer des ensembles de classes aléatoires de tailles petites par rapport au nombre total de classes (Q) et de faire l'apprentissage du classifieur multi-labels pour chaque ensemble aléatoire en utilisant la méthode LP. Les classes sont ensuite déterminées par une stratégie de vote en utilisant un seuil à déterminer. La méthode *RAkEL* a une complexité inférieure à celle de la méthode LP.

ECC (Ensembles of Classifier chains) : cette méthode [15] a pour but de résoudre le problème de propagation de l'erreur dans une chaine de classifieur, qui est un inconvénient majeur de la méthode Classifer chains (CC) (déjà expliqué dans la méthode de transformation Binary relevance) par l'utilisation de 'm' classifieurs chacun fait l'apprentissage seul et génère une chaine aléatoire ; l'apprentissage de chaque classifieur se fait sur un ensemble d'instances choisi aléatoirement avec remise (bagging Brieman [21]) et le vote final sera déterminé par rapport à un seuil fixé par l'utilisateur.

Un exemple de méthodes d'ensembles d'adaptation est : **PCTs (Ensembles of predictive clustering trees)** par Kocev [22] a utilisé un ensemble de predictive clustering tree (PCT) où l'arbre est considéré comme une hiérarchie de clusters, le nœud principal correspond à toutes les données et il est partitionné en sous clusters. Chaque PCT dans l'ensemble fait une prédiction de labels et à la fin les résultats sont combinés en utilisant un vote.

4 Motivations

La classification multi-labels est une extension de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, chaque individu pouvant appartenir à plusieurs classes simultanément. Ce type de classification est requis par un grand nombre d'applications actuelles telles que la classification d'images et l'annotation de vidéos, la catégorisation du texte, l'aide au diagnostic médical ....

L'objectif de la classification multi-labels est de prédire les labels possibles pour une instance donnée, certains chercheurs ont recours à un seul classifieur pour prédire les labels parmi les méthodes de transformation ou d'adaptation comme par exemple MLKnn. Cependant d'autres préfèrent utiliser plusieurs classifieurs à la fois pour prédire les labels d'une instance donnée, en finalité ils font appel à un vote en choisissant un seuil pour assigner les labels les plus adéquats à cette instance. Selon Chuan et al. [23] cette 2ème approche c'est à dire l'utilisation des méthodes d'ensembles est la plus performante car elle a une possibilité de généralisation très grande et son succès est liée à la diversité de la banque d'apprentissage de chaque classifieur et à sa précision de classification.

La question qui se pose est *pourquoi faire ?*, notre objectif est d'appliquer ces algorithmes dans le cadre médical par l'utilisation d'un système d'aide au diagnostic médical (SADM) qui détermine selon certains critères les maladies associées simultanément à un sujet. Cela dépend de la précision de classification et du temps de réponse qui doit être relativement court, surtout dans des cas où le temps est crucial et la vie des personnes est en jeu, citons un simple exemple le cas où le SADM est utilisé dans le service des urgences, le médecin espère connaitre de façon exhaustive l'état de santé de la personne en danger en un temps très réduit pour pouvoir agir efficacement.

5 Conclusion

Dans ce chapitre nous avons exposé les notions fondamentales de la classification multi-labels, et les différentes familles de méthodes : **méthodes de transformation**, **méthodes d'adaptation** et **les méthodes d'ensembles**, nous avons présenté aussi brièvement l'objectif de notre travail. Dans le chapitre 2, nous allons présenter l'état de l'art des méthodes en classification multi-labels.

Chapitre 2

État de l'art des approches de classification multi-labels

1 Introduction

Ces dernières années, différentes extensions du problème de la classification classique ont émergé, parmi lesquelles se trouvent les problèmes de la classification multi-labels et les problèmes d'ordonnancement de labels (*Ranking*).

Pour la classification, une ou plusieurs classes sont associées simultanément à une instance. Dans le cas du problème d'ordonnancement de labels, il s'agit d'apprendre pour chaque instance x un ordre total sur l'ensemble de sortie.

A travers les nombreuses recherches effectuées ces dernières années, nous distinguons plusieurs approches proposées dans la littérature pour résoudre ce problème. Dans ce chapitre, notre travail consiste à présenter un ensemble de travaux réalisés récemment, à en expliquer les principes de base, et aussi pour se situer parmi ces travaux afin de cerner les difficultés potentielles de ce domaine ainsi que les voies de développement à éviter.

Dans la littérature, nous constatons que les approches proposées s'intéressent soit à la **classification** multi-labels, soit à **l'ordonnancement de labels (Ranking)**. Nous distinguons dans la 1 ère famille trois types de méthodes : la méthode *d'adaptation*, la méthode de *transformation* ou bien *un ensemble de méthodes issu de la 1 ère ou de la 2 ème famille*. Sur cette répartition, ce chapitre a été réalisé en ajoutant aussi deux notions très importantes, il s'agit de la ***Corrélation (Dépendance)*** entre les labels et la ***Sélection de variables*** et son impact sur les méthodes de classification multi-labels.

2 Travaux dans la classification multi-labels

Nous présentons dans cette partie un ensemble de travaux sur la classification multi-labels. Tsoumakas & Kataki [16] sont les premiers a avoir fait une overview du domaine. Ils expliquent aux chercheurs les notions de bases de ce nouveau domaine, les différentes approches de bases ainsi que leurs inconvénients et avantages respectifs. De plus ils expliquent la différence entre des notions proches de la classification multi-labels qui sont en réalité différentes telles que :*multiple labels problem*,*multiple instance learning*, *Ranking*, *hierarchical classification*.

Dans le même contexte, nous trouvons d'autres travaux [24, 25] qui traitent en profondeur la problématique de la classification multi-labels. Ils présentent une étude comparative entre les différents algorithmes proposés dans ce domaine ainsi que les mesures d'évaluations et les banques de données les plus utilisées.

Également, Tsoumakas et al. [7], Zhang & Zhou [4] ainsi que Madjarov et al. [8], Malik & Tarique [26] présentent les différentes approches en classification multi-labels. En premier lieu, les auteurs nous présentent les notions et définitions de bases sur la problématique des données multi-labels. En second lieu, ils présentent les travaux réalisés en deux grandes familles : les algorithmes de transformation de problèmes et les algorithmes d'adaptation. Un autre volet qui est abordé concerne les données multi-labels à grandes dimensions et l'implication des méthodes de réduction et d'extraction de données dans le domaine de la classification multi-labels. D'autres aspects statistiques, sur la classification hiérarchique ainsi que le ranking sont aussi abordés.

Dans tous ces travaux, les auteurs confirment la nécessité de l'exploitation des dépendances entre les labels dans ce domaine et concluent que c'est la clé de réussite de la classification multi-labels, et c'est pourquoi nous constatons ces dernières années l'émergence de beaucoup de travaux destinés à résoudre cette problématique.

La classification multi-labels est devenue de plus en plus importante dans le domaine de l'intelligence artificielle vu le nombre d'applications réelles proposées dans la littérature. Dans cette approche, nous distinguons plusieurs familles d'algorithmes permettant d'attribuer un ensemble de labels simultanément à un exemple d'apprentissage. Dans cette partie nous allons citer quelques travaux s'adressant à résoudre cette problématique.

2.1 Méthodes d'adaptation

La première famille de méthode pour résoudre le problème de la classification c'est les méthodes d'adaptation, nous citons quelques-unes dans cette partie :

Dans [5], Zhang et al. présentent une adaptation de L'algorithme K plus proches voisins (KNN) aux données multi-labels appelée $MLknn$(Multi-label K Nearest Neighbours). Cet algorithme utilise le même principe de base de KNN pour la recherche des k plus proches voisins, la différence se constitue dans le fait d'utiliser le calcul de probabilités de l'hypothèse (prior probability) pour déterminer les labels d'un exemple de test (posterior probability).

Deux ans plus tard, Younes et al. proposent une nouvelle méthode pour la classification multi-labels appelé *DMl knn* (Dependent Multi-Label k- Nearest Neighbor) [27], dérivée de la méthode d'ensemble de transformation $MLknn$ qui utilise les réseaux bayésiens.
Cette méthode modifie la version de base de $MLknn$ et prend en considération les dépendances entre les différents labels de la banque de données. Pour chaque instance, l'algorithme identifie les k plus proches voisins dans la banque d'exemples, et prend en considération les classes de ces voisins en utilisant un principe appelé MAP (global maximum a posteriori) pour assigner un ensemble de labels à cette instance. A la différence de l'algorithme de base $MLknn$, cet algorithme prend en considération le nombre de tous les voisins de l'exemple en question et non pas seulement le nombre des voisins ayant la même classe et utilise de ce fait, la matrice de dépendances pour définir les relations entre les différents labels.

Dans leur travail [28], Zhang et al. présentent une nouvelle méthode *MLNB* (multi-labels naives bayes) qui est une adaptation des réseaux bayésiens classiques pour la classification multi-labels. Les mécanismes de sélection de variables sont présentés pour améliorer cette méthode d'adaptation. La première étape consiste à utiliser la méthode d'analyse en composantes principales (ACP) pour éliminer les attributs non significatifs et redondants. La seconde partie consiste à utiliser les techniques de sélection de sous ensemble d'attributs basés sur les algorithmes génétiques pour choisir le sous ensemble d'attributs le plus approprié à la prédiction. Les expérimentations sur des données synthétiques et réels ont montré des performances très intéressantes pour cet algorithme d'adaptation multi-labels.

2.2 Méthodes de transformation

La deuxième famille englobe les méthodes de transformations, nous présentons quelques travaux qui portent sur les trois types de transformations : BR(Binary Relevance), LP (Label powerset) et la méthode *Pairwise*, nous avons déjà présenté dans le chapitre1 le principe de chaque approche et dans cette partie nous présenterons les améliorations proposées dans différents travaux.

L'une des méthodes les plus basiques parmi une panoplie de méthodes de classification multi-labels est *Binary relevance*, elle associe un classifieur binaire pour la prédiction de chaque label mais son principal inconvénient est le fait qu'elle ne tienne pas compte des corrélations entre les labels ce qui est un problème.

Dans un premier temps, Read et al. dans [15] proposent une approche qui a résolu cette lacune pour pouvoir bénéficier des avantages de BR : la facilité de traitement (complexité algorithmique et temps d'exécution). Cet algorithme se base sur l'utilisation d'une chaine de classifieurs, chaque classifieur permet la prédiction d'un label et prend en considération la dépendance avec les autres labels, comme dans le cas de la méthode *Binary Relevance* on construit l classifieurs binaires. Mais afin d'intégrer d'éventuelles dépendances entre les labels, le j ème classifieur utilise dans ses attributs d'entrée les prédictions des $j-1$ classifieurs précédents.

Pareillement, Alvares-Cherman et al. [29] présentent aussi une amélioration de la même méthode par l'utilisation des arbres de décisions et les réseaux bayésien comme classifieurs de base pour prendre en considération les dépendances entre les labels. L'approche proposée dans ce papier a montré son efficacité surtout sur les banques de données non volumineuses.

Un autre type de transformation est le *Label Powerset* [7] qui transforme le problème de la classification multi-labels en un problème de classification multi-classes. Cette approche a été étudiée par différents auteurs afin de tirer profit de ses avantages et de l'améliorer, nous citerons :

Tsoumakas et al. qui présentent dans [14,18] la méthode Random k label-sets ($RAkEL$) une amélioration du LP. Elle consiste à décomposer de manière aléatoire l'ensemble de labels en sous-ensembles et à construire par la suite un classifieur LP pour chacun. Pour chaque label λi une décision moyenne est calculée, la décision finale est positive pour un label donné si la décision moyenne est plus grande qu'une valeur seuil t.

Par la suite, Sawsan et al. proposent dans [30] la méthode *RAkEL évidentielle* qui utilise la méthode *RAkEL* conjointement avec la théorie des fonctions de croyances dans le but de réduire la perte d'information causée par l'utilisation de la méthode *RAkEL*(du

fait que l'apprentissage de classifieurs est effectué sur un sous-ensemble de labels) tout en tenant compte des corrélations possibles entre les différentes classes. L'utilisation de la théorie des fonctions de croyances rend possible l'association d'une fonction de masse à chaque classifieur. Ces fonctions sont ensuite combinées par un opérateur adapté dans le but de donner une décision finale sur l'appartenance d'un individu à un ensemble de labels .

Une autre nouvelle approche de transformation du problème de classification multi-labels appelée *PPT* (pruned problem transformation) a été proposée par Read [19]. Elle se base sur le même principe de *RAkEL* mais au lieu de choisir les sous ensemble de labels (k label-sets) aléatoirement, *PPT* prend en considération les corrélations entre les différents labels, un autre avantage est que l'algorithme utilisé n'est pas complexe donc le traitement est facile du coté temps d'exécution et complexité algorithmique surtout en utilisant le *SVM*.

Hung et al. exposent dans [31] une méthode qui se base sur la transformation *LP* ayant le même principe que *RAkEL* mais qui utilise aussi le processus d'apprentissage de la méthode *Adaboost* [32].

Récemment, Gasse et al. [33] proposent une nouvelle approche hybride pour l'apprentissage structurel des réseaux bayésiens appelée :*H2PC* qui construit au départ la structure du réseau bayésien et ensuite utilise « Bayesian-scoring greedy hill-climbing search » pour orienter le graphe. En seconde partie de leurs travail, ils proposent d'appliquer cette nouvelle approche aux données multi-labels et ils s'intéressent en particulier à la méthode Label powerset qui tienne compte des dépendances entre les labels. L'utilisation de *H2PC* permet de tracer graphiquement grâce à la structure du réseau bayésien produite les dépendances entre les labels ce qui est très important pour plusieurs applications.

On distingue également d'autres travaux visant à améliorer un autre type de transformation appelé : *Pairwise*.

Les faiblesses de *BR* ont été résolués en utilisant l'apprentissage par *Pairwise Preferences* qui utilise pour chaque paire de labels un classifieur pour prédire un des deux labels (voir détails chapitre 1). Mais cette méthode souffre d'une part du nombre quadratique de classifieurs nécessaires pour chaque paire de labels surtout si le nombre de labels dans la banque est élevé, d'autre part de sa complexité algorithmique.

Dans [34], une amélioration de la méthode *pairwise comparison* est présentée par la combinaison entre la technique qui transforme le probleme de 'Ranking' en un problème de bipartition pour séparer entre les labels pertinents et non pertinents, par l'introduction d'un label de calibration [35] avec l'approche "QWeighted voting". Cette dernière est destiné en réalité à la classification multi-classes et qui a été adapté au problème multi-labels. Son rôle est d'arrêter le calcul du ranking (l'ordonnancement de labels) lorsque la bonne séparation entre les 2 parties (labels pertients et non pertinents) est obtenue [36].

Mêmement, Madjarov et al. présentent une méthode de classification multi-labels éfficace appelé *TSA* (Two Stage Architecture) [37] avec 3 analyses d'implémentation différentes : Two Stage Voting Method (*TSVM*), Two Stage Classifier Chain Method (*TSCCM*), Two Stage Pruned Classifier Chain Method (*TSPCCM*) pour résoudre les inconvénients de la méthode *Pairwise*. Les résultats montrent que les méthodes proposées dans ce papier surpassent plusieurs algorithmes de la littérature en terme d'exactitude et en terme de rapi-

dité de calculs, même par rapport aux méthodes Pairwise pour l'apprentissage multi-labels.

La majorité des algorithmes proposés dans l'état de l'art souffrent du problème de temps d'exécution qui est considérable et de la nécessité d'un grand espace de mémoire afin qu'ils fonctionnent efficacement.

Dans ce contexte, Tsoumakas et al. proposent une nouvelle approche dans [38] pour la classification multi-labels appelé *HOMER* (Hierarchy Of Multi-label classiffiERs). Cette approche a la particularité de classer les classifeurs multi-labels en une hiérarchie afin d'optimiser le temps d'exécution et la consommation en espace mémoire, puisque ces 2 paramètres deviennent très importants pour les banques avec un grand nombre de labels. Delà, chaque classifieur apprend sur un ensemble réduit de labels avec une distribution équilibrée d'exemples, ce qui permet d'avoir une amélioration dans les performances de prédictions des labels, la distribution de labels dans la hiérarchie du nœud père vers le nœud enfant est faite grâce à un nouveau algorithme qui a été également proposé dans ce travail appelé : *balanced kmeans* qui utilise le principe de *balanced clustering* [39].

2.3 Méthodes d'ensembles

Les méthodes d'ensembles utilisent un ensemble de classifieurs issu de la famille de méthodes de transformation ou bien d'adaptation.

Chuan et al. dans [23] développent dans leur travail une méthode appelée *EnML* (Multi-Label Ensemble Learning). Le but de cette méthode est de construire un ensemble de classifieurs multi-labels qui apprend à partir de banques d'apprentissages diverses pour augmenter la précision de classification. *EnML* permet de résoudre le problème de diversité et augmente le taux de précision en se basant sur les algorithmes évolutionnaires pour optimiser chaque classifieur, dans l'objectif d'avoir les labels les plus optimaux à une instance donnée.

Kocev et al. traitent dans [40] et [22] la problématique de la prédiction de données structurées en utilisant deux types d'ensembles pour la prédiction de différentes sorties structurées de manière locale et globale.
La méthode globale consiste à utiliser un seul modèle pour la prédiction de données structurées et la méthode locale consiste à utiliser une collection de modèles. Chacune fait la prédiction d'une partie de données structurées et la sortie finale c'est la combinaison de leurs réponses. Les résultats montrent que l'ensemble d'arbres locaux et globaux donne des résultats meilleurs que l'utilisation d'un seul modèle.

Dans la partie qui reste de ce travail, nous répondons à deux questions importantes : *Quel est l'impact de Corrélations entre les labels dans la classification multi-labels ?* et *Quels seront les performances d'un classifieur dans le cas d'une sélection de variables dans ce domaine ?*

Corrélations de labels dans la classification multi-labels

Ces dernières années, la problématique de prendre en considération les corrélations entre les labels dans la classification multi-labels pour améliorer les performances des méthodes a été étudiée par beaucoup de chercheurs dans plusieurs travaux. Les algorithmes et méthodes proposés ont pour but de définir des outils pour exploiter les dépendances entre les labels qui est un point très important pour une classification multi-labels plus correcte.

Parmi lesquels nous citons quelques-uns.

Dans [41], Montanes et al. présentent une approche pour résoudre ce problème, l'efficacité de cette approche se résume en 2 points essentiels : la capacité de l'utiliser et dans la classification et dans le ranking, l'agrégation entre 2 modèles : l'un prend en considération les corrélations entre les labels et l'autre non, ce qui rend le modèle très performant.

Dans la même idée, Tahir et al. [42] proposent l'utilisation des méthodes d'ensembles hétérogènes (c'est-à-dire utiliser plusieurs classifieurs de différents types d'algorithmes) pour solutionner le problème de corrélation de labels ainsi que les exemples d'apprentissages déséquilibrés. Car on peut trouver un label avec plusieurs exemples d'apprentissages alors que d'autres labels ont très peu d'exemples et ça pose un problème lors de l'apprentissage). Les auteurs combinent l'ensemble de méthodes de la classification proposées dans la littérature telles que (*RAkEL*, *MLknn*, calibrated label ranking (*CLR*),ensemble classifier chains (*ECC*), Instance Based Logistic Regression (*IBLR*)) pour former ce qu'on appelle les méthodes d'ensembles hétérogènes, la méthode résultante est appelé : *EML*.

Dans la majorité de méthodes proposées dans l'état de l'art du domaine, l'exploitation de labels se fait globalement c'est-à-dire elles considèrent pour toutes les instances d'apprentissage la même relation de corrélations entre les labels. Par contre, ce n'est pas toujours vrai car dans les applications réelles chaque instance est caractérisée par une relation de dépendance différente. Par exemple dans le cadre médical chaque patient est caractérisé par des labels reliés par une relation bien définie, donc on ne peut pas considérer pour tous les patients une même relation de dépendances.

Huang et al. traitent cette problématique [43] et proposent un algorithme appelé *ML-LOC* (Multi-Label learning using Local Correlation) qui exploite les dépendances entre les labels localement en utilisant une méthode qui regroupe les instances en plusieurs groupes. Chaque groupe contient les instances qui partagent le même sous ensemble de labels corrélés et qui sera considéré comme un nouvel attribut lors de l'apprentissage. Les premiers résultats sur des données multi-labels sont très compétitifs par rapport aux algorithmes de l'état de l'art.

Dans [44], Dembczynski et al. élaborent une étude sur ce problème, contribuant ainsi à une meilleure compréhension de ce domaine et mettent l'accent sur la présence de deux types de dépendances qui sont les dépendances : Marginales et conditionnelles. Ils présentent trois scénarios dans lesquels l'exploitation de l'un de ces types peuvent améliorer les performances des classifieurs.

Sélection de variables dans la classification multi-labels

Dans le processus du test pour déterminer les labels d'une instance, les approches de classification multi-labels proposées dans la littérature se basent sur un même ensemble de variables pour tous les labels, mais cette stratégie peut échouer dans le cas d'un grand nombre de labels qui nécessitent chacun un ensemble de variables spécifiques.

Sur la base de cette réflexion, Zhang propose dans [45], une approche s'appuyant sur les caractéristiques spécifiques du label, l'algorithme proposé est appelé *Lift*.
Le principe de cet algorithme est de construire un ensemble spécifique d'attributs pour chaque label en utilisant l'analyse par méthode de regroupement (clusters) pour ces exemples négatifs et positifs, ainsi les variables spécifiques à chaque classe sont déterminées. Et puis

appliquer le processus d'apprentissage et de test sur le résultat obtenu par cette analyse. Cette approche a été testée sur 16 banques de données multi-labels et les résultats sont meilleurs par rapport aux plus performantes méthodes de la classification multi-labels de la littérature.

L'identification des sous-ensembles de variables aléatoires parmi des centaines de variables non pertinentes ou redondantes est un thème très important dans le domaine de la reconnaissance de formes. Ce sujet a attiré l'attention de beaucoup de chercheurs récemment et a été étudié dans [40, 46, 47]. Car la classification en se basant sur un ensemble de variables limité et contenant en plus des variables non pertinentes et redondantes rend cette tâche non performante [48] et dans le domaine de classification mono-label et dans la classification multi-labels.

Dans ce même contexte, Gharroudi et al. présentent dans [49] trois méthodes de sélection de variables par approche enveloppe dans le cadre des exemples multi-labels, qui sont basés sur les forêts aléatoires *BRRF*(Binary Relevance Random Forest), *LPRF* (Label Powerset Random Forest), et *PCTRF* (Predictive Clustering Tree Random Forest). Ces trois approches diffèrent dans la façon de prendre en considération les dépendances entre les labels dans le processus de sélection de variables. Les résultats montrent que les forêts aléatoires donnent de bons résultats dans le cadre de sélection de variables pour les données multi-labels.

3 Contributions

En s'inspirant des méthodes présentées dans l'état de l'art, nous proposons dans ce projet de fin d'études de Master, d'établir une étude comparative entre deux méthodes *RAkEL* et *MLknn* appartenant respectivement aux deux familles de transformation et d'adaptation. Elles sont connues par leurs capacités de classification très élevées en un temps très réduit, deux critères qui nous intéressent le plus dans le cadre médical.

Le choix de ces deux algorithmes de classification parmi plusieurs méthodes dans chaque famille a été fait en prenant en considération : le but de notre travail qui est la classification multi-labels, la complexité algorithmique, ainsi que leurs capacités de classification reconnues dans la littérature. Les méthodes *RAkEL* et *MLknn* répondent à tous ces critères puisqu'elles ont des principes simples avec des capacités de classification très grandes.

En effet, l'objectif de la classification multi-labels est de prédire les labels possibles pour une instance donnée. Il y a certains chercheurs qui préfèrent utiliser un seul classifieur pour prédire les labels parmi les méthodes de transformation ou d'adaptation comme par exemple *MLknn*, et d'autres préfèrent utiliser plusieurs classifieurs à la fois pour prédire les labels d'une instance donnée. En finalité, ils font appel à un vote en choisissant un seuil pour assigner les labels les plus adéquats à cette instance.

Selon Chuan Shi et al. [23], cette 2ème approche c'est à dire l'utilisation des méthodes d'ensembles est la plus performante car elle a une possibilité de généralisation très grande et son succès est lié à la diversité de la banque d'apprentissage de chaque classifieur et sa précision de classification. En plus, 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. Cette raison nous a orienté vers la proposition d'améliorer la méthode *MLknn* par

l'application des méthodes d'ensembles qu'on nommera *Bagged MLknn*.

Pour la méthode *RAkEL*, nous avons choisi cet algorithme parmi plusieurs algorithmes d'ensembles de transformation pour plusieurs raisons. Premièrement, car il présente plusieurs avantages et son principe est très intéressant. C'est une amélioration d'un type de transformation appelé *Label powerset* qui transforme le problème d'apprentissage multi-labels en un problème de classification multi-classes. Deuxièmement, *RAkEL* appartient aux méthodes d'ensembles qui présentent beaucoup d'avantages [16] (voir détails chapitre 3). De plus, selon Dietterich [50], plus les classifieurs dans les méthodes d'ensembles sont divers, plus leurs performances sont meilleures. La méthode *RAkEL* répond à cette condition et assure la diversité des classifieurs par le choix aléatoire de sous ensemble de labels(labelsets).

Finalement, dans [14], les auteurs ont testé l'algorithme *RAkEL* en utilisant comme prédicteurs les algorithmes suivants : *SVM*, *C4.5*, *Naive Bayes*, nous proposons dans ce travail de tester les Forêts aléatoires (un cas particulier du *Bagging*) qui a montré son efficacité et qui présente beaucoup d'avantages (voir chapitre 4).

Le tableau suivant résume les contributions qui vont être expliquées en détails dans la suite de ce travail :

Méthodes	Algorithmes	**Contributions**
Transformation	*RAkEL* [14]	• *RAkEL* avec Forêts aléatoires. • Stratégie de vote : vote pondéré.
Adaptation	*MLknn* [5]	• Création de *Bagged MLknn*. • Stratégie de vote : vote pondéré.

TABLE 2.1 – Tableau explicatif des contributions de ce travail

4 Conclusion

Dans ce chapitre, nous avons présenté quelques travaux sur la classification multi-labels qui est devenue un sujet d'actualité, et qui attire l'attention de beaucoup de chercheurs. Nous avons également mis l'accent sur l'importance de deux notions qui sont : La corrélation entre les labels et la sélection de variables, deux critères importants pour évaluer le succès des méthodes de classification multi-labels.

La littérature du domaine de classification multi-labels, nous a permis de faire des constatations que nous voudrions prendre en considération pour d'éventuelles améliorations des méthodes proposées dans ce travail. Il s'agit de la méthode d'ensemble d'adaptation *Bagged MLknn* ainsi que l'algorithme d'ensembles de transformation *RAkEL* basé sur les Forêts aléatoires. L'aspect théorique de nos approches sera présenté dans le chapitre suivant.

Chapitre 3

Principe des approches multi-labels proposées

1 Introduction

Actuellement de nombreux chercheurs s'interessent au domaine de la classifcation multi-labels. Ces études sont réparties entre deux familles de méthodes : les méthodes de **transformation**, les méthodes **d'adaptation** (voir chapitre 1).

Dans ce chapitre, notre objectif est de réaliser une comparaison entre deux méthodes appartenant aux deux familles à savoir : *RAkEL* (Random k label-sets) qui applique une transformation du problème multi-labels en un problème de classification multi-classes. La seconde méthode qui s'appelle *MLknn* (Multi-labels K Nearest Neighbors) est une adaptation de l'algorithme classique *KNN* aux données multi-labels. Ces deux méthodes de classification multi-labels ont des principes simples mais très efficaces. Elles ont prouvé leurs performances dans de multiples applications multi-labels.

2 Propositions

Les méthodes *RAkEL* et *MLknn* présentent beaucoup d'avantages. Dans cette partie de notre travail nous présentons en détail, leurs principes, leurs avantages et les raisons pour lesquelles nous les avons choisies parmi plusieurs appartenant aux deux grandes familles d'adaptation et de transformation.
Dans ce projet de fin d'études, nous proposons des modifications pour améliorer leurs performances par :

1. l'application du *Bagged MLknn* pour améliorer les performances de l'algorithme d'adaptation *MLknn* individuel.
2. l'utilisation des *Forêts* aléatoires comme méthode d'apprentissage pour la méthode $RAkEL_o$.
3. l'amélioration des résultats de chaque méthode par remplacement du vote ordinaire classique "vote majoritaire" par le vote pondéré. Ce choix est justifié par le fait que le vote classique dépend du choix d'une majorité des classifieurs qui donnent une même classe pour une donnée, mais sans prendre en considération les performances de chaque classifieur alors qu' en réalité ils différent. Par contre le vote pondéré sert à valider et pondérer ces classifieurs en prenant en considérations la capacité de

classification de chacun individuellement dans l'opération du vote et cela nous donne des résultats plus performants.

3 Ensemble de classifieurs $k$ppv en classification multi-labels (*Bagged MLknn*)

L'une des contributions proposées dans ce projet de fin d'études est la création d'un ensemble de classifieurs *Bagged MLknn*. Avant d'expliquer le principe de cette approche proposée il est important de comprendre le principe des méthodes d'ensembles qui nous permettent de réaliser cette tâche.

3.1 Les méthodes d'ensembles

Les méthodes d'ensembles sont des méthodes permettant d'utiliser une collection de prédicteurs et agréger l'ensemble de leurs prédictions.

Dans la classification, l'agrégation des classes fournies par les classifieurs peut se faire à l'aide d'un vote tel que : le vote majoritaire. La Figure 3.1 expose un schéma représentatif des méthodes d'ensembles.

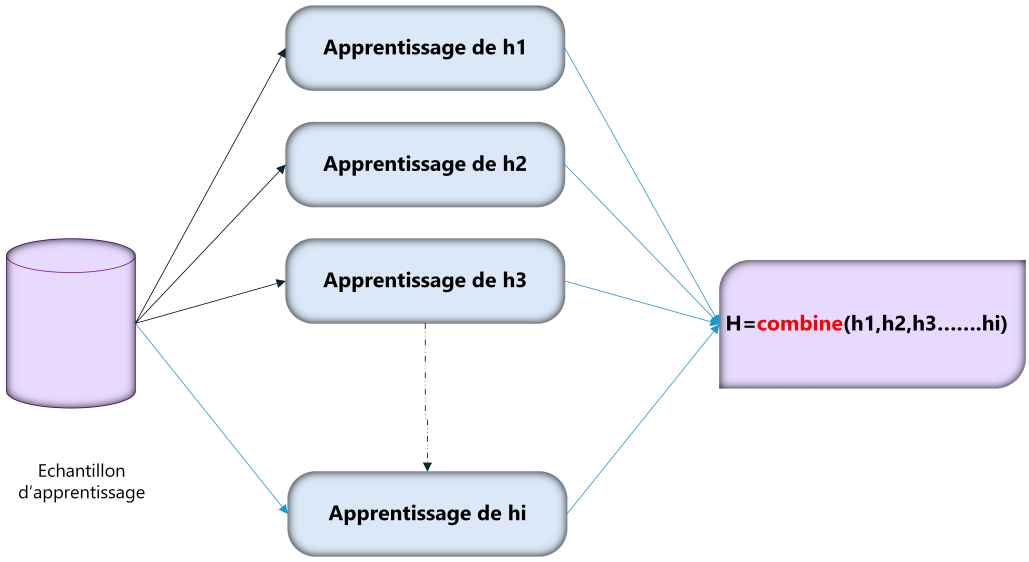


FIGURE 3.1 – Schéma représentatif des méthodes d'ensembles

L'objectif de ces méthodes est de trouver un classifieur final qui soit meilleur que chacun des classifieurs individuels [51]. La méthode d'ensemble est dite performante si la collection de prédicteurs construite vérifie les deux conditions suivantes :

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

L'utilisation des méthodes d'ensembles garantit :

– **La précision** : les prédictions obtenues sont très précises puisqu' on combine la réponse de plusieurs prédicteurs (même faiblement efficaces).

– **L'efficacité** : un problème complexe peut être décomposé en de multiples sous problèmes plus simples à résoudre (approche diviser pour Régner) [52].

3.2 Les types de méthodes d'ensembles

Il existe plusieurs types de méthodes d'ensembles, nous distinguons deux types les plus émergents :

– **Les méthodes d'ensembles hétérogènes** : La réponse finale est obtenue en combinant les résultats des différents algorithmes sur un même ensemble d'apprentissage.

– **Les méthodes d'ensembles homogènes** : Dans ce cas la réponse finale est obtenue en combinant les résultats d'un même algorithme sur un même ensemble d'apprentissage. Ces méthodes utilisent des stratégies adaptatives (*Boosting*) ou aléatoires (*Bagging*).

3.3 *Bagging*

Le Bagging est une méthode d'ensemble introduite par Breiman (1996) [21]. Le mot *Bagging* est la contraction des mots **B**ootstrap et **Agg**regat**ing**.

Le *Bagging* repose sur la méthode *Boostrap* pour améliorer la prédiction d'un classifieur, les observations vont être tirées uniformément et avec remise pour créer de nouveaux échantillons (*Bootstraps*) sur chacun desquels un classifieur est construit. Le classifieur final est obtenu par un vote à la majorité parmi les classifieurs intermédiaires. L'idée majeure est donc d'obtenir divers classifieurs, basés sur de nombreux échantillons. L'un des avantages du Bagging est la réduction de la variance quand les prédicteurs sont «instables».

Les étapes du Bagging selon Leo Breiman [21] sont présentées ci-dessous :

Algorithm 1 Les étapes du *Bagging*

Entrée : Ensemble d'apprentissage $'Sm'$

Générer k échantillons « indépendants »par tirage aléatoire avec remise dans l'échantillon $'Sm'$

pour Chaque échantillon **faire**

Apprendre un classifieur en utilisant le même algorithme d'apprentissage

fin pour

Obtenir la prédiction finale pour un nouveau exemple par vote (simple) des classifieurs

$$\mathrm{H(x)}=\mathrm{sign}(\sum_{i=1}^{T} h_i(x))$$

Sortie : Prédiction finale

La figure3.2 ci-dessous illustre le processus du *Bagging*

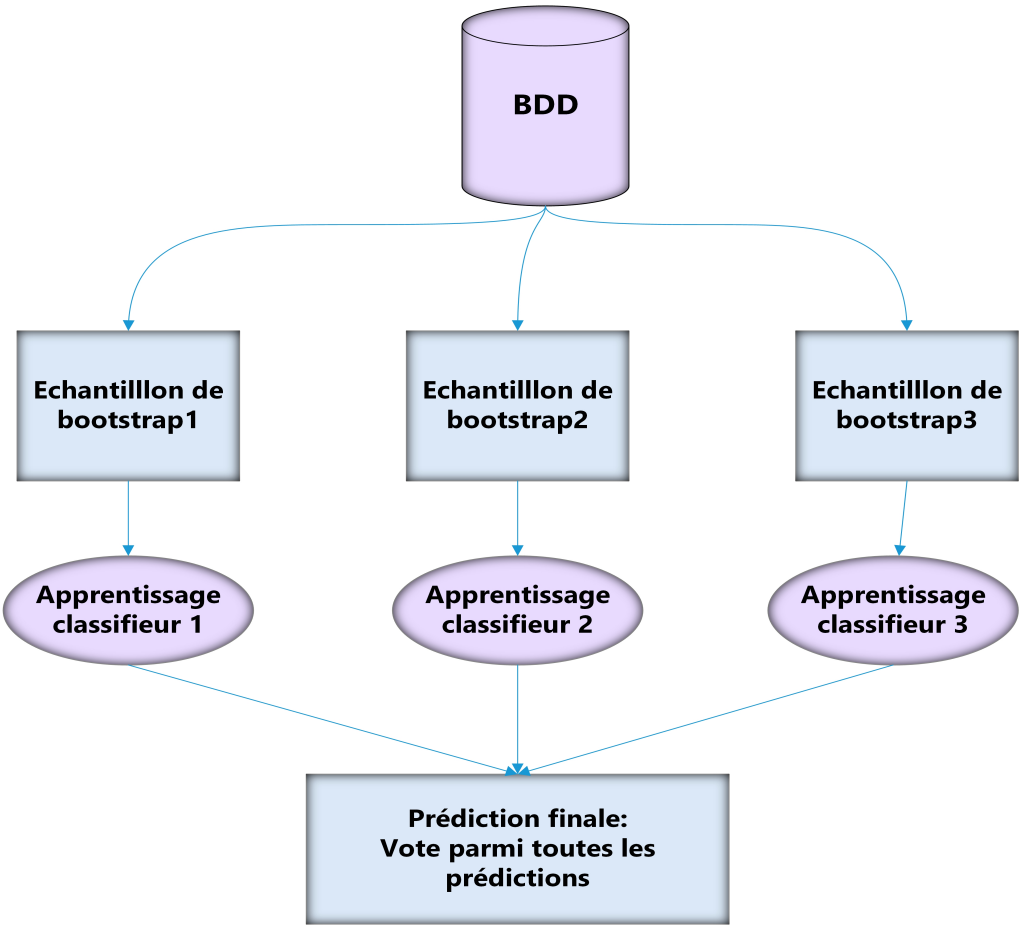


FIGURE 3.2 – Processus du Bagging

Bootstrapping

Selon Efron & Tibshiran [53], le *Bootstrap* est un principe de ré-échantillonnage statistique traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. L'idée du *Bootstrap* est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble de données observé et ce à l'aide d'un tirage aléatoire avec remise.

A partir de cette définition, on peut tirer les conclusions suivantes :

– L'opération du tirage aléatoire avec remise d'un ensemble d'exemples permet d'obtenir un Bootstrap.
– Le bootstrapping d'un ensemble d'apprentissage L produit un nouveau ensemble L_i qui présente en moyenne $1 - e^1 \approx 63\%$ instances uniques différentes de L [54], donc ça permet de créer plusieurs bag à partir d'un ensemble d'apprentissage.

La figure 3.3 représente des échantillons de bootstrap d'une banque de données :

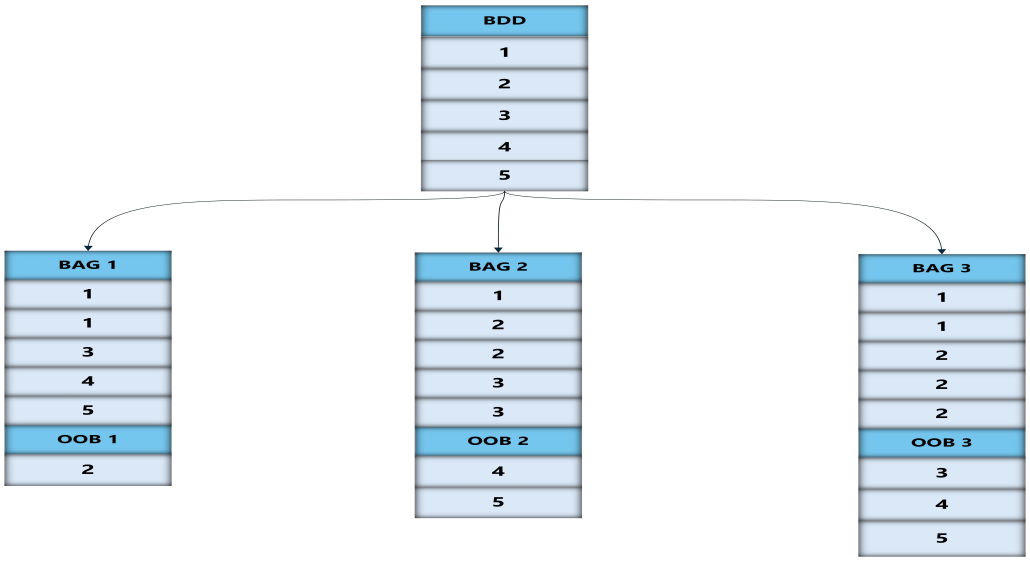


FIGURE 3.3 – Exemple de génération des échantillons bootstrap d'une banque de données

Chaque classifieur élémentaire $h(x)$ de l'ensemble sera entrainé sur un des $'L'$ échantillons bootstrap, de sorte qu'ils soient tous entrainés sur un ensemble d'apprentissage différent.

La figure 3.4 illustre le procédé du Bagging appliqué à un ensemble d'arbres de décision.

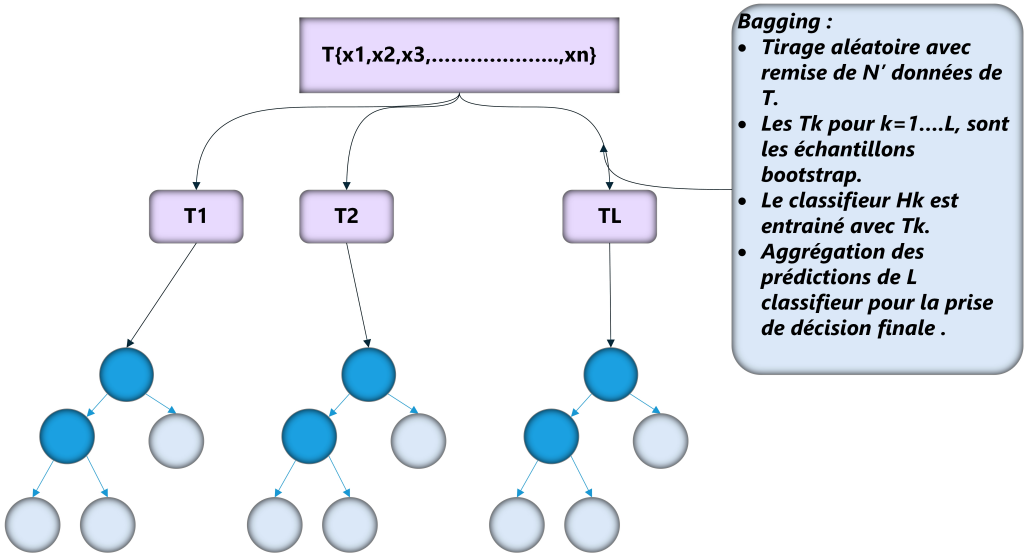


FIGURE 3.4 – schéma explicatif du processus du Bagging

– **Les Mesures Out-Of-Bag** : "En dehors du bootstrap" est l'ensemble des exemples qui ne sont pas sélectionnés dans les échantillons bootstrap. Ce paramètre introduit par la méthode bootstrap permet l'évaluation interne du classifieur qui peut être utilisé par exemple comme pondération dans le vote pondéré .

Les *"Outs Of Bag"* (OOBs) sont des mesures très intéressantes pour faire l'évaluation des données sans avoir recours à un ensemble de données de validation [55].

– **Agrégation** : c'est la combinaison des résultats de chaque hypothèse en utilisant un vote.

Le Bagging peut se faire sur n'importe quel algorithme à savoir : *SVM*, *KNN* ....

3.4 Les k plus proches voisins en classification multi-labels *MLknn*

MLknn (multi-labels K Nearest Neighbors) est une méthode d'adaptation [5], c'est une adaptation de l'algorithme *KNN* aux données multi-labels.

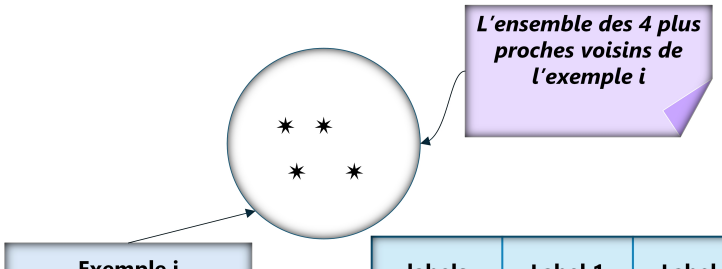
Supposons que nous voulons classer un exemple x, l'approche classique *Knn* calcule d'abord les différentes distances qui le sépare des autres exemples en utilisant dans le cadre général "la distance Euclidienne".

Si nous fixons le nombre de k=10, *Knn* ne va garder que les exemples ayant les 10 distances les plus proches qui sont considérées comme ces 10 plus proches voisins.

Dans le cadre multi-labels, le *MLknn* rajoute à cette notion le fait que le label de l'exemple x est un élément appartenant à l'ensemble de labels de ces K plus proches voisins. Dans l'ensemble des voisins, l'algorithme calcule pour cet exemple la probabilité d'avoir le label $'l'$, et en utilisant un seuil l'algorithme peut décider si ce label est pertinent ou pas. Ce processus est répété pour chaque label de cet ensemble.

MLknn utilise l'inférence bayésienne qui est une méthode d'inférence permettant de déduire la probabilité d'un événement à partir de celles d'autres événements déjà évalués. Elle s'appuie principalement sur le théorème de Bayes, et donc l'algorithme calcule la probabilité de l'hypothèse (prior probability) pour déterminer les labels d'un exemple de test (posterior probability).

Selon Zhang & Zhou [5], prior et posterior probabilités peuvent être calculées directement à partir de la banque d'apprentissage en se basant sur la fréquence. C'est à dire, dans l'ensemble des voisins de l'exemple on compte pour chaque label le nombre d'instances ayant ce label (prior probability) et ceci nous permet de calculer par la suite posterior probability (c'est-à-dire attribuer à l'exemple de test $'x'$ le label $'l'$ si la fréquence de label de ces plus proches voisins est élevé). La Figure 3.5, illustre le schéma de ce processus.



labels	Label 1	Label 2	Label 3	Label 4
Probabilité Seuil=0.5	0.6	0.4	0.9	0.8
Labels finaux pour l'exemple i	1	0	1	1

FIGURE 3.5 – Exemple de classification en utilisant *MLknn*

4 La méthode *RAkEL* (Random k label-sets)

La méthode *RAkEL* est une méthode d'ensemble de transformation pour la classification multi-labels [14, 56]. Elle a l'avantage de transformer un problème de classification multi-labels en un problème de classification multi-classes pour faciliter la prédiction des labels.
RAkEL considère k sous ensemble de labels L appelé (k labels-sets) pour l'apprentissage, c'est à dire au lieu de faire l'apprentissage d'un seul classifieur sur tout l'ensemble de données et faire la prédiction de tous les labels simultanément en se basant sur une transformation de type *classification multi-classes*. La méthode *RAkEL* utilise un principe plus simple qui se résume par les étapes suivantes :

– générer des ensembles de classes aléatoires (k label-sets) de tailles petites par rapport au nombre total de classes.
– appliquer l'apprentissage d'un *LP* (voir chapitre 1) sur chaque label-sets.
– prédiction finale : réaliser par un vote majoritaire de toutes les prédictions de chaque *LP*.

La méthode *RAkEL* a certains avantages [14] :

– elle prend en considération la corrélation entre les labels, puisqu'elle utilise le principe *LP* contrairement à la méthode *BR* (voir chapitre 1).
– elle a une complexité inférieure à celle de la méthode *LP* seule, puisqu'elle fait l'apprentissage de plusieurs classifieurs chacun sur un ensemble de labels (label-sets) au lieu de prendre en considération tous les labels en mêmes temps.

Dans le papier [14], deux stratégies de construction des label-sets ont été proposées :

– *RAkEL* disjoint ($RAkeL_d$), où il n'y a pas de chevauchement entre les k label-sets (c'est-à-dire chaque label-sets permet de prédire un ensemble de labels différent des autres k label-sets).
– *RAkEL* overlapping ($RAkeL_o$) : dans ce cas les k label-sets se chevauchent, c'est-à-dire on peut trouver un label répété dans 2 label-sets différents.

Tsoumakas et al. dans ce papier jugent que cette dernière stratégie est la meilleur car on peut faire un vote entre les différentes prédictions de chaque classifieur et donc les résultats seront meilleurs. Et c'est pour cette raison nous avons décidé d'étudier la méthode *RAkEL* et plus particulièrement *RAkEL* overlapping.

4.1 L'algorithme $RAkEL_o$

Les algorithmes suivants expliquent le processus d'apprentissage (Algorithme 2) et de classification (Algorithme 3) de $RAkEL_o$.

Algorithm 2 Processus d'apprentissage de *RAkEL$_o$*

Entrées : Ensemble de labels L de taille M, Ensemble d'apprentissage D,Taille de label-set k,Nombre de modèles $m \leq \binom{M}{k}$, S Ensemble de labels
$S \leftarrow L^k$
pour $i = 1$ **to** $\min(m, \mid L^k \mid)$ **faire**
$R_i \leftarrow$ sélectionner k label-sets aléatoirement de S ;
Apprentissage des classsifieurs 'LP' h_i à partir de D etR_i ;
fin pour
Sortie : Ensemble de label-sets $R_i$avec les classifieurs h_i

Algorithm 3 Processus de test de *RAkEL$_o$*

Entrées : Ensemble de labels L de taille M, k label-sets R_i construit dans la partie apprentissage, les classifieurs h_i, Nombre de modèles m, nouvelle instance x.
pour $j = 1$ **to** M **faire**
$Sum_j \leftarrow 0$;
$Votes_j \leftarrow 0$;
pour $i = 1$ **to** m **faire**
pour tout Labels $\lambda_j \in R_i$ **faire**
$Sum_j \leftarrow Sum_j + h_i(x, \lambda j)$;
$Votes_j \leftarrow Votes_j + 1$;
fin pour
pour $j = 1$ **to** M **faire**
$Average_j \leftarrow Sum_j / Votes_j$;
si Average >0.5 **alors**
$Result_j \leftarrow 1$
sinon
$Result_j \leftarrow 0$
finsi
fin pour
fin pour
fin pour
Sortie : Vecteur résultats *Result*

Dans cette méthode, trois paramètres sont à identifier :

- le nombre d'ensembles aléatoires (i.e. nombre de classifieurs).
- la taille de ces ensembles.
- le seuil utilisé lors de l'application de la stratégie de vote.

La Figure 3.6 illustre un schéma du processus de la méthode *RAkEL$_o$* d'un exemple à K=3 (nombre de labels considéré=nombre de k label-sets=nombre de classifieurs=3).

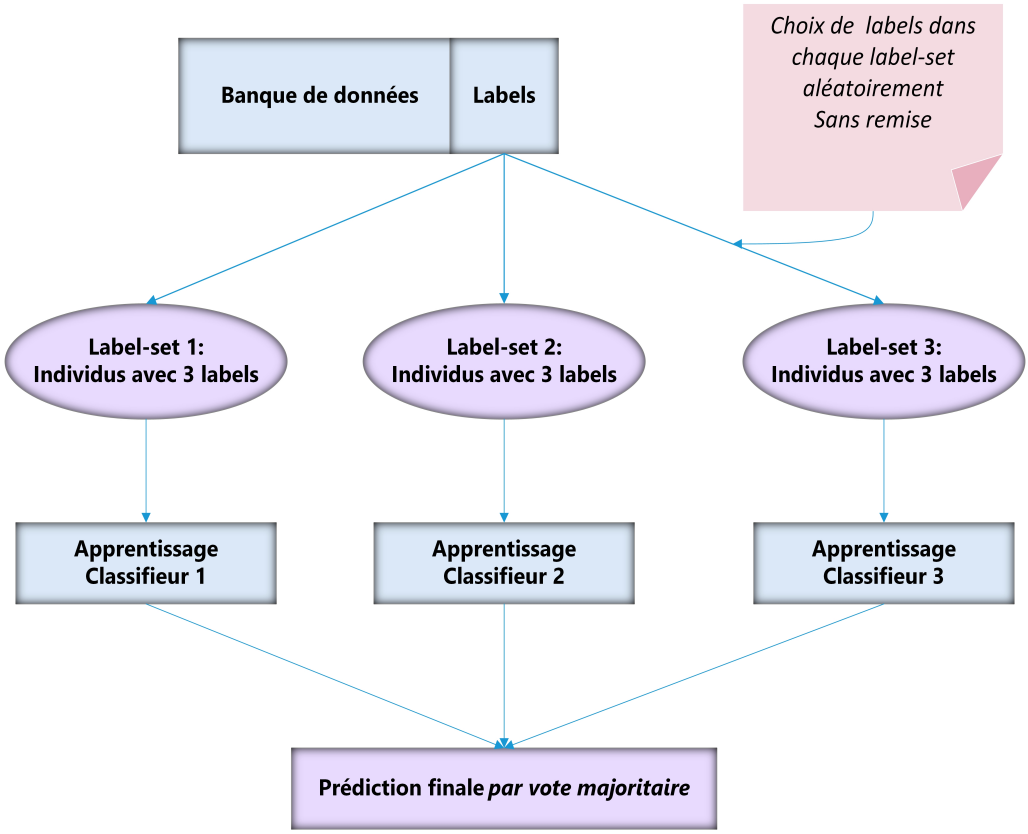


FIGURE 3.6 – Schéma explicatif de l'algorithme *RAkEL* overlapping

Le choix de labels dans chaque labels-sets se fait aléatoirement sans remise (sans doublons), mais les différents k label-sets peuvent se chevaucher.
Ci-dessous un exemple de classification par *RAkEL* Table 3.1 :

		Prédictions					
Modèles	**Label-sets**	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
h1	$\{\lambda_1, \lambda_2, \lambda_6\}$	1	0	-	-	-	1
h2	$\{\lambda_2, \lambda_3, \lambda_4\}$	-	1	1	0	-	-
h3	$\{\lambda_3, \lambda_5, \lambda_6\}$	-	-	0	-	0	1
h4	$\{\lambda_2, \lambda_4, \lambda_5\}$	-	0	-	0	0	-
h5	$\{\lambda_1, \lambda_4, \lambda_5\}$	1	-	-	0	1	-
h6	$\{\lambda_1, \lambda_2, \lambda_3\}$	1	0	1	-	-	-
h7	$\{\lambda_1, \lambda_4, \lambda_6\}$	0	-	-	1	-	0
	Vote moyen	3/4	1/4	2/3	1/4	1/3	2/3
	Prédiction finale	1	0	1	0	0	1

TABLE 3.1 – Exemple de classification par *RAkEL* overlapping avec k=3, nombre d'hypothèses=7 sur des données multi-labels ayant 6 labels

Nous avons exposé dans cette partie le principe de l'algorithme *RAkEL* ainsi que les 2 stratégies possibles pour le choix de k label-sets (*RAkEL* overlapping, *RAkEL* Disjoint). Notre contribution porte sur l'application d'une méthode d'ensemble (un cas particulier du Bagging), il s'agit des Forêts aléatoires.

Dans [57], Jesse Read indique que le choix de l'algorithme pour résoudre le problème de la classification multi-labels dépend de la nature du problème et précise 2 notions que nous recherchons dans notre travail (la rapidité et l'efficacité), qui peuvent être assurées en utilisant un algorithme se basant sur les arbres de décisions. Nous utilisons donc les Forêts aléatoires dans la méthode *RAkEL* pour confirmer cette supposition. Nous testons également le comportement de tel algorithme en changeant la stratégie d'agregation, pour cela nous utilisons : le vote majoritaire et le vote pondéré.

4.2 Les forêts aléatoires (*Random Forest*)

Les forêts aléatoires est un cas particulier du Bagging, introduit par Breiman en 2001 [58], il utilise les arbres de décisions comme outils de classification. Brieman a proposé cet algorithme afin d'améliorer le Bagging en ajoutant une seconde randomisation. L'objectif est donc de rendre plus indépendants les arbres de l'agrégation en ajoutant du hasard dans le choix des variables qui interviennent dans les modèles.

Une forêt aléatoire est un ensemble d'arbres de décision binaires dans lequel a été introduit de l'aléatoire. Soit $\theta_i \ldots \theta_M$ des variables, une forêt aléatoire est un ensemble de classifeurs $d_i(x, \theta_i), i = 1 \ldots M$, où les classifeurs d_i sont construits sur le même modèle que les arbres binaires.
Le nouveau classifieur correspondant à la forêt aléatoire est calculé en prenant la majorité de votes de chacun des classifeurs $d_i, i = 1 \ldots n$

Une forêt aléatoire génère un jeu d'arbres doublement perturbé car elle est construite en générant :

- des sous-ensembles aléatoires de caractéristiques pour chaque arbre.
- des sous-ensembles aléatoires de données d'apprentissages pour chaque arbre (comme dans la méthode Bagging).

Selon Breiman [58], les *Forêts* aléatoires sont en général plus efficaces que les simples arbres de décision.

De nombreux modèles de forêts aléatoires qui ont été créés correspondent à autant de manières d'incorporer de l'aléatoire dans les arbres telles que :

- ***Tree Bagging*** [21] introduit de l'aléatoire dans l'échantillon initial sélectionnant certains points plutôt que d'autres et laisse grandir l'arbre jusqu'à ce que chaque nœud comporte un unique élément.
- ***Random subspace*** [59] consiste à sélectionner à chaque nœud K variables de manière aléatoire et parmi celles-ci, à choisir celle qui minimise un certain critère.
- ***Random Forest-RI*** [58] consiste à mélanger le *CART* sans élagage, le *Bagging* et le *random subspace* pour chaque arbre.
- ***Random Select Split*** [60] sélectionne les K meilleures séparations et en choisit une parmi celles-ci de manière aléatoire. La position de la coupure est également calculée de manière aléatoire.

Dans notre travail, nous nous intéressons aux *Forêts* aléatoires de type Random Forest-RI [58]. Selon Robin Genuer dans [52], en pratique les Random Forests-RI améliorent les performances du Bagging (voir la comparaison des méthodes sur des données de référence dans Breiman (2001) [58]). L'explication heuristique de ces améliorations est due au fait

de rajouter l'aléatoire supplémentaire pour construire les arbres ce qui rend ces derniers encore plus différents les uns des autres, sans pour autant dégrader de façon significative leurs performances individuelles et le prédicteur agrégé est alors meilleur. Dans la partie qui suit nous allons expliqué en détail les notions fondamentales de ce type de *Forêts* aléatoires ainsi que les avantages qu'il présente.

Algorithme des forêts aléatoires

La forêt d'arbre de décision aléatoire est construite selon la procédure suivante [61] :

Algorithm 4 processus de construction des *Forêts aléatoires*

Entrée : Echantillon d'apprentissage :$Z = (x_1, y_1) \dots (x_n, y_n)$, x_i décrit par p variables explicatives, B : le nombre d'arbres formés dans la $Forêt$

pour b=1...B **faire**

Tirer un échantillon aléatoire z_b avec remise parmi Z

Estimer un arbre sur z_b avec randomisation des variables :

Pour la construction de chaque nœud de chaque arbre,tirer uniformément q variables parmi p pour former la décision associée au nœud.

fin pour

Sortie : $Forêt$ aléatoire contenant B arbres

le shéma 3.7 illustre le processus de RF introduit dans le pseudo code (4)

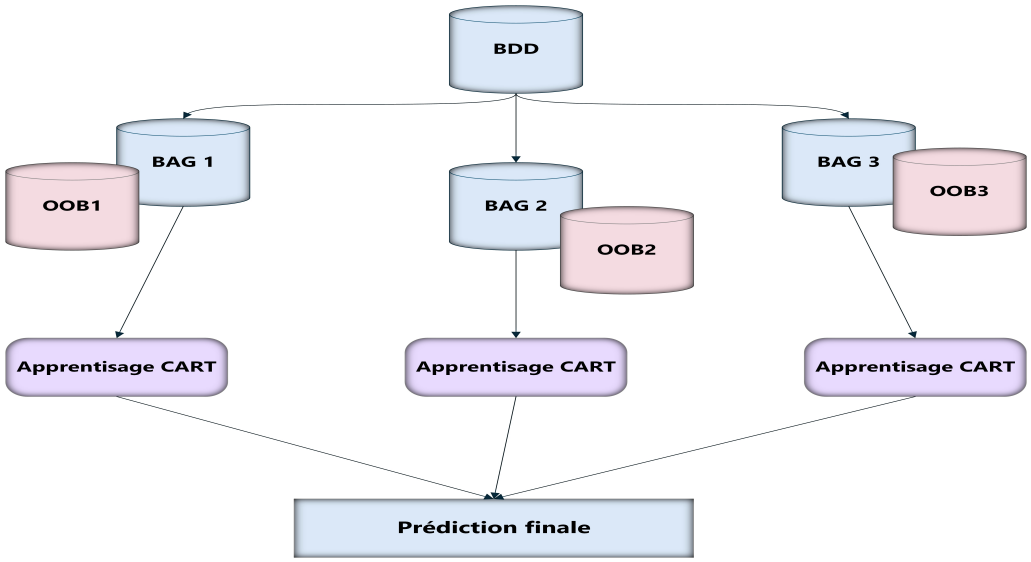


FIGURE 3.7 – Schéma explicatif de l'algorithme *fôrets* aléatoires(Random Forest)

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

1. Le paramètre le plus important est le nombre de variables choisie aléatoirement à chacun des nœuds des arbres. Il est nommé *mtry* [62]. Il peut varier de 1 à p (p :le nombre de toutes les variables de la banque d'apprentissage) et possède une valeur par défaut $\sqrt{p}$: en classification.
2. Le deuxième paramètre est le nombre d'arbres de la forêt, le choix le plus judicieux de ce paramètre étant après plusieurs expérimentations.

La prédiction finale par RF est obtenue grâce à un opérateur de combinaison qui est le vote majoritaire où chaque arbre participe au vote de la classe la plus populaire pour une donnée d'entrée x.

5 Conclusion

Dans ce chapitre nous avons présenté deux méthodes dédiées à la classification multi-labels, ayant des principes différents puisqu'elles appartiennent aux deux familles de méthodes d'adaptation et de transformation, il s'agit de *RAkEL* et *MLknn*.

Nous avons exposé le principe et les avantages de chacune ainsi que nos propositions pour leurs optimisations et améliorations. Premièrement, par le développement d'une nouvelle méthode qui est le ***Bagged MLknn*** pour améliorer la méthode d'adaptation *MLknn*. Deuxièmement, la mise en forme de ***RAkEL Random Forest***, pour tester les performances de la méthode *RAkEL* avec les *Forêts* aléatoires. Les détails de ces deux contributions seront présentés dans le chapitre suivant.

Chapitre 4

Expérimentations et Résultats

1 Introduction

Dans les chapitres précédents, nous avons présenté brièvement nos propositions afin d'améliorer les performances de deux méthodes d'adaptation et de transformation respectivement *Mlknn* et *RAkEL*, par la création d'un *Bagged Mlknn* et *RAkEL Random Forest*. Dans la dernière partie de ce travail, nous nous attelons aux expérimentations et interprétations des résultats appliqués sur trois jeux de données multi-labels couramment utilisés dans la littérature et qui sont : *Yeast*, *Scene*, et *Genbase*.

Le schéma (4.1) ci dessous donne un bref aperçu de ce que nous allons détailler par la suite.

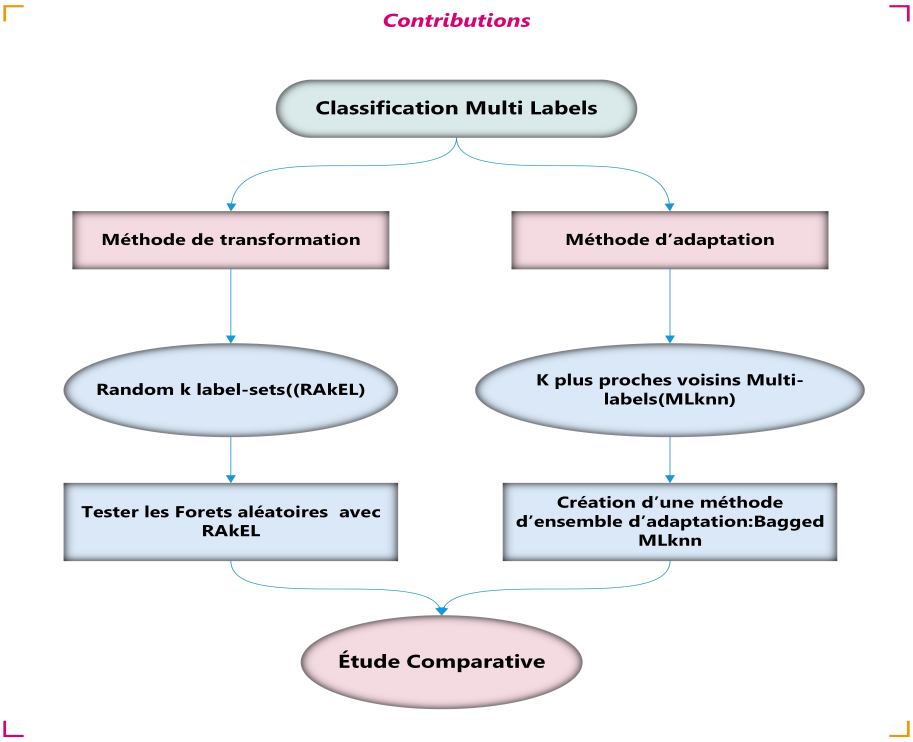


FIGURE 4.1 – Plan de nos contributions dans ce projet de fin d'études

2 Contributions

Tout au long de ce projet de fin d'études, l'objectif de notre travail était d'étudier les algorithmes d'ensembles de classification pour les données multi-labels, en particulier comparer les 2 algorithmes *MLknn* et *RAkEL* appartenant aux deux familles d'adaptation et de transformation respectivement. Les contributions que nous proposons dans cette partie sont :

– L'application d'un *Bagged MLknn* pour améliorer les performances de l'algorithme d'adaptation *MLknn* individuel.
– L'utilisation des *Forêts* aléatoires comme méthode d'apprentissage pour la méthode $RAkEL_o$.
– Amélioration des résultats de chaque méthode par remplacement du vote ordinaire classique " vote majoritaire " par le vote pondéré.

2.1 Contribution 1 : (*Bagged MLknn*)

Initialement, le *Bagging* a été introduit avec comme règle de base, un arbre de décision. Cependant, le schéma est très général et peut être appliqué à d'autres règles de bases : par exemple, la règle du plus proche voisin.

Cette méthode du plus proche voisin "baggé" a été récemment étudiée par Biau & Devroye (2010) dans [63], dans un cadre de régression pour la classification mono-label. Cet article établit la consistance de la méthode du plus proche voisin "baggé".

Cette étude illustre à merveille les bienfaits des méthodes d'ensembles : partant d'une règle basique assez pauvre (la règle du plus proche voisin qui n'est pas consistante), le *Bagging* la transforme en une règle aux très bonnes propriétés asymptotiques (consistance et vitesse optimale de convergence).

De plus, le *Bagging* présente beaucoup d'avantages [64] car il est :
– simple à mettre en place.
– s'adapte facilement à n'importe quelles méthodes d'apprentissage.
– permet de réduire l'impact du choix de l'ensemble d'apprentissage sur les résultats de classification.

C'est pour ces raisons que dans cette contribution nous avons mis en oeuvre une approche *Bagged MLknn*. Selon [5], les résultats obtenus par application de cet algorithme montre leur simplicité et leur facilité. De ce fait, l'utilisation de plusieurs *MLknn* simultanément pour faire la prédiction de labels donne forcément une amélioration des performances du classifieur individuel. De là est née l'idée d'utiliser un *Bagged MLknn*.

Le schéma 4.2 suivant présente le processus de création de *Bagged MLknn* :

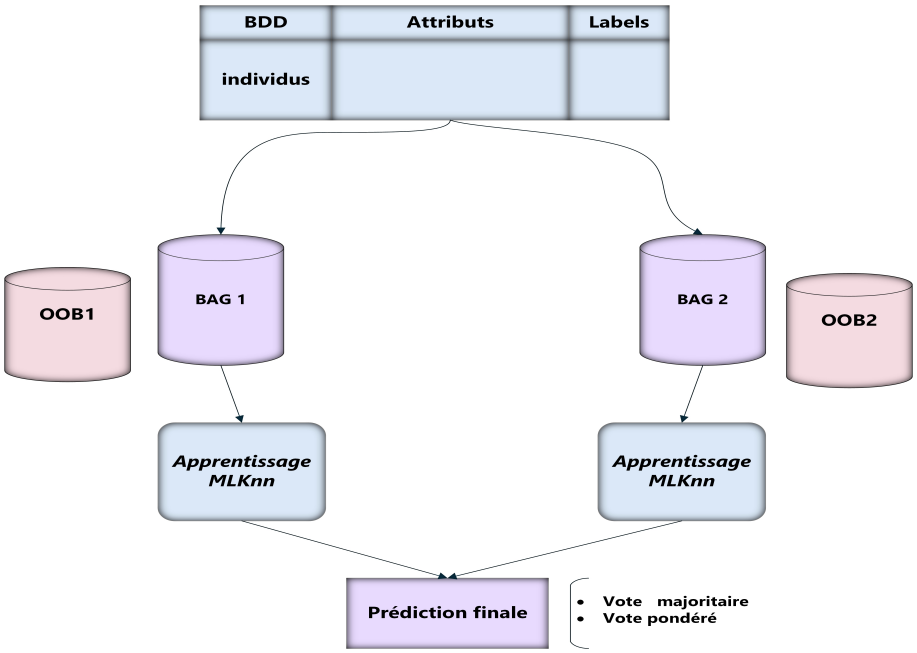


FIGURE 4.2 – Processus de classification avec *Bagged MLknn*

2.2 Contribution 2 : (*RAkEL* avec *RF*)

Dans [14], les auteurs ont testé l'algorithme *RAkEL* en utilisant comme prédicteurs les algorithmes comme : *SVM*, *C4.5*, *Naive Bayes*. La contribution de notre travail consiste à remplaçer ces 3 classifieurs par la méthode *Random forest* qui a montré son efficacité et ses avantages.

Un des avantages des forêts aléatoires [52] est qu'elles sont très performantes aussi bien pour des problèmes classiques (où le nombre d'observations $\ggg$ nombre d'attributs) que pour des problèmes de grandes dimensions (où le nombre d'observations $\lll$ nombre d'attributs), elles sont donc très utiles dans la classification multi-labels puisque généralement les banques de données sont volumineuses.

La méthode *Random forest* permet d'améliorer la méthode *RAkEL* qui a le *même* principe que le *Bagging* puisqu' elle utilise un ensemble de prédicteurs et la prédiction finale est l'agrégation de toutes les prédictions. Selon Brieman [58], les *Forêts* aléatoires améliorent les performances du Bagging. L'explication heuristique de ces améliorations est dans le fait de rajouter un aléa supplémentaire pour construire les arbres, ce qui rend ces derniers encore plus différents les uns des autres ; sans pour autant dégrader de façon significative leurs performances individuelles. Le prédicteur agrégé est alors meilleur.

La plupart des méthodes d'ensembles construisent une collection de prédicteurs qui sont des versions perturbées d'une règle de base. La perturbation introduite doit alors réaliser un compromis entre deux situations :

– Une trop grande perturbation dégrade les prédicteurs individuels et le prédicteur agrégé est alors mauvais.
– Une trop petite perturbation induit des prédicteurs individuels trop similaires entre eux et le prédicteur agrégé n'apporte alors aucune amélioration.

Les excellents résultats des *Forêts* aléatoires en pratique laissent penser qu'elles (avec le paramètre *mtry* bien choisi) réalisent un bon compromis, en injectant la "bonne dose" d'aléatoire.

Les schémas 4.3 et 4.4 ci-dessous expliquent le principe de *RAkEL* avec les *Forêts* aléatoires pour deux arbres :

Étape 1 : génération de k label-sets aléatoirement sans remise.

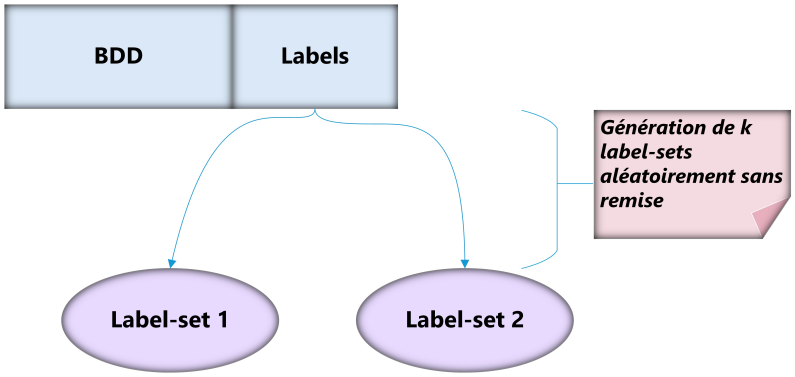


FIGURE 4.3 – Schéma explicatif de l'algorithme *RAkEL* avec *RF* (Etape 1)

Étape2 : Bagging selon les labels-sets généré dans l'étape 1.

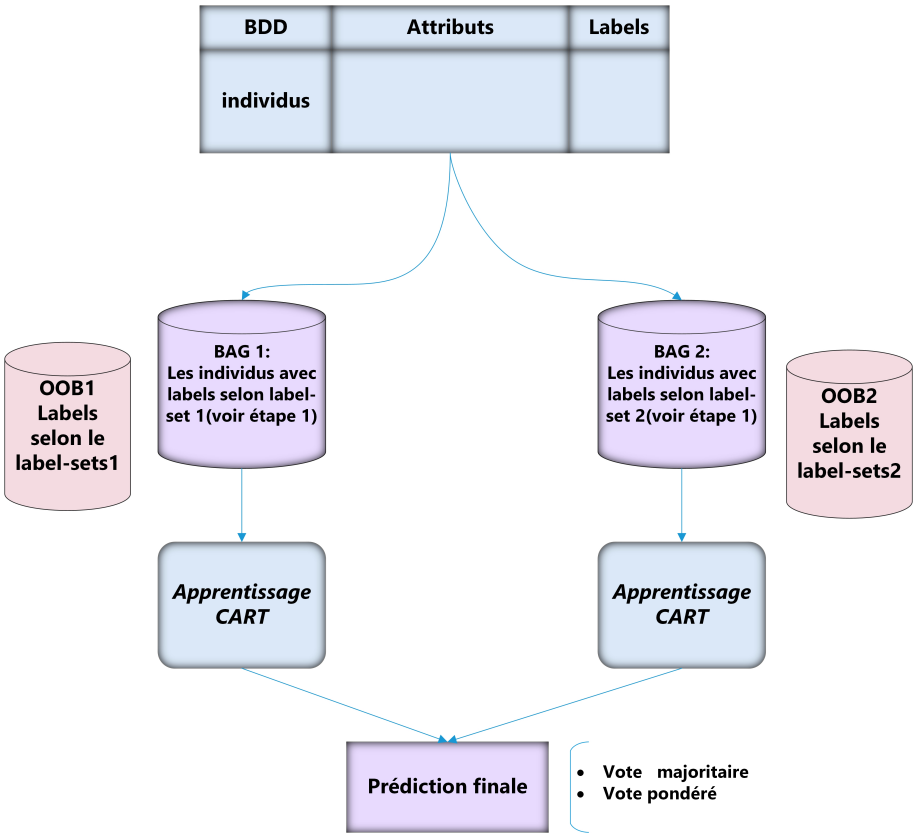


FIGURE 4.4 – Schéma explicatif de l'algorithme *RAkEL* avec *RF* (Etape 2)

3 Banques de données

Dans la littérature, il existe plusieurs banques de données multi-labels pour l'évaluation des algorithmes de classification dans ce domaine. Dans ce travail et afin d'évaluer les performances des approches proposées, nous avons choisi trois banques de données du dépôt Mulan [65] : *Yeast*, *Scene*, et *Genbase* pour comparer les résultats du *Bagged MLknn* et *RAkEL Random Forest* avec *MLknn* [5] et *RAkEL* [14] respectivement. Le choix de ces trois banques de données a été fait selon le nombre de labels contenu dans chacune d'elles : *Scene* avec seulement 6 labels, la banque *Yeast* avec un nombre moyen de labels égale à 14 et la banque *Genbase* est la plus grande contenant 27 labels. De ce fait nous avons voulu voir et tester le comportement de nos approches sur des banques de données qui diffèrent dans le nombre de labels.

3.1 La banque de données *Yeast*

Le jeu *Yeast* [10] contient des données concernant les analyses fonctionnelles des gènes chez la levure Saccharomyces cerevisiae. La banque d'apprentissage contient 2417 gènes décrits par 103 attributs et regroupés en 14 classes, le nombre moyen de chaque label pour chaque gène est de 4.25. Cette banque constitue un outil d'évaluation très important qui a été étudié dans la littérature [66, 67].

3.2 La banque de données *Scene*

Les données *Scene* [68], contiennent six catégories de scènes naturelles : Lever/coucher du soleil, Plage/désert, Mer/Paysages maritimes, Montagnes, Verdure/Forêt et Ville. Cette banque de données contient 2407 images où chaque image est décrite par 294 attributs, le nombre de labels pour ce jeu de données est égale à 6.

3.3 La banque de données *Genbase*

Le jeu de données *Genbase* [69] comporte des données biologiques qui présentent des analyses fonctionnelles de gènes où chaque instance est une protéine et les labels représentent les classes de chaque instance. Cette banque de données contient 662 exemples avec 27 labels, et caractérisé par 1185 attributs sans compter la première colonne qui a été supprimée de la banque car elle présente l'identificateur de chaque gène qui est inutile pour la classification.

3.4 A quel point la banque de données est-elle multi-labels ?

C'est une question qui doit *être* prise en considération dans le domaine de la classification multi-labels, car dans certaines applications le nombre de labels pour chaque exemple est petit par rapport au nombre total de labels L et dans d'autres banques il est très grand et cela influe sur les performances des méthodes de classification [7].

Pour bien étudier le contenu de la banque de données, des mesures statistiques sont proposées dans la littérature : *'Label Cardinality'*, *Label Density* et *label-sets Distinct*, *Diversity* et qui sont définies comme suit :

Soit D l'ensemble de données multi-labels contenant $\mid D \mid$ exemples d'apprentissages tel que :$(x_i, Y_j), i = 1 \cdots \mid D \mid$ et j=$1 \cdots \mid L \mid$

Cardinalité des labels : *Label cardinality* (LC [7]) de D est utilisé pour quantifier le nombre moyen de labels qui caractérise chaque exemple des données multi-labels lors de l'apprentissage dans D.

$$\text{Label-Cardinality}=\frac{1}{m}\sum_{i=1}^{m} \mid Y_i \mid$$

Densité des labels : *Label Density* (LD [7]) de D prend en considération le nombre de labels du domaine, c'est le nombre moyen de labels qui caractérise les exemples des données multi-labels lors de l'apprentissage dans D divisé par le nombre de labels de toute la banque L.

$$\text{Label-Density}=\frac{1}{m}\sum_{i=1}^{m} \frac{|Y_i|}{L}$$

Diversité : C'est le nombre total de labels dans la banque.

Deux ensembles de données avec un même «*Label Cardinality*» mais avec une grande différence du «*Label Density*» cause un comportement différent du modèle d'apprentissage multi-labels.

Nombre de labels-sets différents dans la banque : *Label-sets distinct* [7], calcule le nombre de combinaisons possibles de labels dans le jeu de données, c'est aussi un critère très important surtout pour certains algorithmes de transformation qui opèrent sur des label-sets. Et donc ça influe sur les performances de l'algorithme de classification, citons par exemple l'algorithme appartenant à la famille de transformation : Random k label-sets($RAKEL$).

Le tableau 4.1 ci-dessous présente les statistiques [65] des 3 banques de données utilisées dans ce travail, à savoir : **Yeast**, **Scene**, **Genbase**

BDD	Exemples	Attributs		Labels	*LD*	*LC*	*Distinct*
		Numérique	Discret				
Yeast	2417	103	0	14	0.30	4.23	198
Scene	2407	294	0	6	0.17	1.07	15
Genbase	662	0	1185	27	0.05	1.25	32

TABLE 4.1 – Quelques statistques sur les banques de données

D'après les statistiques dans le tableau 4.1 , nous constatons que la banque *Yeast* est plus dense (LD=0.30) que les banques *Genbase*(0.05) et *Scene*(0.17) et même plus diverse avec 4 labels pour chaque exemple en moyenne contrairement aux banques *Scene* et *Genbase* qui ont en moyenne 1.5 label pour chaque exemple.

4 Mesures d'évaluations

L'évaluation d'un système d'apprentissage multi-labels se différencie de celle d'un système d'apprentissage mono-label, pour cela plusieurs critères existent. La figure 4.5 résume les différentes mesures utilisées dans la littérature [8].

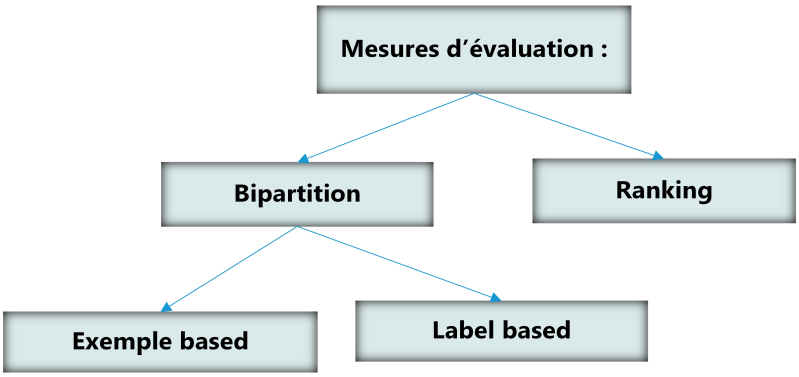


FIGURE 4.5 – Mesures d'évaluation pour les données multi-labels

Dans ce travail, nous nous intéressons en particulier à la famille Bipartitions based puisque nous sommes dans le cadre de la classification. Dans cette famille de mesures d'évaluations, nous distinguons 2 groupes :

Exemple based

Les mesures d'évaluation sont calculées en considérant les exemples d'apprentissages, ces mesures calculent les performances obtenues pour chaque exemple avec tous les labels et à la fin une moyenne de tous les résultats est calculée (Figure 4.6).

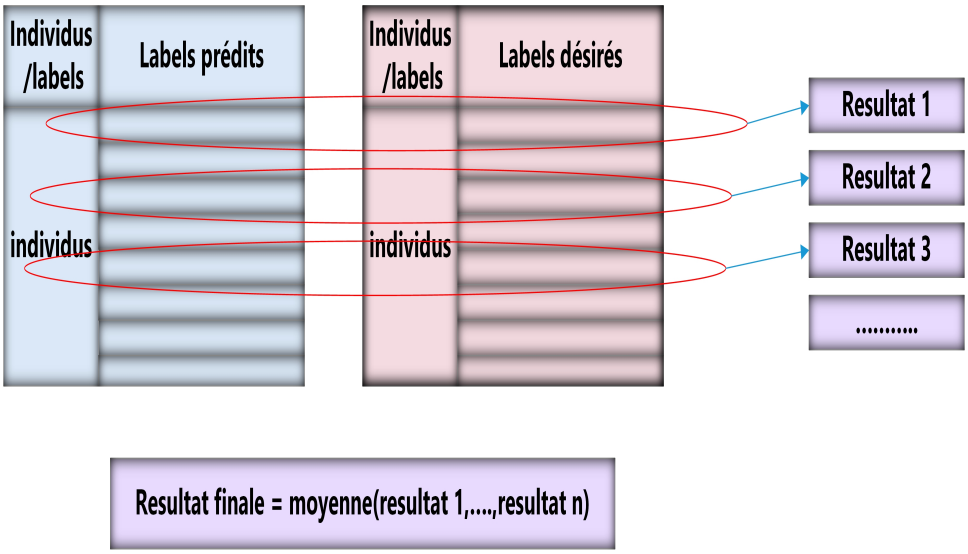


FIGURE 4.6 – schéma explicatif du principe d'évaluation 'Exemple based'

Label based

Calcule les performances obtenues pour chaque label séparément avec tous les exemples et fait une moyenne des résultats retrouvés pour tous les labels (Figure 4.7).

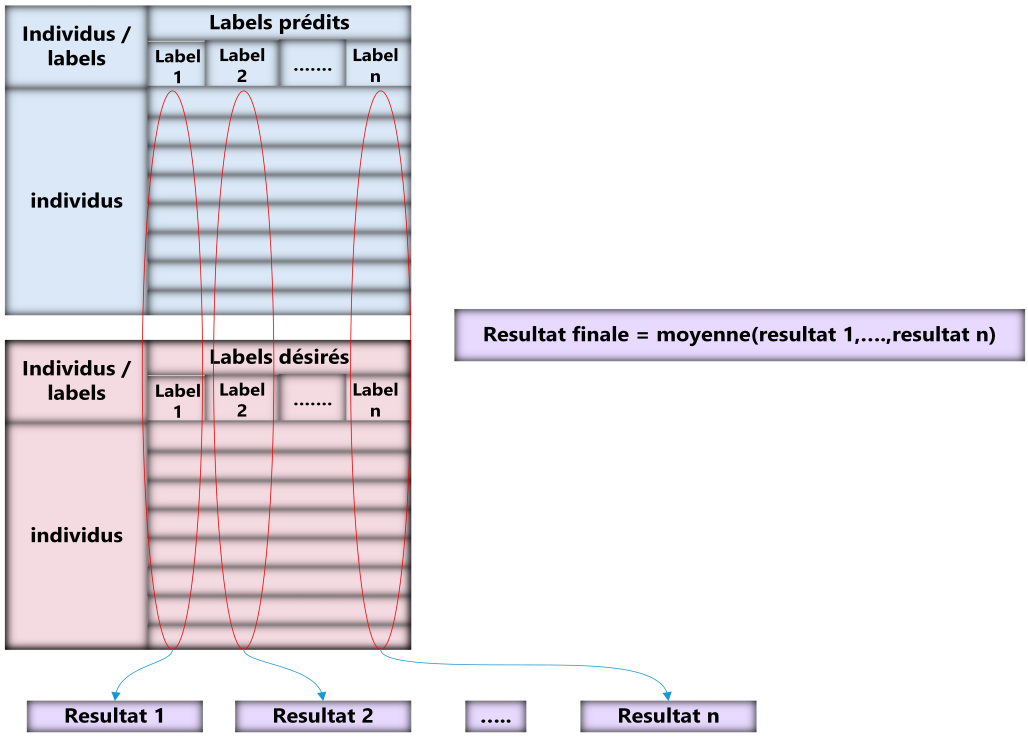


FIGURE 4.7 – schéma explicatif du principe d'évaluation 'Label based'

Selon Min-Ling Zhang [70], le choix de la famille de mesures d'évaluation se fait en considérant le but de l'apprentissage, il est préférable d'utiliser *Label based* dans les domaines d'applications tels que : la recherche d'information, et *Example based* pour le domaine de la classification.

Dans ce travail, puisque nous nous intéressons au domaine de la classification multi-labels, nous avons choisis 4 mesures appartenant au 1^{er} groupe et qui sont : *Accuracy*, *Fmesure*, *Subset Accuracy* et *Hamming Loss*.

Étant donné un ensemble de test S= $(x_1, Y_1), \ldots, (x_M, Y_M)$ et un classifieur H. Soient $\hat{Y}_i = H(x_i)$ l'ensemble de labels prédits pour x_i et Y_i le vrai ensemble de labels associés à x_i. Nous donnons ci-dessous les définitions de quelques métriques d'évaluation utilisées :

4.1 Accuracy

Cette métrique mesure le degré de similarité entre l'ensemble de classes prédit et l'ensemble de labels désiré [16].

$$\text{Accuracy}=\frac{1}{M}\sum_{i=1}^{M}\frac{|Y_i\cap\hat{Y}_i|}{|Y_i\cup\hat{Y}_i|}$$

4.2 Mesure F

La mesure F correspond à une moyenne harmonique de deux autres métriques : la précision et le rappel [16].

$$\text{mesure F}=\frac{1}{M}\sum_{i=1}^{M}\frac{2|Y_i\cap\hat{Y}_i|}{|Y_i|+|\hat{Y}_i|}$$

4.3 Subset Accuracy

est une mesure très stricte qui considère la classification correcte si tous les labels prédits par le classifieur sont corrects [8].

$$\text{Subset Accuracy}=\frac{1}{M}\sum_{i=1}^{M} I(\mid Y_i \mid=\mid \hat{Y}_i)$$

4.4 Mesure du Coût de Hamming

Cette métrique évalue combien de fois une paire exemple-classe est mal classée, plus cette métrique est faible plus les résultats sont bons. Les performances sont dites parfaites lorsque la valeur est nulle [16].

$$\text{Coût du Hamming } =\frac{1}{M}\sum_{i=1}^{M}(\mid Y_i \mid\triangle\mid \hat{Y}_i \mid)$$

où $\triangle$ est la différence symétrique entre les deux ensembles(les vrais labels et les labels prédits).

5 Résultats et Discussions

5.1 Les k plus proches voisins multi-labels mono classifieur (*MLknn* simple)

Dans cette partie, nous évaluons l'algorithme *MLknn* [5] sur les trois banques : *Genbase*, *Scene* et *Yeast* en appliquant 10 validations croisées. Nous varions le nombre de k plus proches voisins k=4, 6, 8, 10 et 12 en gardant la même valeur par défaut du paramètre $smoothing = 1$. Les résultats réalisés sont calculés par les mesures d'évaluations :*Accuracy*, *Subset Accuracy*, *Fmesure*, *Hamming Loss*.

De la variation des différents k, nous remarquons que ce paramètre $'k'$ n'influe pas de manière significative sur les performances de l'algorithme *MLknn*. Cette constatation réaffirme les résultats de Zhang & Zhou dans leurs article [5], pour cela nous présentons dans le tableau 4.2 pour chaque banque, les résultats obtenus avec une seule valeur de $k = 4$ (les valeurs après ±désignent des valeurs de variances).

	Genbase	Yeast	Scene
Accuracy	4, 8±0.1502	30, 2 ±0, 4271	17, 87±0, 2706
Subset Accuracy	0	0	0
Fmesure	9, 16 ±0.5003	46, 38 ±0.5958	30, 32 ±0, 5612
Hamming Loss	95 ±1, 50E-05	70 ±4, 27E-05	82 ±2, 71E-05

TABLE 4.2 – Résultats obtenus par *MLknn* sur les trois banques de données

Nous constatons du tableau 4.2 que les résultats sont de faibles performances, la mesure *Subset Accuracy* est égale à zéro pour toutes les banques. Cette mesure est la mesure la plus stricte car elle s'incrémente "si et seulement si tous les labels pour une instance sont bien classés", les valeurs égales à zéro de cette mesure peuvent s'expliquer par le fait que le voisinage d'une instance donnée déterminé par cet algorithme n'est pas le plus approprié.

La même remarque pour les valeurs de la mesure *Accuracy* et *Fmesure* qui réalisent des résultats médiocres puisqu'ils ne dépassent même pas 50% de bonne reconnaissance,

ceci est plus constaté pour la banque *Genbase* qui a de très faibles valeurs de la mesure *Accuracy* et *Fmesure*.

Pour la mesure *Hamming Loss* nous remarquons que les valeurs sont très élevées ce qui indique une grande perte d'information.

Afin d'améliorer ces performances, nous avons mis en œuvre une nouvelle approche appelée *Bagged MLknn*, qui utilise plusieurs classifieurs de type *MLknn*. Chacun fait la prédiction de labels pour un exemple séparément et la prédiction finale consiste à combiner les résultats de tous les classifieurs selon une stratégie : par vote majoritaire ou pondéré.

5.2 Ensemble de k plus proches voisins en classification multi-labels (*Bagged MLknn*)

Application avec la banque Genbase

Les résultats d'application du classifieur *Bagged MLknn* sur la banque **Genbase** sont représentés dans la figure 4.3 et 4.4. Notons que la mesure *Hamming Loss* est égale à zéro *même* en augmentant le nombre de classifieurs (MLknn).

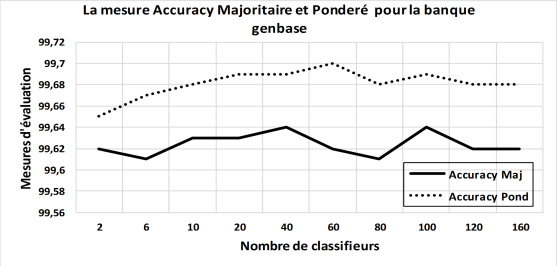


Figure 4.3 – Résultats de la mesure *Accuracy* Majoritaire et Pondéré pour la banque *Genbase*

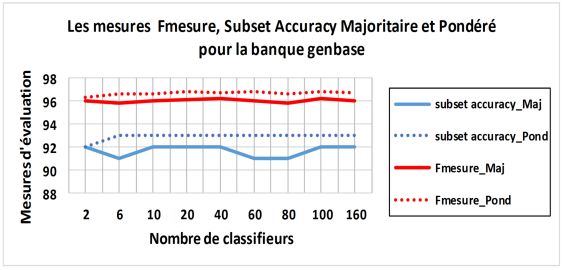


Figure 4.4 – Résultats des mesures *Fmesure* et *Subset Accuracy* Majoritaire et Pondéré pour la banque *Genbase*

En comparant les résultats de la première partie (*MLknn* simple) pour la banque *Genbase* avec ces résultats obtenus en utilisant *Bagged MLknn*, nous constatons une grande amélioration. La mesure *Accuracy* dans la première partie était de 4.8% seulement alors que dans cette partie nous avons atteint les 99.64 % en utilisant 100 arbres (voir la figure 4.3) par vote majoritaire. Notre proposition d'utilisation du vote pondéré à la place du vote majoritaire a donné de bons résultats puisque nous observons clairement cette amélioration dans la *même* figure, la *même* remarque a été notée pour les mesures *Fmesure* et *Subset accuracy* où le vote pondéré améliore grandement les performances avec le vote majoritaire.

Dans la figure 4.4 nous apprécions les résultats obtenus pour les 2 mesures *Fmesure* et*Subset Accuracy*, dont les résultats dans la première partie étaient très faibles et *même* égales à zéro pour la mesure *Subset Accuracy*. Cependant, grâce à *Bagged MLknn* nous avons pu atteindre pour cette mesure un taux de 93% par l'agrégation pondérée ce qui indique que 93% des exemples de la banque ont été simultanément bien classés par rapport à tous les labels ce qui est très intéressant.

Concernant la mesure *Hamming Loss* nous remarquons une large amélioration. Les valeurs par application de *MLknn* simple étaient très élevées cela implique une grande perte d'information, alors que par application de *Bagged MLknn* les mesures de *Hamming Loss* étaient égales à zéro et c'est le cas idéale pour une telle mesure.

Application avec la banque Yeast

La figure 4.5 représente les résultats obtenus pour la mesure *Accuracy* et *Fmesure* sur la banque *Yeast*, et la figure 4.6 représente les résultats obtenus pour *Hamming Loss* et *Subset Accuracy*.

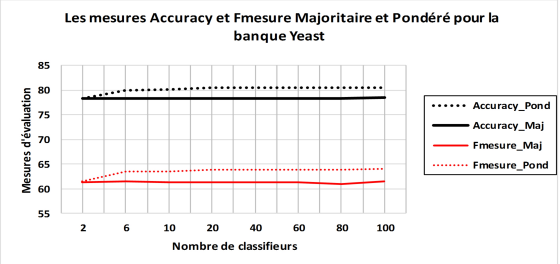


Figure 4.5 – Résultats des mesures *Accuracy* et *Fmesure* Majoritaire et Pondéré pour la banque *Yeast*

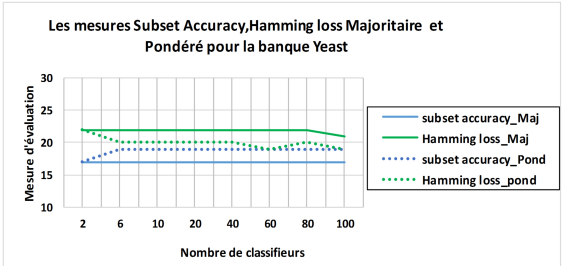


Figure 4.6 – Résultats des mesures *Hamming Loss* et *Subset Accuracy* Majoritaire et Pondéré pour la banque *Yeast*

Nous remarquons une amélioration pour toutes les mesures par rapport aux résultats obtenus avec *MLknn* simple (tableau 4.2). Dans la figure 4.5 nous observons clairement que le taux de la mesure *Accuracy* qui était avant de 30% (voir tableau 4.2) a augmenté considérablement jusqu' à 80% en utilisant le vote pondéré, *même* chose pour *Fmesure* qui est passé de 46% à environ 65% ce qui indique un bon compromis entre les deux mesures de précision et de rappel.

Concernant les deux mesures *Subset Accuracy* et *Hamming Loss* notons également une grande amélioration. Pour la mesure *Subset Accuracy* passant d'une valeur égale à zéro à un taux de 19% par le vote pondéré, ceci nous confirme que les capacités du classifieur MLknn ont été améliorées par la création du Multi classifieurs MLknn (*Bagged MLknn*). Quant à la mesure *Hamming Loss* nous remarquons qu'il y'a une diminution de 70% (voir tableau4.2)à 19% (par vote pondéré) chose qui est très importante puisque des valeurs faibles de *Hamming Loss* indiquent une bonne classification et le nombre de paires de labels mal classés est réduit.
Nous distinguons également que dans tous les cas de mesures, la statégie de vote que nous avons proposé (vote pondéré) donne de meilleurs résultats et améliore ceux réalisés par le vote majoritaire.

Application avec la banque Scene

Finalement, nous discuttons les résultats pour la banque **Scene**. La figure 4.7 et la figure 4.8 représentent les taux obtenus par rapport à toutes les mesures d'évaluations.

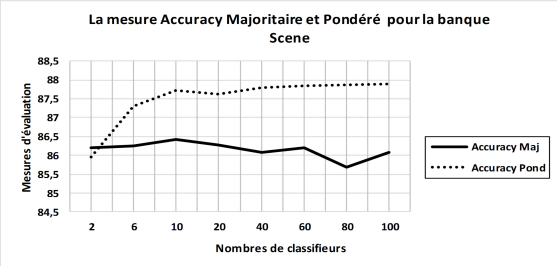


Figure 4.7 – Résultats des mesures *Accuracy* Majoritaire et Pondéré pour la banque *Scene*

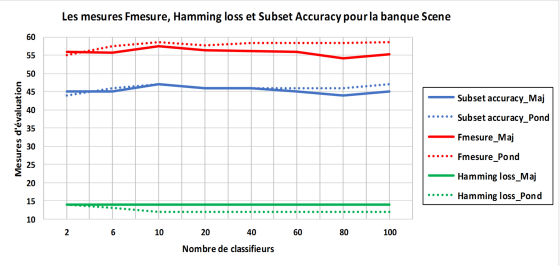


Figure 4.8 – Résultats des mesures *Hamming Loss*, *Subset Accuracy* et *Fmesure* Majoritaire et Pondéré pour la banque *Scene*

Dans la figure 4.7, la courbe qui représente la mesure *Accuracy* par vote pondéré est nettement plus représentative et plus lisse que celle du vote majoritaire. Le taux d'*Accuracy* réalisé *grâce* à la méthode *Bagged MLknn* proposée améliore les résultats de *MLknn* (voir tableau 4.2) en passant de 17 % à 88 % de bonne reonnaissance.

Il est clairement constaté dans la figure 4.8, une large élévation des valeurs de *Subset Accuracy* qui a atteint *grâce* à la méthode Bagged MLknn les 46% après qu'elles étaient des valeurs égales à zero en utilisant MLknn simple (voir tableau 4.2). *Même* chose pour les deux mesures :*Fmesure*(en passant de 30% à 45%) et pour le *Hamming Loss*,nous remarquons une baisse très représentative de 82% à 12% ce qui implique une faible perte d'information (voir tableau 4.2).

Discussions

D'après les résultats présentés sur les trois banques, nous pouvons apprécier l'efficacité de l'algorithme proposé qui représente une amélioration de la méthode d'adaptation *MLknn*.
Ces performances s'expliquent par le fait d'utiliser une méthode d'ensemble (*Bagged MLknn*) qui garantie deux notions importantes :

– La précision : les prédictions obtenues sont très précises puisque on combine la réponse de plusieurs prédicteurs (*même* individuelement faible (puisque nous avons déja vu qu'un mono classifieur *MLknn* est non efficace).

– L'efficacité : un problème complexe peut *être* décomposé en de multiples sous problèmes plus simples à résoudre (approche diviser pour Régner)

5.3 La méthode *RAkEL RF (Random k label-set Random forest)*

Dans cette partie, nous présentons les résultats d'évaluation de l'algorithme de transformation *RAkEL Random Forest* avec vote majoritaire et pondéré pour les trois banques de données :*Yeast*, *Genbase* et *Scene* en utilisant 10 validations croisées. Nous avons varié le nombre d'arbres dans la $f\hat{o}ret$(M) ainsi que le nombre de k label-sets selon le nombre de labels total dans chaque banque, sachant que les banques dont nous disposons contiennent un nombre de labels : Grand pour la banque *Genbase*, Moyen pour la banque *Yeast* et Petit pour la banque *Scene*.

Finalement, nous évaluons ces résultats en utilisant les mêmes mesures d'évaluation que la partie *Mlknn* : *Accuracy*, *Subset Accuracy*, *Fmesure* et *Hamming Loss*.

Résultats avec la banque *Yeast*

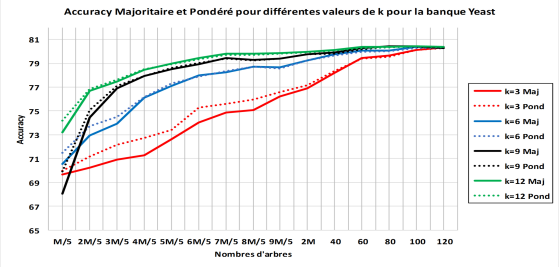


Figure 4.9 – Résultats des mesures *Accuracy* obtenus par *RAkEL RF* pour la banque *Yeast*

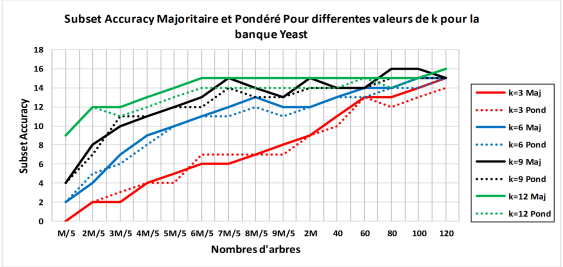


Figure 4.10 – Résultats de *Subset Accuracy* obtenus par *RAkEL RF* pour la banque *Yeast*

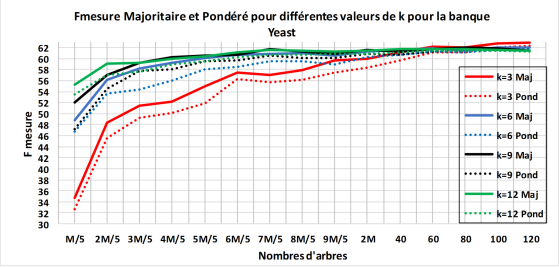


Figure 4.11 – Résultats de *Fmesure* obtenus par *RAkEL RF* pour la banque *Yeast*

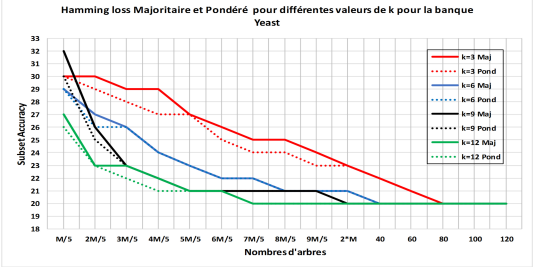


Figure 4.12 – Résultats de *Hamming Loss* obtenus par *RAkEL RF* pour la banque *Yeast*

Discussions et Interprétations

La figure (4.9) représente le taux d'*Accuracy* pour la banque *Yeast* en variant le nombre de k label-sets. L'utilisation de la méthode d'ensembles Random Forest comme classifieur de base pour *RAkEL* a donné des résultats satisfaisants, le meilleur taux de classification obtenu est d'environ 80% nous remarquons dans la figure que ce taux est atteint pour différentes valeurs de k labels-sets mais différemment. Dans le cas de k=12 et k=9 ce taux est obtenu en utilisant un nombre d'arbres réduits de $6M/5$ et $7M/5$ respectivement.

Comme première observation, plus le nombre de k label-sets augmente et s'approche du nombre réel de labels qui est égal à 14 pour la banque *Yeast* plus le taux de classification augmente. Ce taux peut être également atteint avec utilisation d'un nombre plus faible de k label-sets mais dans ce cas de figure l'utilisation d'un grand nombre d'arbres est requis.

De ce fait, nous pouvons conclure que le nombre d'arbres peut rattraper la précision perdue en utilisant un nombre de k label-sets minimal. Un autre cas qui affirme cette constatation est le fait d'utiliser un nombre de k label-sets très petit k=3, pour avoir un bon taux de classification nous avons varié le nombre d'arbres de $M/5$ jusqu'au 120 et c'est

l'un des avantages des méthodes d'ensembles puisque nous combinons la réponse de plusieurs prédicteurs (*même* faiblement efficaces), les prédictions obtenues sont très précises.

En comparant les résultats obtenus par vote pondéré et majoritaire, nous pouvons apprécier une nette amélioration des résultats par vote pondéré, mais seulement dans le cas d'un nombre minimal de k label-sets et avec utilisation d'un nombre faible d'arbres dans la $Forêt$ aléatoire. Dans le cas des valeurs petites de k label-sets et de nombres d'arbres, les prédicteurs ne donnent pas de meilleurs résultats. Il en ressort donc logiquement que combiner les prédictions de chaque arbre en utilisant le vote majoritaire- où chaque arbre participe au vote de la classe la plus populaire ne donne pas de bons résultats.

Par contre, dans le cas du vote pondéré *même* avec utilisation d'un nombre réduit de k label-sets et de nombre d'arbres, la combinaison des prédictions qui sont pondérées par les performances locales de chaque arbre permet d'avoir un classifieur très performant.

Pour des valeurs de k labels-sets élevées, le vote majoritaire donne presque les *mêmes* résultats que le vote pondéré. Car pour un grand nombre de labels dans chaque labels-sets les classiffieurs deviennent très puissants *même* en utilisant un nombre réduit d'arbres dans la $Forêt$. De ce fait, combiner leurs prédictions par vote majoritaire donne de bons résultats équivalents à ceux du vote pondéré.

Dans les figures (4.10, 4.11, 4.12), nous confirmons les remarques citées ci-dessus concernant la relation entre le nombre de k label-sets et le nombre d'arbres. Puisque nous observons la *même* chose pour les mesures :*Subset Accuracy*, *Fmesure* et *Hamming Loss*, si le nombre de k label-sets est réduit, nous devons alors utiliser un grand nombre d'arbres pour avoir de bonnes performances. Dans le cas contraire, un nombre réduit d'arbres est suffisant pour atteindre une bonne reconnaissance.

Pour choisir entre ces 2 alternatives, nous pouvons évoquer le temps de calculs nécessaires pour chacune. L'utilisation d'un grand nombre de k label-sets avec peu d'arbres nécessite un temps réduit d'apprentissage et de test, contrairement au deuxième cas, où l'utilisation d'un nombre élevé d'arbres pour rattrapper la précision perdue à cause de l'utilisation d'un nombre faible de k labels-sets nécessite un temps de calculs considérable.

Résultats avec la banque *Scene*

Les résultats d'évaluation de la méthode *RAkEL RF* sur le jeu de données *Scene* sont présentés dans les figures ci-dessous. Nous utilisons des valeurs de k label-sets égales à :3, 4, 5. Ce choix a été fait en respectant le nombre de labels dans la banque qui est de 6 labels.

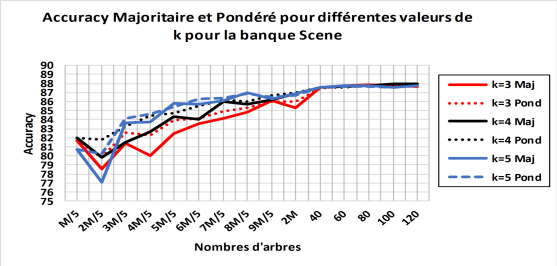


Figure 4.13 – Résultats de la mesure *Accuracy* obtenus par *RAkEL RF* pour la banque *Scene*

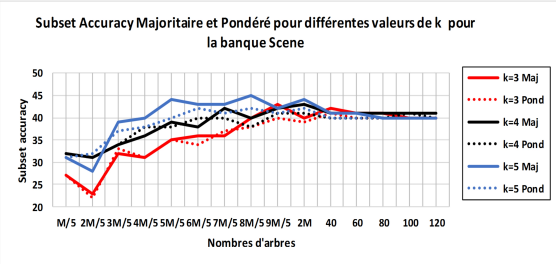


Figure 4.14 – Résultats de *Subset Accuracy* obtenus par *RAkEL RF* pour la banque *Scene*

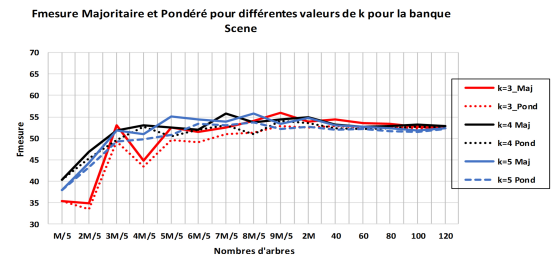


Figure 4.15 – Résultats de *Fmesure* obtenus par *RAkEL RF* pour la banque *Scene*

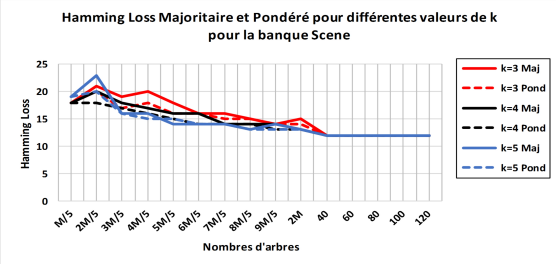


Figure 4.16 – Résultats de *Hamming Loss* obtenus par *RAkEL RF* pour la banque *Scene*

Discussions et Interprétations

Dans la figure 4.13, nous remarquons que le meilleur taux de classification obtenu est de 88% avec une stabilisation de cette valeur à partir d'un nombre d'arbres égal à 40. Pour cette banque, les différentes valeurs de k donnent presque les mêmes performances et cela peut être expliqué par le fait que les trois valeurs de k label-sets sont proches (k=3, 4, 5) et même très proches du nombre total de labels dans cette banque.

Concernant la mesure *Subset Accuracy* (voir figure 4.14), nous pouvons observer que pour des valeurs élevées de k label-sets avec un nombre réduit d'arbres, nous obtenons de meilleurs résultats. En augmentant ce nombre nous obtenons une certaine stabilité de cette mesure d'évaluation, une valeur de 40% de cette mesure indique que le classifieur *RAkEL RF* est performant, puisque cela signifie que 40% des exemples ont été bien classés par rapport à tous les labels simultanément.

Dans les figures (4.15, 4.16), nous confirmons les remarques précédentes sur l'impact de la variation du nombre de k label-sets sur les performances de classifieur. Puisque nous obtenons presque les mêmes résultats de *Fmesure* (dans la figure 4.15) pour les différentes valeurs de k même en augmentant le nombre d'arbres. Ces résultats sont très interessants, car la convergence de toutes les courbes vers une même valeur de 54% de *Fmesure* indique que le classifieur est relativement précis, puisque cette mesure c'est une moyenne harmonique entre la précision et le rappel.

Dans le cadre applicatif de la banque *Scene*, les 2 stratégies de votes testées (vote majoritaire et pondéré) donnent presque les mêmes résultats ce qui confirme l'efficacité des

classifieurs individuels, vue que la combinaison de leurs réponses nous a permis d'obtenir de bons résultats soit par vote majoritaire ou pondéré.

Finalement, pour la mesure *Hamming Loss* (voir la figure 4.15) il y a une convergence de toutes les courbes vers une valeur minimale de 12%, ce qui indique d'une part que le classifieur est stable et d'autre part une faible perte d'information.

Résultats avec la banque *Genbase*

Dans cette dernière partie nous présentons les résultats obtenus par application de l'algorithme d'ensembles de transformation *RAkEL RF* sur la banque de données *Genbase*. Les figures suivantes illustrent ces résultats :

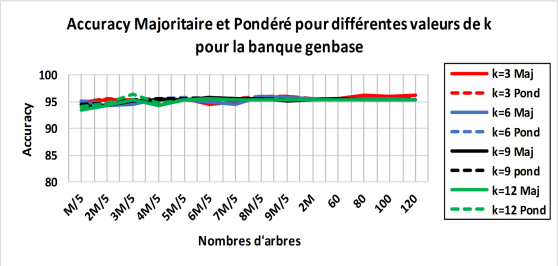


Figure 4.17 – Résultats de la mesure *Accuracy* obtenus par *RAkEL RF* pour la banque *Genbase*

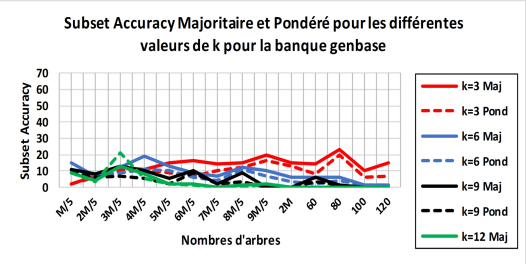


Figure 4.18 – Résultats de *Subset Accuracy* obtenus par *RAkEL RF* pour la banque *Genbase*

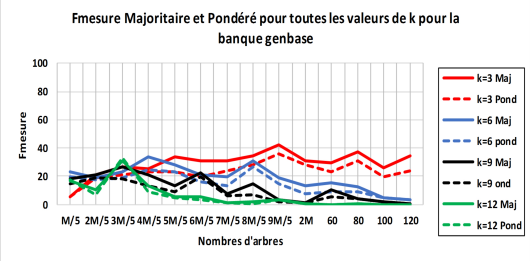


Figure 4.19 – Résultats de *Fmesure* obtenus par *RAkEL RF* pour la banque *Genbase*

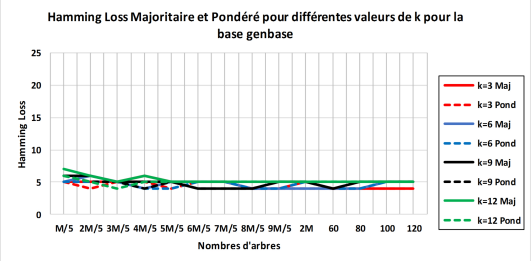


Figure 4.20 – Résultats de *Hamming Loss* obtenus par *RAkEL RF* pour la banque *Genbase*

La figure 4.17 représente les taux de classification obtenus en variant le nombre de k label-sets ainsi que le nombre d'arbres.

Nous observons que les valeurs de la mesure *Accuracy* tournent autour des 95% de bonne classification et cela pour toutes les valeurs de k label-sets avec les deux stratégies de votes, cela signifie que le classifieur est performant et stable. Nous remarquons aussi qu' avec la valeur la plus élevée de k label-sets(k=12) et à partir d'un nombre d'arbres=$5M/5$, le classifieur commence à se stabiliser. Alors que pour les autres valeurs de k, la stabilité est atteinte en utilisant un nombre élevé d'arbres. La *même* conclusion obtenue avec les autres banques de données est émise encore une fois.

En résumé, pour avoir de bonnes performances, on doit toujours garder un compromis entre le nombre d'arbres et le nombre de k label-sets, car c'est vrai que l'augmentation du nombre d'arbres améliore les performances du classifieur *même* dans le cas d'utilisation d'un nombre faible de k. Mais, trouver une valeur de k permettant d'avoir un classifieur performant avec peu d'arbres est vraiment très intéressant, puisque ça nécessite un temps de calcul minimal.

Pour les mesures *Subset Accuracy* et *Fmesure*(voir les figures (4.18, 4.19)), les meilleurs résultats sont obtenus en utilisant une valeur faible de k label-sets (k=3), mais en augmentant cette valeur les taux obtenus régressent et tendent vers une valeur nulle.

Cette régression des performances de l'algorithme *Rakel RF* peut être expliqué par le fait que nous utilisons d'une part les $Forêts$ aléatoires(RF) qui sont doublement perturbées et d'autre part, la méthode *RAkEL* qui utilise elle même de l'aléa dans la phase de la création de k label-sets.

Dans *RF* nous utilisons de l'aléatoire dans la phase du *Bagging*(voir chapitre3) ainsi que dans la phase de la création des arbres dans la $Forêt$ par le choix aléatoire de variables (le paramètre *mtry* [58]).

La banque de données *Genbase* qui est à notre disposition contient un grand nombre de variables (égale à 1185 (voir le tableau 4.1)). Dans l'algorithme *RF* la valeur mtry est obtenue en calculant la racine du nombre de variables dans la banque de données [58].
Pour la banque *Genbase*, cette valeur est très élevée impliquant une injection d'une grande perturbation, ce qui cause une dégradation des performances des prédicteurs individuels et le prédicteur agrégé est alors mauvais. Ceci explique les résultats médiocres par l'application en plus d'une valeur élevée de k label-sets (c'est à dire une autre forte perturbation).

Dans la figure 4.20, nous constatons que les résultats par rapport à la mesure *Hamming Loss* sont compétitifs, puisque nous avons obtenu pour presque toutes les différentes valeurs de k des valeurs minimales ce qui indique que le nombre de paires exemples-labels mal classées est faible.

5.4 *Bagged MLknn* Vs. *RAkEL RF*

Dans cette partie nous élaborons une comparaison entre nos deux contributions présentées dans ce chapitre qui sont *Bagged MLknn* et *RAkEL RF*.

En comparant les résultats obtenus par l'application de ces deux méthodes sur les mêmes banques de données : *Yeast*, *Scene* et *Genbase*, nous notons qu'ils sont très proches et très compétitifs mais l'algorithme d'ensemble d'adaptation *Bagged MLknn* est moins complexe que *RAkEL RF* vu qu' il utilise l'aléatoire seulement dans la phase du *Bootstrapping*. Alors que dans le 2éme algorithme nous combinons la méthode *RF* qui est déjà doublement perturbée puisqu'elle utilise de l'aléatoire dans la phase du *Bagging* et dans le choix de variables qui interviennent dans le modèle, avec la méthode *RAkEL* qui utilise à son tour de l'aléatoire dans le choix de k label-sets.
Cette complexité pose un problème seulement pour les banques volumineuses avec un grand nombre de variables, mais un bon choix de paramètres de cet algorithme permet de tirer profit de ses avantages.

Cette étude nous a permis d'aboutir aux conclusions suivantes sur les propriétés de chaque approche :

Pour *Bagged MLknn*

– La variation du nombre de k plus proches voisins n'influe pas de manière significative sur les performances du classifieur et donne presque les *mêmes* résultats pour différentes valeurs de k.
– L'utilisation des méthodes d'ensembles par la création d'un *Bagged MLknn* a donné des résultats compétitifs à ceux obtenus par k plus proches voisins multi-labels simple(*MLknn simple* [5]).
– Les résultats obtenus en appliquant le vote pondéré améliore ceux obtenus par vote majoritaire.
– La méthode *Bagged MLknn* garantit deux notions importantes **la précision** et **l'efficacité** *grâce* à l'utilisation des méthodes d'ensembles.

Pour *RAkEL RF*

– Pour avoir de bons résultats il faut toujours garder un compromis entre le nombre d'arbres et le nombre de k label-sets.
– Pour les banques volumineuses avec un nombre élevé de variables telle que *Genbase*, il vaut mieux utiliser de faibles valeurs de k label-sets avec un nombre approprié d'arbres.
– Le paramètre *Distinct* (voir tableau 4.1) joue un *rôle* majeur dans cet algorithme, il représente le nombre de label-sets différents dans la banque et il intervient beaucoup plus dans le calcul de la mesure *Subset Accuracy*. En effet, un nombre faible de ce paramètre (comme dans la banque *Scene* et *Genbase*) implique que nous avons suffisamment d'exemples d'apprentissages pour pouvoir prédire les différentes combinaisons de labels (*Subset Accruacy* élevé) mais dans le cas contraire nous aurons un nombre faible d'exemples d'apprentissages pour chaque combinaison de labels. En conséquence, la prédiction sera faible (comme par exemple dans le cas de *Yeast*, les valeurs de *Subset Accuracy* sont faibles car cette banque possède un nombre élevé de *Distinct*).

6 Conclusion

Dans ce chapitre nous avons présenté les résultats d'expérimentations de deux nouveaux algorithmes : *Bagged MLknn* et *RAkEL RF*, appliqués sur des données d'évaluation couramment utilisées dans la littérature et évalués *grâce* aux différentes mesures d'évaluation qui sont :*Accuracy*, *Subset Accuracy*, *Fmesure* et *Hamming Loss*.

Les résultats obtenus indiquent une très bonne performance des deux méthodes. Nous avons également élaboré une étude comparative entre ces deux approches, en mettant l'accent sur les paramètres qu'il faut prendre en considération et l'impact de la nature de données utilisées sur le comportement des deux algorithmes *Bagged MLknn* et *RAkEL RF*.

Conclusion générale & perspectives

Nous nous sommes intéressés dans ce projet de fin d'études au domaine de la classification *multi-labels*, afin de tirer profit de ses avantages et d'appliquer les algorithmes existants pour résoudre la problématique de poly pathologies chez les patients. Cette problématique est affirmée par les praticiens, vu que sur le terrain les patients peuvent être atteints de plusieurs pathologies simultanément.

Sur la base de cette problématique, l'idée est née de réaliser un système d'aide au diagnostic médical qui permet de faciliter cette démarche de diagnostic. Le domaine de la classification *multi-labels* est le mieux placé pour traiter la faisabilité de cette proposition.

Afin d'atteindre cet objectif, nous avons tout d'abord étudié les approches et les notions fondamentales de ce domaine, ainsi que l'état de l'art et les différentes approches proposées pour solutionner ce problème. Cela nous a permis de détecter les difficultés potentielles et les voies de développement prometteuses de ce domaine.

Dans la littérature, nous distinguons deux principales familles de méthodes pour résoudre le problème de la classification *multi-labels* ,les méthodes d'adaptation et les méthodes de transformation. Dans ce travail nous avons étudié deux algorithmes très intéressants appelés :*MLknn* (Multi Label k nearest neighbours) et *RAkEL* (Random k label-sets) qui appartiennent respectivement à ces deux familles. Nous avons également proposé deux nouvelles approches, l'une basée sur les méthodes d'ensembles appelées *Bagged MLknn*, l'autre est l'application de la méthode *RAkEL* avec les *Forêts* aléatoires. Les résultats obtenus par les deux contributions sont très prometteurs.

Ce travail de fin d'études, nous ouvre plusieurs portes et pistes pour la continuité de ce travail. De ce fait, plusieurs aspects sont à prendre en considération :

– Un premier point qu' il serait intéressant de traiter, est l'étude en détail de la problématique de la classification *multi-labels* dans le domaine médical, et l'analyse de cette problématique avec des experts du domaine afin de se repérer dans ce vaste domaine, et donc tester les algorithmes les plus appropriés parmi une panoplie de méthodes existantes.

– En parcourant l'état de l'art, nous avons compris l'*intêret* de deux notions importantes : L'exploitation de dépendances entre les labels et l'impact de la sélection de variables dans ce domaine.

Récemment, la problématique d'exploitation de dépendances localement a été proposé dans [43] par Huang & Zhou, nous nous sommes intéressés à cette vision puisque chaque patient est caractérisé par une relation différente de dépendance entre les labels (pathologies), cette relation de corrélation diffère d'un sujet à un autre donc exploiter les

dépendances localement est très bénéfique.

– Dans le passé récent, la majorité de méthodes dédiées à cette tâche ont traité ce problème globalement, c'est-à-dire en considérant pour toutes les instances d'apprentissages la même relation de corrélation entre les labels alors que ce n'est pas toujours vrai car dans les applications réelles chaque instance est caractérisée par une relation de dépendance différente.

– Une sélection de variables étant à envisager, car dans le processus du test, pour déterminer les labels d'une instance, les approches de classification *multi-labels* proposées dans la littérature se basent sur un même ensemble de variables pour tous les labels. Mais cette stratégie peut échouer dans le cas d'un grand nombre de labels qui nécessite chacun un ensemble de variables spécifiques. Sur la base de cette réflexion, Min-Ling Zhang propose dans [45], une approche qui s'appuie sur les caractéristiques spécifiques du label.

Bibliographie

[1] Pierre Henri Comble, jean louis Renauld-Salis, Philippe Lagouarde, and stephane Darmoni, "Etude des systemes d aides a la decision medicale," Tech. Rep., etude commanditee par la haute autorite de sante et realisee par Cegedim-activ, 2010.

[2] "Equipe mcale hypertension online,hypertension arterielle et situations particulieres," `http://www.hypertension-online.com/08-sitpart9.shtml`, 2008.

[3] Zatimi Nadjima, "prise en charge de l'hypertension arterielle du diabetique," medecine interne EPH Boufarik,Algerie,`http://larevuemedicale-dz.com/prise-en-charge-de-lhypertension-arterielle-du-diabetique`.

[4] Min-Ling Zhang, Zhi-Hua Zhou, and Nanjing University, "A review on multi-label learning algorithms," *IEEE Computer Society*, 2013.

[5] Min-Ling Zhang and Zhi-Hua Zhou, "Ml-knn : A lazy learning approach to multi-label learning," *PatternRecognition*, vol. 40(7), pp. 2038–2048, 2007.

[6] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multilabel classification of music into emotions," *Proc. 9th International Conference on Music Information Retrieval*, pp. 325–330, 2008.

[7] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas, "Mining multi-label data," *In Data Mining and Knowledge Discovery Handbook*, pp. 667–685, 2010.

[8] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Saso Dzeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45(9), pp. 3084–3104, 2012.

[9] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms," *Proceedings of the 5th Hellenic conference on Artificial Intelligence : Theories, Models and Applications*, pp. 401–406, 2008.

[10] A. Clare and R.D. King, "Knowledge discovery in multi-label phenotype data," *Proceedings of the 5th European Conference on PKDD*, pp. 42–53, 2001.

[11] K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *Journal of Machine Learning Research 3*, pp. 1025–1058, 2003.

[12] M.L. Zhang and Z.H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18(10), pp. 1338–1351, 2006.

[13] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval*, pp. 274–281, 2005.

[14] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23(7), pp. 1079–1089, 2011.

[15] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank, "Classifier chains for multi-label classification," *Machine Learning and Knowledge Discovery in Databases ,Lecture Notes in Computer Science*, vol. 5782, pp. pp 254–269, 2009.

[16] G. Tsoumakas and I. Katakis, "Multi label classification : an overview," *International Journal of Data Warehouse and Mining*, vol. 3 (3), pp. 1–13, 2007.

[17] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," *Proceedings of the 8th IEEE International Conference on Data Mining*, pp. 995–1000, 2008.

[18] G. Tsoumakas and I. Vlahavas, "Random k-labelsets : an ensemble method for multilabel classification," *Proceedings of the 18th European conference on Machine Learning*, pp. 406–417, 2007.

[19] J. Read, "A pruned problem transformation method for multi-label classification," *Proceedings of the New Zealand Computer Science Research Student Conference*, pp. pp. 143–150, 2008.

[20] E. Hullermeier, J. Furnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artificial Intelligence*, vol. 172(16), pp. 1897–1916, 2008.

[21] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24(2), pp. 123–140, 1996.

[22] D. Kocev and J. Stefan, *Ensembles for predicting structured outputs*, Ph.D. thesis, Ljubljana,Slovenia, 2011.

[23] Shi Chuan, Kong Xiangnan, Yu Philip S, and Wang Bai, "Multi-label ensemble learning„" *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science*, vol. 6913, pp. 223–239, 2011.

[24] Mohammad S Sorower, "A literature survey on algorithms for multi-label learning," Tech. Rep., 2010.

[25] Domains Araken M Santos, Anne M P Canuto, and Antonino Feitosa Neto, "A comparative analysis of classification methods to multi-label tasks in different application," in *10th International Conference on Hybrid Intelligent Systems (HIS 2010), Atlanta, GA, USA*, 2010.

[26] G. Malik and M. Tarique, "On machine learning techniques for multi-class classification," *International Journal of Advancements in Research Technology*, vol. 3(2), 2014.

[27] Z. Younes, F. Abdallah, T. Denoeux, and H. Snoussi, "A dependent multi-label classification method derived from the k-nearest neighbor rule," *EURASIP Journal on Advances in Signal Processing*, p. 14, 2011.

[28] Min-Ling Zhang, José M. Peña, and Victor Robles, "Feature selection for multi-label naive bayes classification," *Information Sciences*, vol. 179, pp. 3218–3229, 2009.

[29] Everton Alvares-Cherman, Jean Metz, and Maria Carolina Monard, "Incorporating label dependency into the binary relevance framework for multi-label classification," *Expert Systems with Applications : An International Journal*, vol. 39(2), pp. 1647–1655, 2012.

[30] Sawsan Kanj, Fahed Abdallah, and Thierry Deoeux, "Evidential multi-label classification using the random k-label sets approach," *Proceedings of the 2nd Int. Conf. on Belief Functions.*, pp. 21–28, 2012.

[31] Hung Yi L, Shou De Lin , and Hsin Min Wang, "Generalized k-labelset ensemble for multi-label classification," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan*, pp. 2061–2064, 2012.

[32] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37(3), 1999.

[33] Maxime Gasse, Alex Aussem, and Haytham Elghazel, "A hybrid algorithm for bayesian network structure learning with application to multi-label learning," *Expert Systems with Applications*, vol. 41(15), pp. 6755–6772, 2014.

[34] Eneldo Loza Mencia, Sang-Hyeun Park, and Johannes Furnkranz, "Efficient voting prediction for pairwise multilabel classification," *Neurocomputing*, vol. 73(7-9), pp. 1164–1176, 2010.

[35] Johannes Furnkranz, Eyke Hullermeier, Eneldo Loza Mencia, and Klaus Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73(2), pp. 133–153, 2008.

[36] S.H.Park and J.Furnkranz, "Efficient pairwise classification," *Proceedings of 18th European conference on Machine Learning (ECML07),Warsaw,Poland,Springer,Berlin*, pp. 658–665, 2007.

[37] Gjorgji Madjarov, Dejan Gjorgjevikj, and Saso Dzeroski, "Two stage architecture for multi-label learning," *Pattern Recognition*, vol. 45(3), pp. 1019–1034, 2012.

[38] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," *Proc. ECML/PKDD Workshop on Mining Multidimensional Data (MMD08)61*, 2008.

[39] A. Banerjee and J. Ghosh, "Scalable clustering algorithms with balancing constraints," *Data Mining and Knowledge Discovery*, vol. 13, pp. 365–395, 2006.

[40] Kocev Dragi, Vens Celine, Struyf Jan, and Dzeroski Saso, "Tree ensembles for predicting structured outputs," *Pattern recognition*, vol. 46(3), pp. 817–833, 2013.

[41] Elena Montanes, Jose Ramon, Quevedoand Juan, and Jose del Coz, "Aggregating independent and dependent models to learn multi-label classifiers," *Machine Learning and Knowledge Discovery in Databases ,Lecture Notes in Computer Science*, vol. 6912, pp. 484–500, 2011.

[42] Muhammad Atif Tahir, Josef Kittler, and Ahmed Bouridane, "Multilabel classification using heterogeneous ensemble of multi-label classifiers," *Pattern Recognition Letters*, vol. 33(5), pp. 513–523, 2012.

[43] Sheng-Jun Huang and Zhi-Hua Zhou, "Multi-label learning by exploiting label correlations locally," *Expert Systems with Applications : An International Journal archive*, vol. 41(6), pp. 2989–3004, 2014.

[44] K. Dembczynski, W. Waegeman, W. Cheng, and E Hullermeier, "On label dependence and loss minimization in multi-label classification," *Machine Learning*, vol. 88(1-2), pp. 5–45, 2012.

[45] Min-Ling Zhang, "Lift : Multi-label learning with label-specific features," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia,Spain*, 2011.

[46] Q. Gu, Z. Li, and Han, "Correlated multi-label feature selection," *CIKM*, pp. 1087–1096, 2011.

[47] N. Spolaor, E.A. Cherman, M.C. Monard, and H.D Lee, "A comparison of multilabel feature selection methods using the problem transformation approach," *Electr. Notes Theor. Comput. Sci*, vol. 292, pp. 135–151, 2013.

[48] I. Guyon and Elisseeff, "An introduction to variable and feature selection.," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[49] Ouadie Gharroudi, Haytham Elghazel, and Alexandre Aussem, "A comparison of multi-label feature selection methods using the random forest paradigm," in *Canadian Conference on Artificial Intelligence, AI*, 2014.

[50] T. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Systems*, 2000.

[51] Guillaume Lecue, *Methodes d'agregation : optimalite et vitesses rapides*, Ph.D. thesis, Universite Paris VI, 2007.

[52] Robin Genuer, *Forets aleatoires : aspects theoriques, selection de variables et applications*, Ph.D. thesis, Universite Paris-Sud11, 2010.

[53] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, New York, NY, 1993.

[54] Caron Stephane, "Une introduction aux arbres de decision http ://scaron.info," 2011.

[55] Bernard Simon, "Forets aleatoires : De l'analyse des mecanismes de fonctionnement a la construction dynamique 2009," `https://tel.archives-ouvertes.fr/tel-00598441`.

[56] Grigorios Tsoumakas and Ioannis Vlahavas, "Random k-labelsets an ensemble method for multilabel classification machine learning," *ECML ,Lecture Notes in Computer Science*, vol. 4701, pp. 406–417, 2007.

[57] Jesse Read, "Multi-label classification, tutorial, department of signal theory and communications madrid, spain 2013," .

[58] Leo Breiman, "Random forests," *Machine Learning*, vol. 45(1), pp. 5–32, 2001.

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

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

[61] Sebastien Gadat, "Laboratoire de statistique et probabilites, umr 5583 cnrs-ups," `www.lsp.ups-tlse.fr/gadat`.

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

[63] G. Biau, F. Cerou, and Guyader, "On the rate of convergence of the bagged nearest neighbor estimate," *Journal of Machine Learning Research*, vol. 11, pp. 687–712, 2010.

[64] Alexis Lechervy, "Fouille de donnees et apprentissage," `https://lechervy.users.greyc.fr/cours/appM1/methodeensembliste.pdf`, 2014.

[65] Grigorios Tsoumakas, Eleftherios, Spyromitros Xioufis, and Jozef Vilcek, "Mulan :a java library for multi-label learning," `http://mulan.sourceforge.net/datasets-mlc.html`.

[66] A. Elisseef and J. Weston, "A kernel method for multi-labelled classification," *T. G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, Cambridge*, pp. 681–687, 2002.

[67] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, "Combining microarray expression data and phylogenetic proteines to learn functional categories using support vector machines," *Proceedings of the 5th Annual International Conference on Computational Biology, Montreal, Canada*, pp. 242–248, 2001.

[68] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, pp. 1757–1771, 2004.

[69] S. Diplaris, G. Tsoumakas, P. Mitkas, and I Vlahavas, "Protein classification with multiple algorithms," *Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005) Greece*, pp. 448–456, 2005.

[70] Min-Ling Zhang, "Learning from multi-label data, tutoriel,school of computer science & engineering,southeast university,china 2009," `http://cse.seu.edu.cn/people/zhangml`.

Annexe : Interface Graphique

L'application réalisé permet de tester les algorithmes proposés dans notre Projet de fin d'études de Master IBM. Il s'agit des deux algorithmes appelé : Bagged MLknn et Rakel Random Forest.

– Bagged MLknn est une méthode d'ensemble d'adaptation.

– Rakel RF est une méthode d'ensemble de transformation qui utilise un type de transformation du problème multi-labels en un problème multi-classes.

FIGURE 1 – *Fenêtre* principale

La *fenêtre* **Information sur l'application** : Cette fenêtre donne une brève description sur l'application. Comme montré ci-dessous dans la figure(2)

FIGURE 2 – *Fenêtre* informations sur l'application

La *fenêtre* Chargement de la banque de données

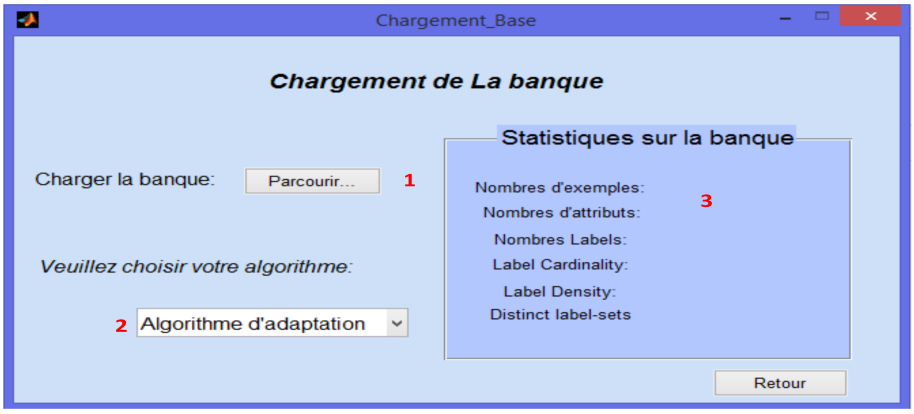


FIGURE 3 – *Fenêtre* chargement de la banque

1. C'est pour choisir la banque de données pour le test.
2. Pour choisir le type d'algorithme à appliquer.
3. Des champs pour afficher les informations sur la banque de données chargée.

Dans la partie qui suit, nous présentons un exemple d'application avec la banque de données *Genbase* :

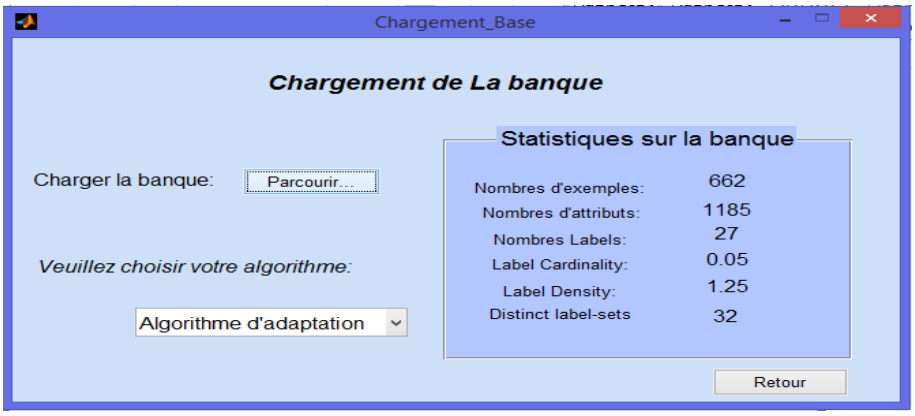


FIGURE 4 – *Fenêtre* chargement de la banque *Genbase*

Partie 1 : Choix Algorithme d'Adaptation :

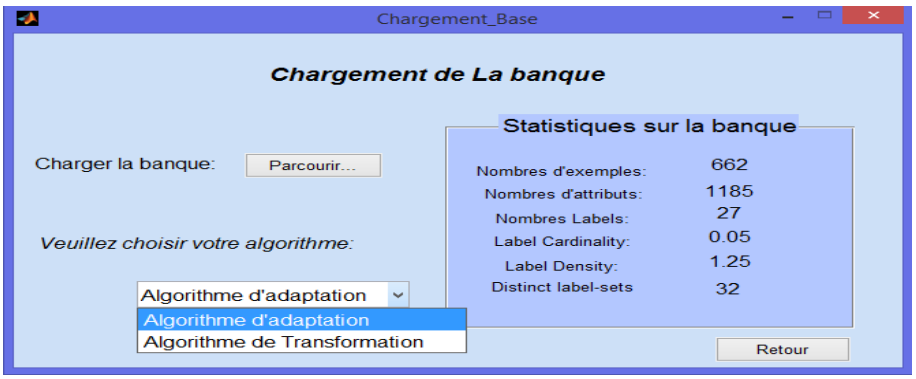


FIGURE 5 – *Fenêtre* choix algorithme d'adaptation

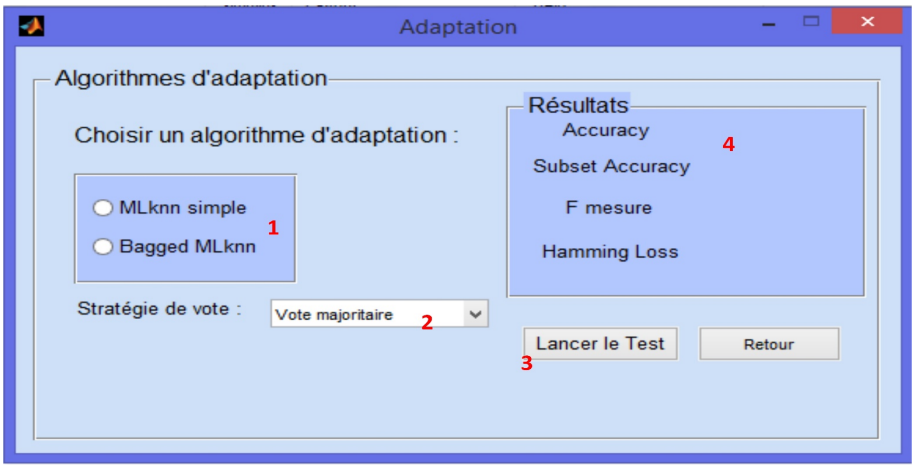


FIGURE 6 – *Fenêtre* Algorihmes d'adaptation

1. Permet de choisir le type d'algorithme d'adaptation (MLknn simple ou Bagged Mlknn).
2. Pour choisir la stratégie du vote.
3. Lancer le test.
4. Permet d'afficher les résultats du test

Dans le cas du **choix de MLknn simple**(voir les figures 7,8) :

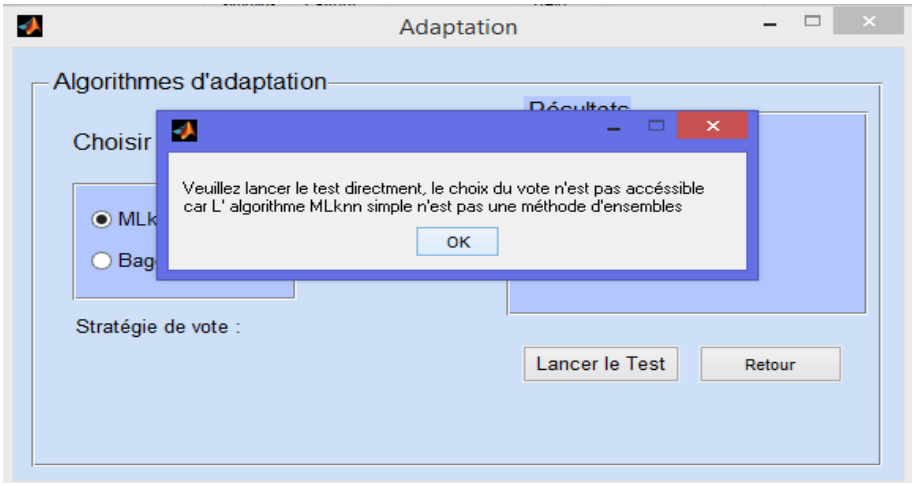


FIGURE 7 – *Fenêtre* Algorithme d'adaptation(MLknn simple)

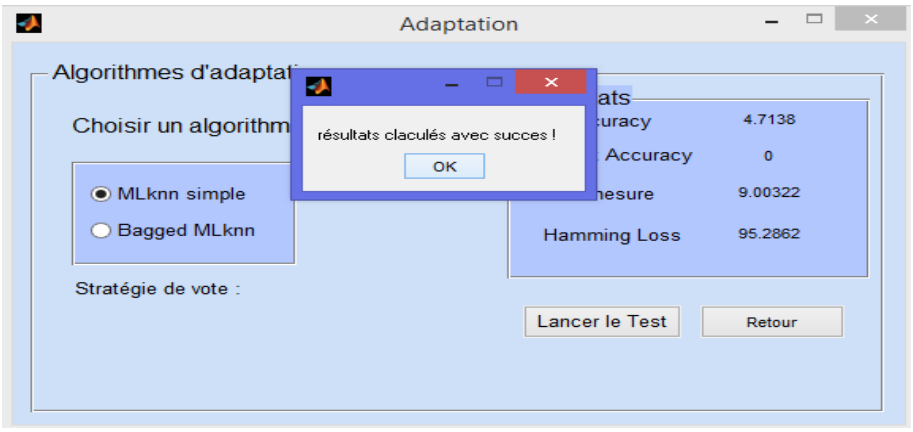


FIGURE 8 – *Fenêtre* Résultats de l'algorithme d'adaptation (MLknn simple) sur Genbase

Dans le cas du **choix de Bagged MLknn avec vote majoritaire** :

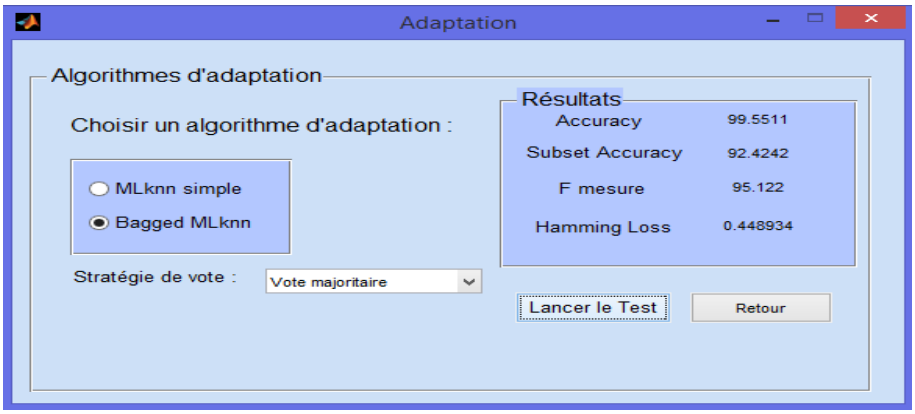


FIGURE 9 – *Fenêtre* Résultats de l'algorithme d'adaptation (Bagged MLknn Majoritaire) sur Genbase

Dans le cas du **choix de Bagged MLknn avec vote Pondéré** :

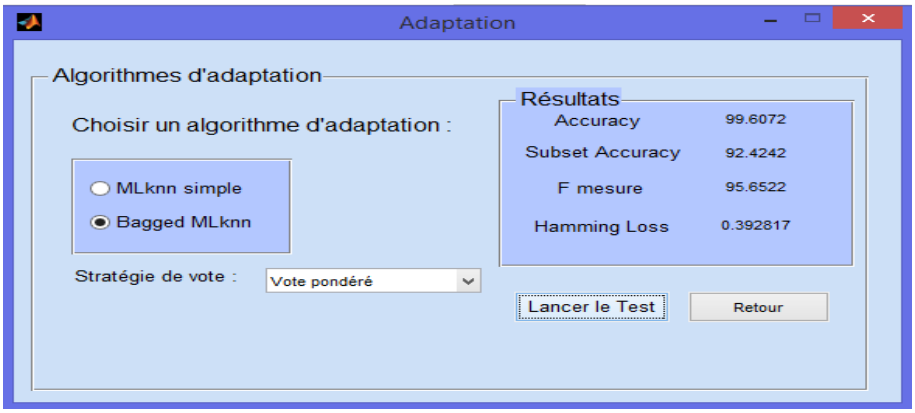


FIGURE 10 – *Fenêtre* Résultats de l'algorithme d'adaptation (Bagged MLknn Pondéré) sur Genbase

Partie 2 : Tester l'algorithme de transformation sur la banque de données *Genbase*

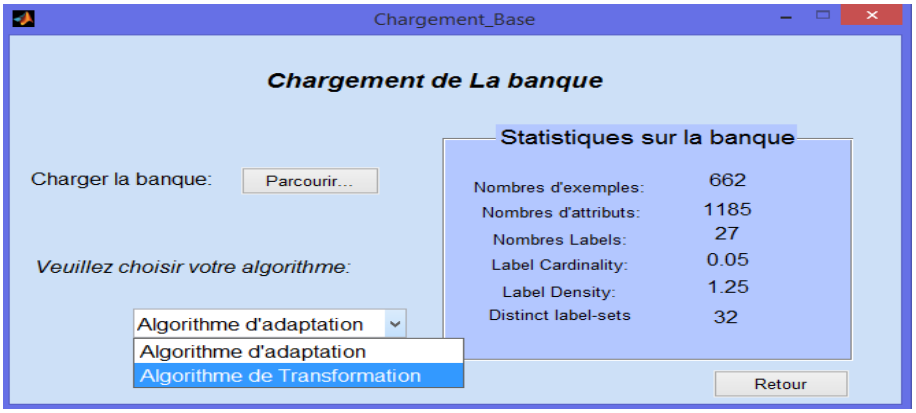


FIGURE 11 – *Fenêtre* choix algorihme de transformation

Dans le cas sur **choix de Rakel Random Forest avec vote majoritaire** :

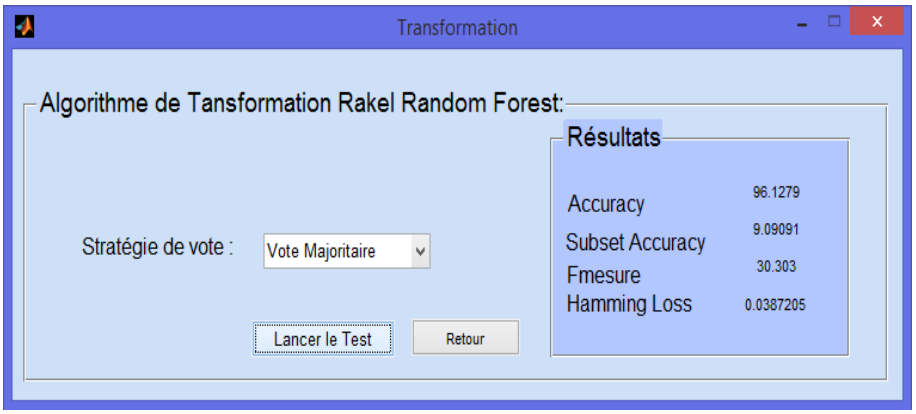


FIGURE 12 – *Fenêtre* Résultats de l'algorithme de transformation(RAkEL RF Majoritaire) sur Genbase

Dans le cas sur **choix de Rakel Random Forest avec vote Pondéré** :

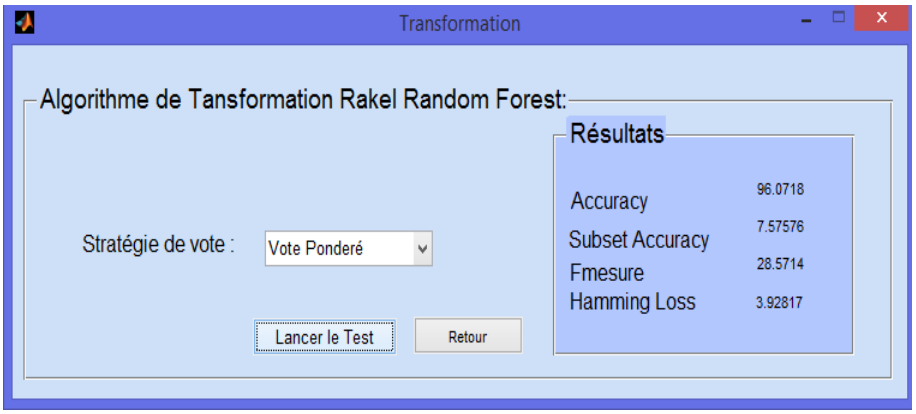


FIGURE 13 – *Fenêtre* Résultats de l'algorithme de transformation(RAkEL RF Pondéré) sur Genbase