l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Noureddine Yassine
NAIR BENREKIA

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans

École doctorale : ED 503 (STIM)

Discipline : Informatique et applications, section CNU 27
Unité de recherche : Laboratoire d'informatique de Nantes-Atlantique (LINA)

Soutenue le 3 novembre 2015

Classification interactive multi-label pour l'aide à l'organisation personnalisée des données

JURY

Rapporteurs :	**M. Younès BENNANI**, Professeur des universités, Université de Paris 13
	M^me^ Christel VRAIN, Professeur des universités, Université d'Orléans
Examinateur :	**M. Philippe PREUX**, Professeur des universités, Université de Lille 3
Directrice de thèse :	**M^me^ Pascale KUNTZ**, Professeur des universités, Université de Nantes
Co-directeur de thèse :	**M. Frank MEYER**, Ingénieur de recherche, Orange Labs

Remerciements

« The whole point of being alive is to evolve into the complete person you were intended to be »

-Oprah Winfrey

Ce manuscrit conclut trois belles années de thèse et si c'était à refaire, je n'aurai rien changé et je n'aurai pas fait mieux. Je sais que j'ai tout donné jusqu'aux dernières minutes avant la soutenance. Je voulais vraiment réussir et mon vœu s'est bien réalisé.

Bien que j'avais une très forte motivation et beaucoup de courage mais sincèrement je pense que -sans la présence d'un certain nombre de personnes- cette vie aurait été beaucoup moins belle. Je tiens donc à les remercier -un par un- pour leur faire part de ma reconnaissance.

Premièrement, je tiens à remercier chaleureusement ma directrice de thèse Pr. Pascale KUNTZ et mon encadrant industriel Frank MEYER pour la confiance qu'ils m'ont témoignée tout au long de la thèse. Je les remercie également de m'avoir accordé la liberté dans le choix de mes directions de recherche et de m'avoir permis de faire ce que j'aime faire. Ils n'ont jamais douté de mon potentiel et c'est ce qui m'a donné de la force pour arriver jusqu'au bout. Leur bienveillance et leur rigueur scientifique m'ont permis de m'améliorer en tant que chercheur et avant tout en tant que personne. Merci Pascale pour tout le temps que tu m'as consacré malgré tes nombreuses occupations et malgré les centaines kilomètres qui nous séparent ; la notion de distance n'existait pas pour nous ! C'était toujours un grand plaisir pour moi de venir te rendre visite à Nantes et de discuter avec toi de mes nouveaux résultats. Merci Frank de m'avoir premièrement accepté pour ce beau sujet de thèse ; c'est de là où tout à commencer. Merci de m'avoir accompagné avec plein d'encouragements chaque jour de cette thèse jusqu'aux dernières minutes au café avant la soutenance ; j'en garde vraiment un très beau souvenir. Et quel plaisir pour moi que vous ayez fait la connaissance de mes parents ...

Puis, je remercie Pr. Younes BENNANI et Pr. Christel VRAIN de m'avoir fait l'honneur d'accepter d'être rapporteurs de ma thèse. Je les remercie pour la rapidité avec laquelle ils ont lu mon manuscrit, malgré un emploi du temps certainement chargé et je les remercie pour leurs beaux compliments et leurs critiques toutes aussi constructives. Je remercie également Pr. Philippe PREUX qui m'a fait l'honneur de présider ce jury lors de ma soutenance.

Ensuite, je voudrais remercier mon chef d'équipe à Orange Labs Fabrice CLEROT pour sa grande attention et sa disponibilité. Sa passion pour la recherche et sa rigueur scientifique exceptionnelles m'ont beaucoup inspiré et continueront toujours à m'inspirer. Nos nombreuses belles discussions et ses précieux conseils –depuis le début jusqu'à la fin de la thèse- m'ont permis d'apprendre beaucoup de choses dans plusieurs domaines et sur plusieurs plans et m'ont surtout permis de croire en mes idées et en mes hypothèses. Quel honneur pour moi !

J'ai eu la chance de travailler au sein de deux belles équipes de recherches. Je remercie d'abord tous les membres de mon équipe « Profiling & Datamining » au sein d'Orange Labs d'avoir -de près ou de loin- contribué à l'aboutissement de ma thèse. Je garde de beaux souvenirs de mon séjour parmi vous. Dans l'ordre alphabétique : Barbara, Bruno, Carine, Christine, Fabien, Françoise, Marc, Marie-Noël, Nicolas, Romain, les trois Sylvies, Tanguy, Raphael, Vincent avec un remerciement spécial pour Christine, Tanguy et Vincent. Tanguy, merci beaucoup pour tout le temps que tu m'as consacré pour m'expliquer des concepts ou pour écouter mes questions. Toujours souriant et prêt à partager ton savoir et ta passion avec les autres. Vincent, merci et milles merci d'avoir été là ! Tu as joué un grand rôle pendant la phase finale de ma thèse. Tes conseils et ton implication m'ont été plus que précieux. Christine, un grand merci pour ta sympathie, ta gentillesse et ton sourire ; tu n'as pas vraiment été une collègue mais une amie.

Mes remerciements vont ensuite à toute la troupe des (post-) doctorants : Elias, Oumaima, Pratik, Robin, Tarek et Tatiana et à tous ceux qui sont partis : Coralie, Dominique, Duc, Pascal et Sami avec un remerciement special pour papi Pascal avec qui j'ai eu le plaisir de partager plein de belles choses pendant mes deux premières années de thèse. Je te remercie infiniment pour ton aide, ton écoute, ton attention et ton amitié. J'ai eu un grand plaisir de chanter pendant la soirée de ta retraite ; j'en garde un très beau souvenir.

Je n'ai malheureusement pas eu l'occasion de beaucoup interagir avec les membres de mon

équipe « COnnaissances et Décisions » à l'école polytechnique de l'université de Nantes. Cependant, je tiens à remercier quelques personnes pour leur gentillesse et leur accueil toujours sympathique. Dans l'ordre alphabétique : Mounira, Philippe, Rajani et Toader, et les doctorants : Amanullah, Anthony, Lambert et Nicolas.

Je tiens aussi à remercier Nicolas LABROCHE et Mustapha LEBBAH d'avoir accepté de suivre mes travaux de thèse pendant les deux premières années. Vos retours m'ont encouragé et m'ont surtout permis de progresser. Merci encore à Nicolas d'avoir pris du temps pour échanger avec moi à EGC 2014 et d'être venu écouter ma présentation.

Et enfin, je voudrais remercier Jesse Read, un chercheur espagnol que je n'ai jamais rencontré mais avec qui j'ai pu échanger à distance via un échange de messages. Très sympathique et toujours réactif à mes nombreuses questions.

Et pour finir, je voudrais envoyer un « merci » plein d'émotions à mes chers parents (Amina et Kader), mes sœurs (Ikram et Rachida) et ma mami (Zahia) qui m'ont soutenu tous les jours de ces cinq dernières années. Vous êtes ma joie ; vous êtes mon bonheur ; vous êtes l'air que je respire. Ce succès est le vôtre ! Je vous le dédie et croyez-moi ce jour –ou tout redeviendra comme avant- viendra dans le futur. Pour le moment, on sourit et on espère le meilleur pour ce qui nous attend. Je remercie aussi : Amine, Sabah et Fouad pour leur soutien et leur encouragement inconditionnels.

Non, ce n'est pas encore fini ! Ces dernières lignes vont naturellement à ma famille lannionaise avec qui j'ai partagé ma joie, mes rires, ma folie, ma musique mais aussi mes doutes et ma peine tous les jours pendant ces trois dernières années. Oh que les souvenirs me hantent . . . ! J'envoie aussi un beau « merci » à tous mes meilleurs amis en Algérie. On ne s'est jamais séparé depuis que je suis parti et on continue à s'aimer de plus en plus tous les jours. Je vous aime tout affectueusement et je vous laisse vous identifier tous seuls.

Table des matières

Table des figures

Liste des tableaux

Notations mathématiques

Notation	Définition
$\mathcal{F}$	espace d'attributs
$\mathcal{L}$	espace de labels
$\mathcal{D}$	jeu de données multi-label
$\mathcal{T}$	ensemble d'apprentissage
$\mathcal{S}$	ensemble d'évaluation ou ensemble de test
m	nombre d'attributs
q	nombre de labels
n	nombre d'exemples d'apprentissage
λ_i	label i
f_i	attribut i
x_i	exemple d'apprentissage i
y_i	ensemble des labels de l'exemple d'apprentissage x_i
y_i^j	label j de l'exemple d'apprentissage x_i
y_i^+	ensemble des labels positifs de l'exemple d'apprentissage x_i
y_i^-	ensemble des labels négatifs de l'exemple d'apprentissage x_i
$h(.)$	classifieur multi-label
$\hat{y}_i$	ensemble des labels prédits pour l'exemple de test x_i
$\hat{y}_i^j$	prédiction du modèle pour le label j de l'exemple de test x_i

TABLE 1 – Résumé des notations mathématiques majeures.

1

Introduction

Sommaire

Avec la démultiplication des volumes de données popularisée par le phénomène du « Big Data », de nombreux efforts se concentrent aujourd'hui sur la capacité des algorithmes d'apprentissage à tirer profit des nouvelles données disponibles pour tenter d'adapter les résultats fournis aux besoins et profils des utilisateurs. Dans des secteurs variés, du marketing à la médecine du futur, la personnalisation des résultats est au centre des préoccupations. De nombreux algorithmes s'appuient sur des traces numériques d'usages recueillies automatiquement sur les différents supports d'activités. Les préférences des utilisateurs se déduisent alors de leurs comportements sans explicitation préalable de leur part. Un exemple d'application paradigmatique est le système de recommandation d'Amazon (Linden et al., 2003). La puissance de ces approches réside dans leur aptitude à passer à l'échelle dans le traitement des informations qui ne cessent de croître en volume et en complexité. La qualité des résultats fournis est variable en fonction des préférences « réelles » des utilisateurs et, selon les domaines d'activité, cette

variabilité peut les détourner des nouveaux services proposés.

La prise en compte des préférences peut également se faire à une échelle individuelle en intégrant l'utilisateur dans le processus d'apprentissage. Lorsque les préférences sont préalablement explicitées, elles peuvent être injectées dans le modèle, par exemple sous la forme d'un ensemble de contraintes (Bilenko et al., 2004; Wagstaff et al., 2001). Les difficultés de la formalisation de ces contraintes renvoient à des questions bien connues depuis les systèmes experts en intelligence artificielle. Une alternative consiste à apprendre « au fur et à mesure » en permettant à l'utilisateur d'interagir avec le système qui tente d'apprendre interactivement les attributs de son comportement intéressants pour le problème traité. Comme l'ont défini très récemment Amershi et al. (2015) « interactive machine learning is a process that involves a tight interaction loop between a human and a machine learner, where the learner iteratively takes input from the human, promptly incorporates that input, and then provides the human with output impacted by the results of the iteration ».

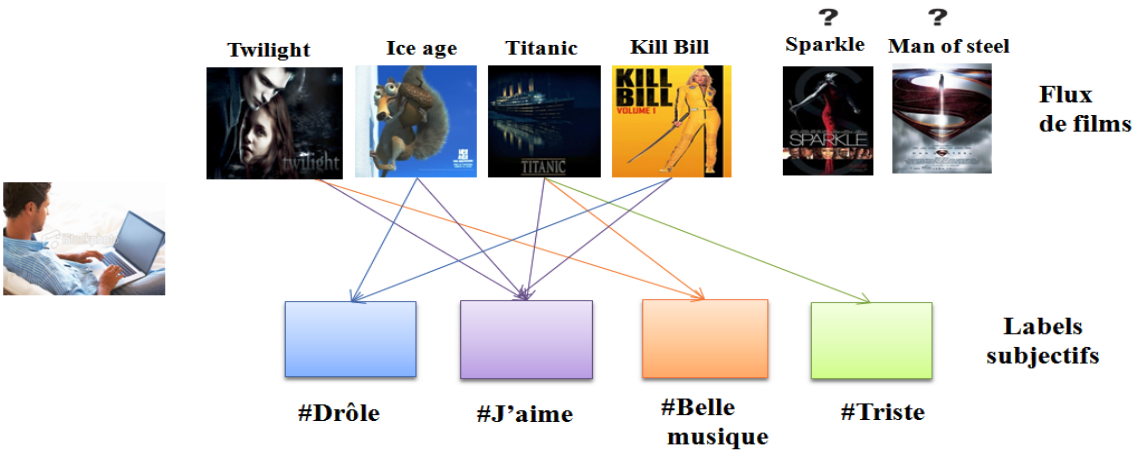


FIGURE 1 – Classification interactive multi-label du catalogue jouet décrit dans la Table 1

1 Personnalisation et classification interactive multi-label

C'est dans ce contexte de l'apprentissage interactif que se positionne cette thèse qui a été initiée à l'origine par une problématique de recommandation de Vidéo à la Demande (*VoD*). Dans ce cadre, l'objectif est de proposer à l'utilisateur des films d'un catalogue qui correspondent « le mieux » à ses préférences. Ses préférences sont apprises initialement à partir de ses avis recueillis sur un ensemble restreint d'exemples (ici de films) (Figure 1, Tableau 1). Il s'agit alors

de développer une approche qui permette d'apprendre un modèle prédictif pour lui proposer de nouveaux films potentiellement intéressants. La qualité du modèle doit alors se renforcer dans une boucle de rétroaction avec les actions de l'utilisateur.

Film	**Année**	**Acteur$_1$**		**Drôle**	**J'aime**	**Belle musique**	**Triste**
Twilight	2008	Pattinson		0	1	1	0
Ice age	2002	Romano		1	1	0	0
Titanic	1997	DiCaprio		0	1	1	1
Kill Bill	2003	Thurman		1	1	0	0
Sparkle	2013	Sparks		x	x	x	x
Man of Steel	2013	Cavill		x	x	x	x

TABLE 1 – Extrait de données liées à un catalogue de *VoD*.

Cette problématique dépasse le cadre de la *VoD* et l'importance donnée à la personnalisation de contenus a conduit au développement de plusieurs systèmes de classification interactive pour différentes applications réelles (e.g. classification d'images (Fogarty et al., 2008), sélection de fichiers (Ritter and Basu, 2009), classification de gestes (Fiebrink et al., 2009), classification de documents (Drucker et al., 2011), tri d'alarmes (Amershi et al., 2011) et création de groupes d'utilisateurs dans les réseaux sociaux (Amershi et al., 2012)). Les validations expérimentales publiées, bien que souvent partielles, sont très encourageantes et ont motivé le sujet de thèse. Cependant, toutes ces approches recourent à une classification mono-label[1] des items qui peut être très contraignante dans un contexte applicatif réel car elle limite fortement l'expressivité de l'utilisateur lorsqu'il interagit avec des données qui sont intrinsèquement multi-label. À Orange Labs, entreprise dans laquelle s'est déroulée cette thèse, l'objectif final -qui dépasse le cadre de notre travail- est de concevoir un système de classification interactive multi-label qui permette aux utilisateurs d'annoter les exemples avec plusieurs labels subjectifs et d'exprimer par conséquent des requêtes de recherche plus complexes sur les données. L'efficacité d'un tel système d'apprentissage interactif dépend de différents facteurs, notamment de la qualité de l'algorithme d'apprentissage, du support de visualisation, et des mécanismes d'interaction. Ces facteurs relèvent de directions de recherche différentes et dans cette thèse l'attention a porté plus particulièrement sur le choix de l'algorithme d'apprentissage. Par exemple, dans le cadre de la *VoD*,

[1]Dans la suite, nous utilisons l'anglicisme « label » qui est complètement banalisé dans la communauté scientifique.

il s'agit de permettre à l'algorithme d'apprentissage de prendre en compte plusieurs émotions complémentaires de l'utilisateur sur un film (ex : J'aime, Belle musique, Triste).

L'apprentissage multi-label a reçu une attention d'importance croissante cette dernière décennie stimulée initialement par des applications en catégorisation textuelle (Schapire and Singer, 2000) qui se sont étendues à des problématiques variées : classification de contenus multimedia tels ques les images (Boutell et al., 2004), audio (Lo et al., 2011), vidéos (Snoek et al., 2006), fouille de web et de règles (Ozonat and Young, 2009; Rak et al., 2005), recommandation de tags (Katakis et al., 2008), recherche d'information (Yu et al., 2005), bio-informatique (Clare and King, 2001). Différentes approches (Sorower, 2010; Tsoumakas and Katakis, 2007; Tsoumakas et al., 2010; Zhang and Zhou, 2013) ont été proposées et des expérimentations approfondies récentes ont permis de déceler l'efficacité de quelques algorithmes (Madjarov et al., 2012). Cependant, les évaluations de performances de ces algorithmes n'intègrent pas les contraintes d'interactivité.

En effet, dans un cadre interactif, le Graal est l'apprentissage avec peu d'exemples en un temps limité. Nous avons donc dans un premier temps comparé les comportements des principales approches de classification multi-label dans un cadre simulé d'interactivité. Les expérimentations ont mis en évidence le potentiel d'une approche de la famille des méthodes ensemble (Random Forest of Predictive Clustering Trees RF-PCT (Kocev, 2011; Kocev et al., 2007) qui entraîne un ensemble d'arbres de décisions multi-label peu corrélés et de faibles performances. Cette approche permet un apprentissage de qualité avec un nombre réduit d'exemples et une progression significative lorsque l'ensemble d'apprentissage croît « un peu ». Sa rapidité la rend compétitive avec les approches rapides qui sont de moindre qualité. Cependant, la forte contrainte de complexité algorithmique posée par l'interactivité nous a conduite à proposer une nouvelle approche d'apprentissage hybride FMDI-RF$^+$ (Factorisation de Matrice Duale Indépendante combinée avec le classifieur RF-PCT) qui associe RF-PCT avec une approche de factorisation de matrice efficace pour la réduction de dimensions. Les résultats expérimentaux indiquent que FMDI-RF$^+$ est aussi précise que RF-PCT dans les prédictions avec clairement un avantage à FMDI-RF$^+$ pour la vitesse de calcul. Notons que dans la majeure partie de la thèse nous avons considéré un protocole d'interaction simplifié qui nous a permis de comparer différentes approches. Un transfert à une application réelle est en cours chez Orange Labs. Elle est

présentée synthétiquement en conclusion mais sa validation est pour l'instant limitée à quelques retours d'usage.

2 Contenu du manuscrit

Ce manuscrit est structuré en quatre chapitres principaux.

1. Le chapitre 2 « **Classification interactive** » introduit les spécificités du problème de classification interactive et présente des systèmes interactifs récents qui ont été développés dans différents cadres applicatifs. Nous avons retenu ici sept systèmes qui couvrent respectivement la classification d'images, la sélection de fichiers, la reconnaissance d'écriture manuscrite, la classification de mouvements, la classification d'alarmes, la classification de documents et la classification de profils dans des réseaux sociaux. Pour chacun d'eux, nous rappelons son objectif et décrivons brièvement l'approche de classification utilisée. Puis, nous concluons par une discussion sur les méthodes d'évaluation proposées pour qualifier la qualité de ces différents systèmes.

2. Le chapitre 3 « **Apprentissage multi-label** » est un état de l'art des méthodes d'apprentissage multi-label proposées dans la littérature. Après une formalisation du problème, qui nous permet d'introduire les notations utilisées dans la suite, notre présentation s'appuie sur la typologie en trois grandes familles utilisée récemment par Madjarov et al. (2012) : les méthodes d'apprentissage par transformation qui réduisent le problème initial en problèmes mono-label, les méthodes d'apprentissage par adaptation qui intègrent les spécificités du cadre multi-label dans des approches d'apprentissage classiques, et les méthodes d'apprentissage ensemble qui combinent des classifieurs. Par ailleurs, nous soulignons leurs avantages et inconvénients et précisons leurs complexités théoriques qui nous donnent une première indication sur leurs capacités à supporter la contrainte de temps de calcul posée par l'interactivité. Puis, nous présentons les différents critères et jeux de données d'évaluation qui sont utilisés dans la littérature et sur lesquels nous nous appuyons pour nos propres travaux. Ces critères peuvent se classer en deux grandes familles : les critères qui portent sur l'évaluation de la classification –à la fois par exemple

et par label- et les critères qui portent sur le ranking. Le chapitre se termine par une discussion sur les résultats expérimentaux des études comparatives récentes.

3. Le chapitre 4 « **Apprentissage multi-label avec contraintes d'interactivité** » présente l'analyse comparative que nous avons effectuée avec les approches décrites au chapitre 3 en les soumettant aux contraintes d'interactivité. Nous avons considéré deux aspects : leurs performances prédictives avec un nombre d'exemples croissant qui reste réduit, et leurs temps de calcul. L'analyse des critères d'évaluation menée au chapitre 3 nous a conduite à mesurer les performances prédictives avec cinq critères complémentaires : deux critères basés sur les rangs (un pour les labels et un pour les exemples que nous avons proposé) et trois critères basés sur la classification (accuracy, F-mesure et multi-label Balanced Error Rate (BER)). Les expérimentations ont été menées sur douze jeux de données de différents domaines (multimedia, biologie et texte) de complexité variable. La présentation des résultats est précédée d'une description du protocole expérimental qui simule la première phase de l'interaction avec une procédure très simplifiée où le nombre d'exemples d'apprentissage est très limité. Cette phase est très importante en pratique pour capter l'intérêt de l'utilisateur et acquérir sa confiance dans le système.

4. Le chapitre 5 « **Apprentissage multi-label à partir de données latentes** » décrit une nouvelle approche appelée FMDI-RF$^+$ pour réduire les temps d'apprentissage et de prédiction du meilleur classifieur retenu à l'issue des expérimentations du chapitre 4 (RF-PCT). Cette nouvelle approche hybride combine l'approche RF-PCT avec une approche de factorisation de matrice rapide pour la réduction de dimensions. Nous développons tout d'abord son principe puis présentons les résultats comparatifs avec RF-PCT qui apprend à partir des données originales de grande dimension.

Nous concluons ce manuscrit avec (*i*) un bilan sur les résultats obtenus dans cette thèse, (*ii*) une description de notre système prototype de classification interactive multi-label *VIPE* et de ses deux cas d'usages cibles et (*iii*) une discussion sur les perspectives à court et moyen terme qui s'orientent autour de deux directions possibles : l'amélioration de la nouvelle approche FMDI-RF+ pour la classification interactive multi-label et l'extension du protocole expérimental pour la comparaison des algorithmes.

2

Classification interactive

Sommaire

1 Introduction

Les approches de classification centrées sur l'utilisateur (« user-centered ») visent à produire des résultats individuels plus adaptés aux préférences de chaque utilisateur que les approches entièrement automatiques (e.g. Amershi (2011); Amershi et al. (2015); Fails and Olsen Jr (2003); Lintott et al. (2008); Porter et al. (2013); Ware et al. (2001)). L'utilisateur devient le coach qui annote, en positif ou en négatif, un nombre limité d'exemples pour expliquer ses préférences. À partir de ces quelques exemples, un algorithme d'apprentissage tente de capturer ces préférences pour apprendre un premier modèle prédictif qui lui fournit des prédictions personnalisées. En fonction de sa satisfaction, l'utilisateur peut soit arrêter l'apprentissage si son concept cible a

bien été appris ou continuer à entraîner le modèle en fournissant plus de précisions sur son concept cible. Pour affiner les prédictions, l'utilisateur peut corriger les mauvaises prédictions ou éventuellement ajouter de nouveaux exemples si son concept cible est difficile à apprendre (i.e. bouclage de pertinence (Salton and Buckley, 1997; Stumpf et al., 2007)). En fonction des actions de l'utilisateur, le modèle se met automatiquement à jour et tente de proposer des prédictions de plus en plus fines au fil du temps (Figure 1).

Formellement, supposons qu'un utilisateur ait pour objectif de classer un ensemble $\mathcal{X}$ de n exemples non-étiquetés $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ où chaque exemple x_i est défini dans un espace de m attributs numériques. Supposons qu'au début de l'interaction t_0, un utilisateur crée un label désiré et l'explique en annotant un petit ensemble $\mathcal{T}_{t_0} = \{(x_i, y_i)|\ i = 1..n_0\}$ de n_0 exemples où y_i est le label associé à l'exemple x_i : $y_i = 1$ (resp. 0) si l'exemple x_i est annoté positivement (resp. négativement). À partir de $\mathcal{T}_{t_0}$, un algorithme d'apprentissage apprend un modèle prédictif h_{t_0}. Si les prédictions fournies par le modèle ne correspondent pas bien aux préférences de l'utilisateur, il peut alors améliorer sa performance prédictive en corrigeant les mauvaises prédictions de h_{t_0} ou en ajoutant de nouveaux exemples à $\mathcal{T}_{t_0}$. Le processus d'apprentissage peut être répété plusieurs fois jusqu'à ce que l'utilisateur soit satisfait des prédictions de son classifieur.

L'importance croissante donnée actuellement aux contenus personnalisés a conduit au développement de plusieurs systèmes de classification interactive pour diverses applications que nous présentons brièvement dans ce chapitre : classification d'images (Fogarty et al., 2008), sélection de fichiers (Ritter and Basu, 2009), classification de gestes (Fiebrink et al., 2009), classification de documents (Drucker et al., 2011), reconnaissance d'écriture manuscrite (Shilman et al., 2006), tri d'alarmes (Amershi et al., 2011) et création de groupes d'utilisateurs dans les réseaux sociaux (Amershi et al., 2012).

2 Exemples de systèmes de classification interactifs

Pour illustrer ce que des utilisateurs, experts ou non, peuvent faire aujourd'hui avec des systèmes de classification interactive, nous présentons quelques systèmes récents dont les fonctionnalités sont résumées dans le tableau 1. L'apprentissage interactif est aussi présent dans d'autres

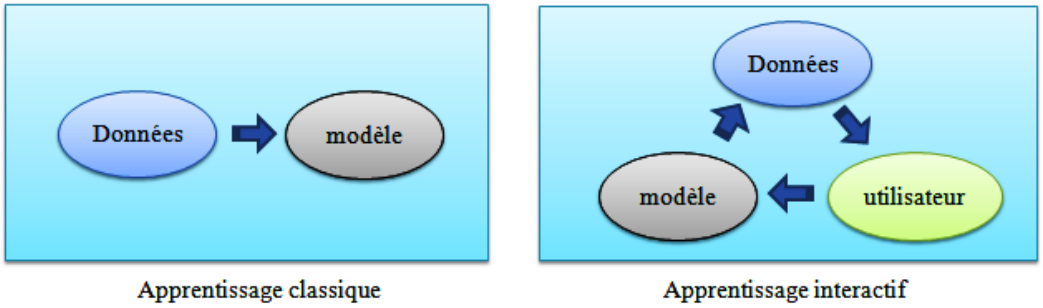


FIGURE 1 – L'apprentissage automatique standard et l'apprentissage automatique interactif.

domaines d'applications -par exemple l'interaction homme-robot (Goodrich and Schultz, 2007; Nicolescu and Mataric, 2001)- plus éloignés de notre secteur d'activité et nous ne les présentons donc pas ici.

cueTip (Shilman et al., 2006) est un classifieur qui permet de corriger les erreurs d'un modèle de reconnaissance d'écriture manuscrite (Figure 2). Le système interprète d'abord ce que l'utilisateur écrit sur une interface visuelle puis affiche la prédiction fournie par un algorithme de reconnaissance d'écriture. L'utilisateur peut ensuite corriger ce résultat si nécessaire en utilisant des gestes simples et intuitifs sur l'écran. Ces gestes sont codés par les développeurs du système. Par exemple, pour effacer une lettre, il faut la raturer et pour remplacer une lettre incorrecte, il faut réécrire au dessus. Les corrections fournies par l'utilisateur sont traduites en contraintes sur les données et transmises au modèle de reconnaissance d'écriture pour un nouvel apprentissage plus fin.

Wekinator (Fiebrink et al., 2009) est un classifieur de gestes qui permet, aux musiciens, compositeurs et nouveaux producteurs de sons, d'associer des sons à des gestes en temps réel. Le classifieur de gestes obtenu peut être utilisé par exemple pour animer des spectacles musicaux. Pour entraîner le classifieur, un utilisateur illustre chaque geste cible (i.e. classe) avec un exemple ainsi que le son qu'il doit déclencher. À partir des exemples illustrés par l'utilisateur, un algorithme d'apprentissage sélectionné de la libraire Weka (Witten and Frank, 2005) apprend un modèle prédictif qui reconnait automatiquement ses gestes et joue leurs sons associés. Pour renforcer si nécessaire le modèle appris, l'utilisateur peut fournir des exemples supplémentaires ou en modifier les précédents.

iCluster (Drucker et al., 2011) est un classifieur de documents qui permet de détecter les documents préférés d'un utilisateur (Figure 4). Avant la phase d'apprentissage, l'utilisateur crée

FIGURE 2 – Le système *cueTip* pour la correction d'erreurs en reconnaissance d'écriture manuscrite (Shilman et al., 2006).

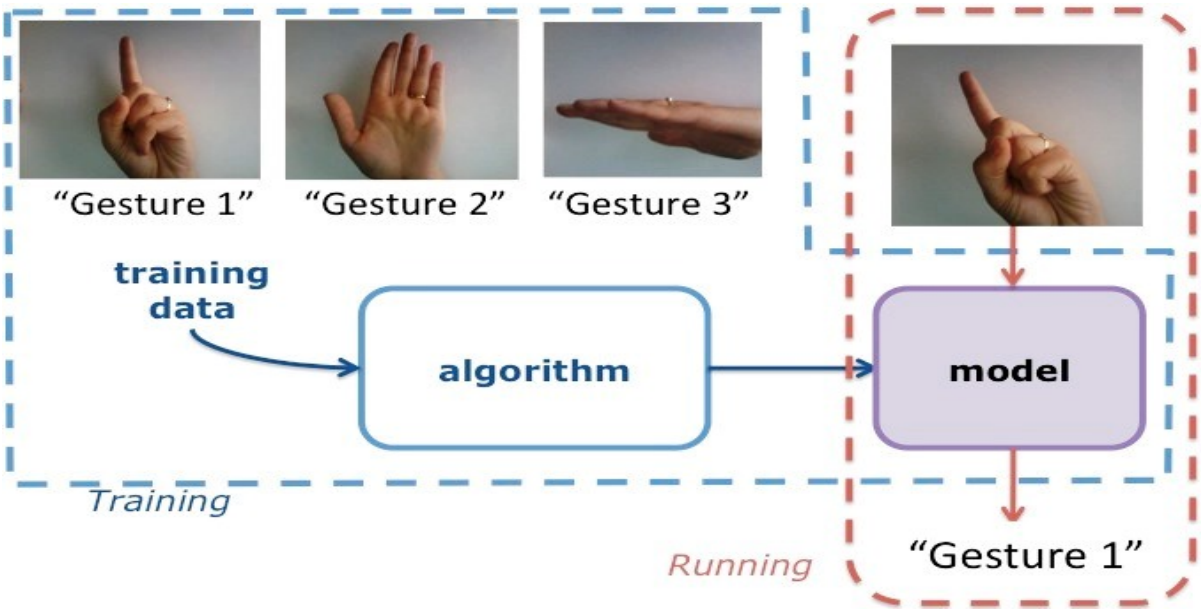


FIGURE 3 – Le système *Wekinator* pour la classification de gestes (Fiebrink et al., 2009).

ses labels cibles. Pour expliquer chaque label, l'utilisateur sélectionne quelques exemples représentatifs (i.e. positifs). Les exemples des autres labels sont considérés comme ses exemples négatifs. En se basant seulement sur quelques exemples fournis par l'utilisateur, une régression logistique apprend un modèle prédictif qui fournit, pour chaque nouveau document, ses probabilités d'appartenance aux différents labels. Un algorithme d'apprentissage de métrique est aussi utilisé pour prédire les 20 documents les plus proches de chaque label cible. De manière

interactive, l'utilisateur inspecte les prédictions des modèles et ajoute au fur et à mesure de nouveaux exemples pour affiner les prédictions.

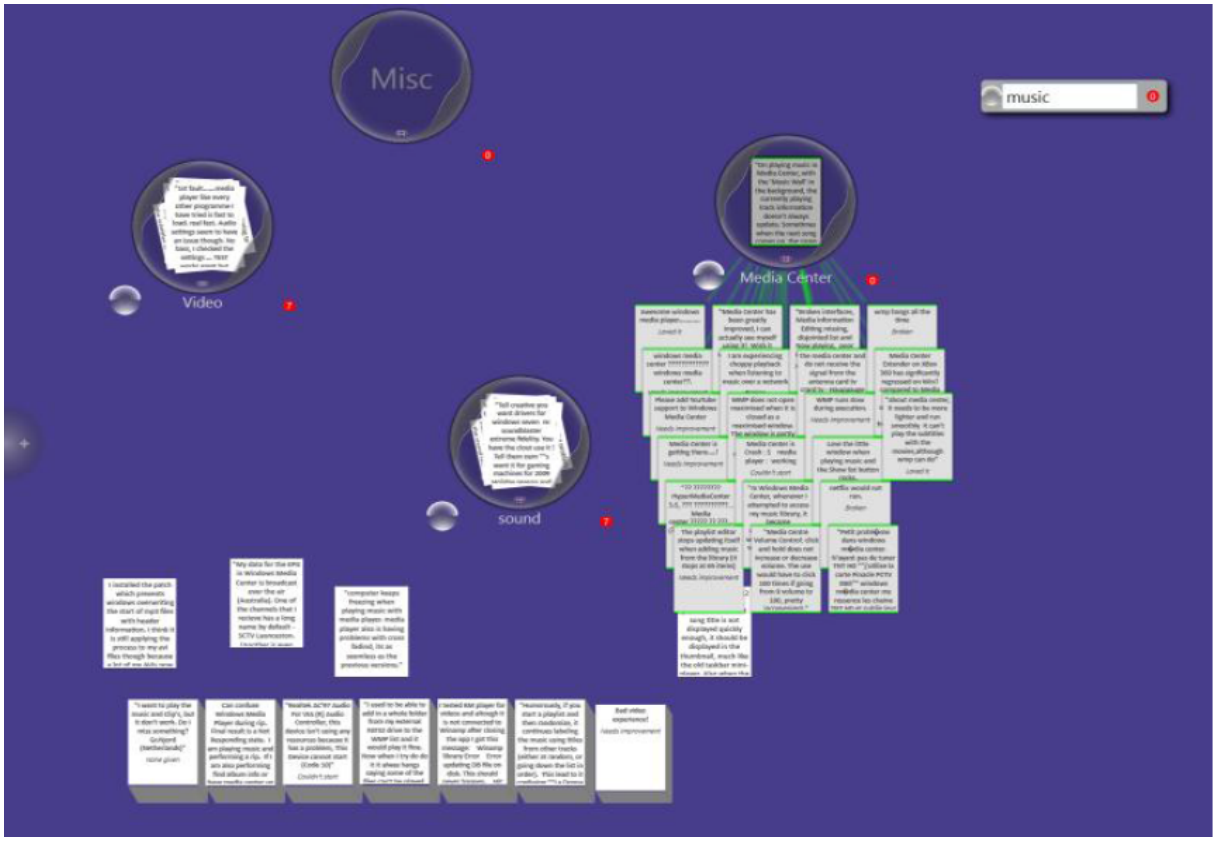


FIGURE 4 – Le système *iCluster* pour la classification de documents (Drucker et al., 2011).

Smart selection (Ritter and Basu, 2009) est un classifieur de fichiers proche du gestionnaire de fichiers de Windows^{TM}. Il permet d'assister un utilisateur dans des tâches de sélection de fichiers complexes (Figure 5). Par exemple, un utilisateur souhaite sélectionner tous les fichiers dont le nom contient le mot "MARKF" ou "JOEP" mais pas le mot "TMP" pour les classer dans un dossier à part. Pour une tâche de sélection, l'utilisateur sélectionne uniquement quelques fichiers désirés. Les fichiers non-sélectionnés sont considérés comme exemples négatifs. À partir de quelques exemples de fichiers sélectionnés, un modèle prédictif est appris pour généraliser automatiquement la sélection aux autres fichiers. Ce modèle est construit à l'aide d'un boosting d'arbres de décision de taille 2. Pour fournir plus de précision, l'utilisateur peut continuer à fournir de nouveaux exemples et contre-exemples en sélectionnant ou en désélectionnant des fichiers jusqu'à ce que tous les fichiers désirés soient sélectionnés.

ReGroup (Amershi et al., 2012) est un classifieur de profils qui permet de reconnaitre des profils désirés dans les réseaux sociaux (Figure 6). Pour expliquer son profil cible, l'utilisateur

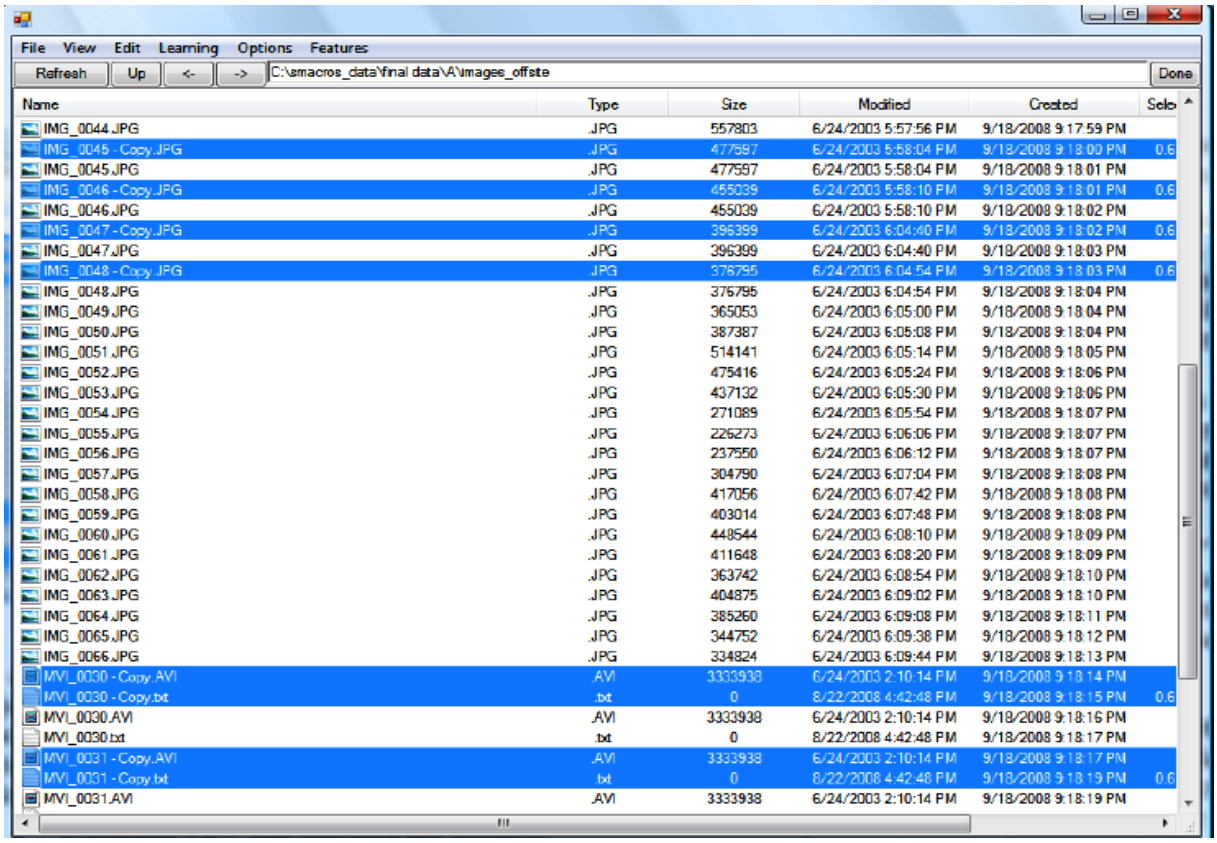


FIGURE 5 – Le système *Smart Selection* pour la selection de fichiers (Ritter and Basu, 2009).

effectue d'abord une recherche par mots clés et sélectionne quelques profils à partir de la liste fournie par le moteur de recherche du réseau (i.e. exemples positifs). Les personnes n'ayant pas été sélectionnées sont considérées comme indésirables (i.e. exemples négatifs). En se basant sur cette première liste, un modèle prédictif est appris pour fournir de nouveaux profils susceptibles d'intéresser l'utilisateur. Ce modèle est appris par un classifieur bayésien naïf. Pour fournir davantage de précisions sur le profil cible, l'utilisateur trie les profils proposés à chaque itération par le modèle et ainsi de suite jusqu'à obtenir une liste satisfaisante.

cueT (Amershi et al., 2011) est un classifieur d'alarmes qui permet d'assister les opérateurs dans le tri alarmes (Figure 7). Un opérateur définit d'abord un ensemble de labels cibles (e.g. alarme inconnue, alarme majeure, alarme normale, *etc*) et annote ensuite manuellement quelques alarmes en fonction de leur gravité. Un algorithme de k plus proches voisins ($k$ppv) se base sur ces exemples d'alarmes et prédit les labels des nouvelles alarmes. Cet algorithme est combiné avec une métrique apprise à partir des actions des opérateurs. Si la distance estimée entre une nouvelle alarme et les exemples de l'ensemble d'apprentissage est inférieure à un seuil, *cueT* recommande à l'utilisateur de créer un nouveau label pour cette alarme inconnue.

FIGURE 6 – Le système *Regroup* pour la création de groupes dans les réseaux sociaux (Amershi et al., 2012).

Pour expliquer davantage quelques alarmes et améliorer les prédictions du modèle, l'opérateur peut étiqueter manuellement de nouveaux exemples ou corriger les mauvaises prédictions.

cueFlick (Fogarty et al., 2008) est un classifieur d'images qui permet de reconnaitre automatiquement un concept visuel désiré par l'utilisateur (Figure 8). Pour expliquer son concept cible (e.g. photos avec des couleurs psychédéliques lumineuses), l'utilisateur interroge d'abord le catalogue d'images avec des mots clés et sélectionne quelques exemples avec et sans les caractéristiques visuelles désirées à partir des résultats retournés. Un algorithme d'apprentissage de k plus proches voisins combiné avec une métrique reclasse ensuite les images. Cette métrique est apprise en fonction des interactions de l'utilisateur avec les prédictions du modèle. L'utilisateur peut sélectionner (i.e. exemple positif) ou déselectionner (i.e. exemple négatif) manuellement d'autres images pour aider son modèle dans l'apprentissage d'un concept complexe.

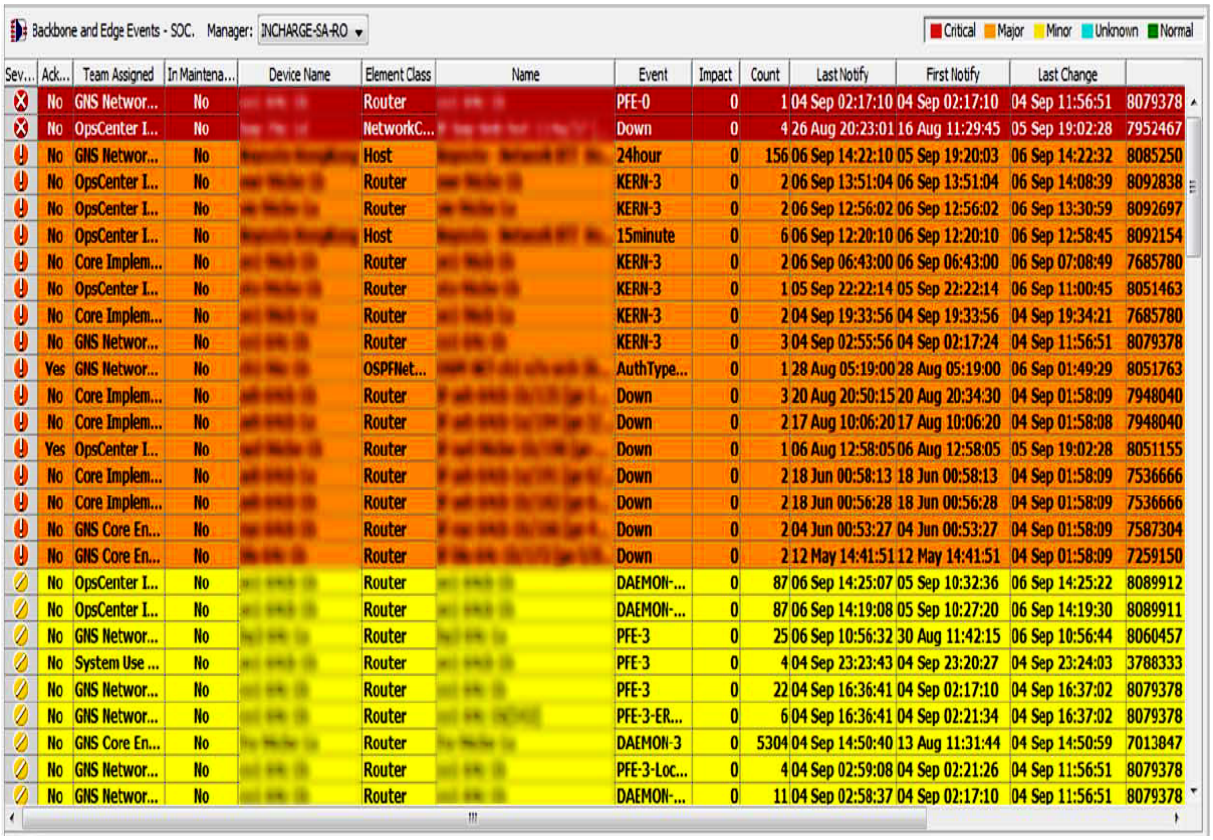


FIGURE 7 – Le système *cueT* pour le tri d'alarmes (Amershi et al., 2011).

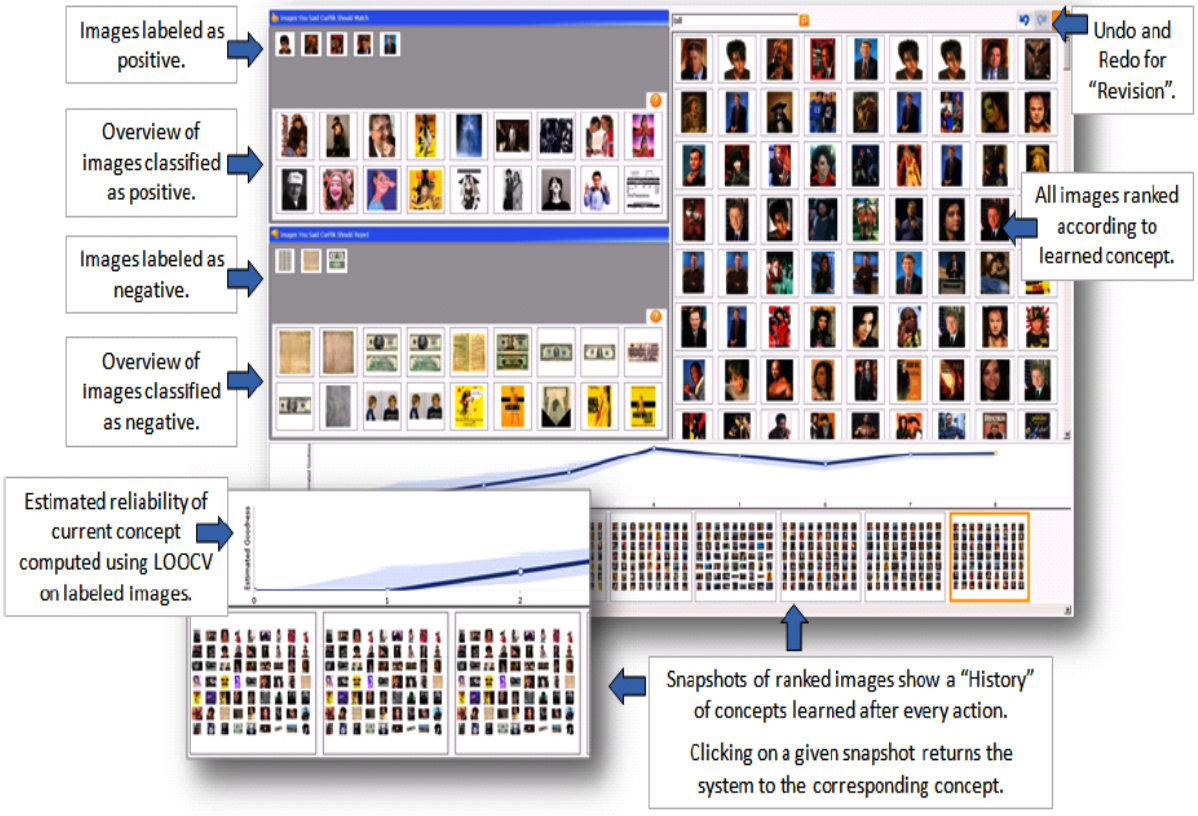


FIGURE 8 – Le système *CueFlik* pour la classification d'images (Fogarty et al., 2008).

Système	Tâche	Algorithme d'apprentissage
Regroup	création de groupes sociaux	bayésien naïf
iCluster	classification de document	régression logistique et $k$ppv + apprentissage de métrique
cueT	tri d'alarmes	$k$ppv + apprentissage de métrique
Wekinator	classification de gestes	Algorithmes de Weka
Smart selection	sélection de fichiers	boosting d'arbres de décision de taille 2
cueFLIK	recherche d'images	$k$ppv + apprentissage de métrique
cueTip	reconnaissance d'écriture manuscrite	modèle de reconnaissance d'écriture contraint

TABLE 1 – Résumé des systèmes de classification interactive récents.

3 Discussion

Les retours expérimentaux décrits dans les articles présentant ces systèmes de classification interactifs semblent prometteurs. Cependant, les évaluations menées sont fondées sur des protocoles encore très limités. Dans la plupart des travaux, elles sont basées sur des avis d'un échantillon très restreint d'utilisateurs (12 individus en moyenne (Amershi et al., 2012; Fogarty et al., 2008; Ritter and Basu, 2009; Shilman et al., 2006)) pour une tâche de classification spécifique. Certains auteurs (e.g. Amershi et al. (2011); Fogarty et al. (2008)) ajoutent quelques mesures quantitatives (e.g. précision, durée de l'expérimentation, vitesse d'apprentissage). Mais, leurs comparaisons avec d'autres algorithmes sont très restreintes et les jeux de test se réduisent souvent à un seul. Et à notre connaissance, il n'existe pas encore de protocole d'évaluation standardisé qui permettrait de comparer, dans un cadre partagé, les performances des différents classifieurs utilisés dans ces systèmes. De façon générale, l'évaluation d'un système pour une tâche complexe dans un environnement qui évolue reste une question délicate discutée dans la communauté Interface Homme Machine (IHM) (e.g. Cockton (2007); Greenberg and Buxton (2008); Zhai (2003)). Dans cette thèse nous n'abordons qu'un aspect restreint – mais important- de la problématique ; celui de la validation des performances d'apprentissage des algorithmes dans un contexte interactif très simplifié qui omet les composantes IHM.

Dans ce contexte, nous pouvons noter que la plupart des systèmes de classification interactive récents se limitent à une classification mono-label où un seul label peut être affecté à la fois à un exemple ; ce qui est peu expressif d'autant plus que les données sont très souvent de nature

multi-label. Par exemple, pour le système *cueFlik* une image peut contenir plusieurs concepts visuels, pour le système *iCluster* un document peut évoquer plusieurs sujets et un utilisateur considéré dans le système *Regroup* peut faire partie de plusieurs groupes sociaux. L'étiquetage multi-label est donc nécessaire pour permettre aux utilisateurs d'exprimer davantage leurs préférences sur les données. Nous abordons cette problématique dans le chapitre suivant.

3

Apprentissage multi-label

Sommaire

1 Introduction

L'apprentissage multi-label est une généralisation de l'apprentissage mono-label classique où chaque exemple x_i peut être étiqueté par un ou plusieurs labels simultanément. L'apprentissage à partir de données multi-label est un problème qui a suscité beaucoup d'attention ces dernières années par la communauté d'apprentissage automatique et les communautés connexes (Madjarov et al., 2012; Sorower, 2010; Tsoumakas and Katakis, 2007; Tsoumakas et al., 2010; Zhang and Zhou, 2013) et a conduit au développement de plusieurs approches d'apprentissage avec différentes stratégies. Ces approches peuvent être organisées en trois grandes familles : approches d'apprentissage par transformation, approches d'apprentissage par adaptation et approches d'apprentissage ensemble. Initialement développés pour la catégorisation de texte, les algorithmes d'apprentissage multi-label ont été depuis appliqués à divers problèmes réels : classification de contenus multimédia tels que les images (Boutell et al., 2004), les sons (Lo et al., 2011) et les vidéos (Snoek et al., 2006), fouille de web et de règles (Ozonat and Young, 2009; Rak et al., 2005), bio-informatique (Clare and King, 2001), recommandation de tags (Katakis et al., 2008) et recherche d'information (Yu et al., 2005).

Dans ce qui suit, nous présentons onze approches parmi les méthodes d'apprentissage les plus représentatives des trois grandes familles. Nous précisons le critère qu'elles optimisent implicitement ou explicitement et rappelons leurs complexités d'apprentissage et de prédiction dans le pire des cas ainsi que leurs avantages et inconvénients.

2 Définition du problème d'apprentissage multi-label

Dans la suite, on considère un espace $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ de m attributs numériques tel que $dom(f_j) \in \mathcal{R}$, et un espace $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ de q labels tel que $dom(\lambda_k) \in \{0, 1\}$ (0 : négatif, 1 : positif). Pour l'apprentissage, on se base sur un jeu de données multi-label $\mathcal{D}$ avec un sous-ensemble de données étiquetées $\mathcal{T} \subset \mathcal{D}$ (i.e. ensemble d'apprentissage) et un sous-ensemble de données non-étiquetées $\mathcal{S} \subset \mathcal{D}$ (i.e. ensemble de test). L'ensemble $\mathcal{T} = \{(x_i, y_i) |\ i = 1..n\}$ est un jeu d'apprentissage multi-label où chaque exemple $x_i \in \mathcal{T}$ est décrit par m attributs ($dom(x_i) \in \mathcal{R}^m$) et est annoté positivement ou négativement par q labels ($dom(y_i) \in \{0, 1\}^q$). Plus précisément, y_i est le vecteur binaire des labels associé à un exemple

x_i tels que y_i^+ et y_i^- représentent respectivement l'ensemble des labels positifs et négatifs de x_i avec $|y_i^+| + |y_i^-| = q$. Ainsi, $y_i^k = 1$ si x_i appartient au label λ_k, sinon $y_i^k = 0$. Cette appartenance peut également être exprimée par un degré compris dans l'intervalle [0..1] (i.e. classification floue ou classification graduelle (Cheng et al., 2010)) mais nous ne considérons pas ces extensions ici. Un algorithme d'apprentissage multi-label apprend un modèle prédictif $h : \mathcal{R}^m \rightarrow [0..1]^q$: pour un nouvel exemple $x_i \in \mathcal{S}$, le modèle appris h prédit l'ensemble des labels les plus probables $\hat{y}_i = h(x_i)$.

La plupart des algorithmes d'apprentissage multi-label retournent des vecteurs de prédictions définies dans l'intervalle [0..1]. Pour transformer ces prédictions en valeurs binaires, une fonction de seuillage est requise. Plusieurs approches de seuillage existent mais *Proportional Cut Method* (*Pcut*) est actuellement l'approche la plus utilisée de part de sa simplicité et de son efficacité. Formellement, *Pcut* cherche à trouver une valeur z^* (le seuil) qui minimise la différence entre le nombre moyen de labels dans l'ensemble d'apprentissage $\mathcal{T}$ et dans l'ensemble de test classé par le modèle $h(\mathcal{S})$ où $f_z : [0..1]^q \rightarrow \{0,1\}^q$ est une fonction de seuillage qui transforme les valeurs supérieures à z en uns (1) ou zéros (0) sinon :

$$z^* = \underset{z \in \{0.00,0.001,\ldots,1.00\}}{\operatorname{argmin}} \left| \frac{1}{|\mathcal{T}|} \sum_{i=0}^{|\mathcal{T}|} |y_i^+| - \frac{1}{|\mathcal{S}|} \sum_{i=0}^{|\mathcal{S}|} |f_z(\hat{y}_i)^+| \right|$$

3 Approches d'apprentissage multi-label

Les approches d'apprentissage multi-label peuvent être organisées en trois grandes familles (Madjarov et al., 2012; Tsoumakas et al., 2010; Zhang and Zhou, 2013) :

1. **Approches d'apprentissage par transformation** : elles transforment le problème d'apprentissage multi-label en un ou plusieurs problèmes de classification ou de régression mono-label,

2. **Approches d'apprentissage par adaptation** : elles adaptent des algorithmes d'apprentissage pour des données multi-label,

3. **Approches d'apprentissage ensemble** : elles utilisent un ensemble de classifieurs issus de la première ou de la deuxième famille d'approches.

Les caractéristiques des approches présentées sont résumées dans le tableau 6. Les complexités de calcul de chaque classifieur (en apprentissage et en prédiction) sont données dans les pires cas en fonction du nombre d'exemples d'apprentissage (n), du nombre d'attributs (m) et du nombre de labels (q). Les approches d'apprentissage par transformation utilisent toutes un classifieur binaire B ou multi-classes M pour l'apprentissage du ou des modèles (algorithme de construction d'un arbre de décision C4.5 ou un SVM (Support Vector Machine)). Nous l'appelons dans ce qui suit "classifieur de base". Pour les complexités de ce type d'approches d'apprentissage, $h_B(n, m)$ (resp. $h_M(n, m, q)$) et $h'_B(m)$ (resp. $h'_M(m, q)$) dénotent les complexités d'apprentissage et de prédiction du classifieur de base B (resp. M) utilisé.

3.1 Approches d'apprentissage par transformation

Pour illustrer les processus d'apprentissage et de prédiction de chaque méthode d'apprentissage par transformation, nous définissons d'abord un petit jeu de données multi-label dans le tableau 1. Nous lui appliquons ensuite toutes les méthodes (présentées ci-dessous) dans les tableaux : 2, 3, 4 et 5.

Exemple jouet : Soit un ensemble d'apprentissage multi-label de 4 exemples où chaque exemple x_i est décrit par m attributs et est annoté positivement ou négativement par 4 labels. Par exemple, x_4 possède 3 labels positifs $y_4^+ = \{\lambda_2, \lambda_3, \lambda_4\}$ et un label négatif $y_4^- = \{\lambda_1\}$. À partir de ces exemples d'apprentissage, un classifieur multi-label apprend un modèle de classification h et tente ensuite de prédire l'ensemble des labels les plus appropriés $\hat{y_5} = h(x_5)$ pour un nouvel exemple x_5.

x_i	f_1	$\ldots$	f_m	λ_1	λ_2	λ_3	λ_4
x_1				1	0	0	1
x_2				0	0	1	1
x_3				1	0	0	0
x_4				0	1	1	1
x_5				?	?	?	?

TABLE 1 – Exemple de jeu de données multi-label

Binary Relevance (*BR*) (Schapire and Singer, 2000) est la méthode la plus populaire et la plus simple de cette classe d'approches. Elle transforme le problème d'apprentissage multi-

label en q problèmes de classification ou de régression mono-label. Pour l'apprentissage de chaque label λ_i $(1 \leq i \leq q)$, un classifieur binaire h_{Bi} est utilisé (e.g. Table 2). Pour un nouvel exemple de test (e.g. x_5), BR retourne l'union des prédictions de chaque classifieur. BR a une complexité de calcul de $\mathcal{O}(q \,.\, h_B(n, m))$ pour l'apprentissage des q modèles et $\mathcal{O}(q \,.\, h'_B(m))$ pour la prédiction des labels d'un nouvel exemple. L'avantage majeur de BR réside dans sa faible complexité en apprentissage (relative à un classifieur de base) qui lui permet de passer facilement à l'échelle et d'être donc une très bonne candidate pour des problèmes d'apprentissage multi-label à partir de données de grande dimension. Cependant, BR ignore l'existence de corrélations potentielles entre les labels. De plus, les classifieurs binaires peuvent souffrir du déséquilibre entre les classes (1 et 0) si le nombre de labels est grand et la densité des labels est faible.

x_i	f_1	...	f_m	λ_1
x_1				1
x_2				0
x_3				1
x_4				0
x_5				?

x_i	f_1	...	f_m	λ_2
x_1				0
x_2				0
x_3				0
x_4				1
x_5				?

x_i	f_1	...	f_m	λ_3
x_1				0
x_2				1
x_3				0
x_4				1
x_5				?

x_i	f_1	...	f_m	λ_4
x_1				1
x_2				1
x_3				0
x_4				1
x_5				?

TABLE 2 – Transformation du jeu de données suivant l'approche d'apprentissage BR.

Classifier Chain (*CC*) (Read et al., 2009) est une amélioration de la méthode BR qui transforme également le problème d'apprentissage multi-label en q problèmes de classification ou de régression mono-label. Cependant, les classifieurs sont entraînés dans un ordre aléatoire défini avant la phase d'apprentissage $[1..,i,..q]$ tel que chaque classifieur binaire h_{Bi} apprenant un label λ_i ajoute tous les labels associés aux classifieurs qui le précédent dans la chaîne (i.e. $\lambda_1, \ldots, \lambda_{i-1}$) dans son espace d'attributs (e.g. Table 3 où l'ordre des labels est [λ_1, λ_2, λ_3, λ_4]). Comme BR, pour un nouvel exemple (e.g. x_5), CC retourne l'ensemble des prédictions générées par l'ensemble des classifieurs. CC a une complexité de calcul de $\mathcal{O}(q \,.\, h_B(n, m+q))$ pour

l'apprentissage des q modèles et $\mathcal{O}(q \ . \ h'_B(m+q))$ pour la prédiction des labels d'un nouvel exemple. Son avantage est sa vitesse d'apprentissage du modèle et sa modélisation des corrélations entre les labels mais sa définition aléatoire de l'ordre d'apprentissage des modèles reste une faiblesse.

x_i	f_1	...	f_m	λ_1
x_1				1
x_2				0
x_3				1
x_4				0
x_5				$\hat{y}_5^1$

x_i	f_1	...	f_m	$f_{m+1} = \lambda_1$	λ_2
x_1				1	0
x_2				0	0
x_3				1	0
x_4				0	1
x_5				$\hat{y}_5^1$	$\hat{y}_5^2$

x_i	f_1	...	f_m	$f_{m+1} = \lambda_1$	$f_{m+2} = \lambda_2$	λ_3
x_1				1	0	0
x_2				0	0	1
x_3				1	0	0
x_4				0	1	1
x_5				$\hat{y}_5^1$	$\hat{y}_5^2$	$\hat{y}_5^3$

x_i	f_1	...	f_m	$f_{m+1} = \lambda_1$	$f_{m+2} = \lambda_2$	$f_{m+3} = \lambda_3$	λ_4
x_1				1	0	0	1
x_2				0	0	1	1
x_3				1	0	0	0
x_4				0	1	1	1
x_5				$\hat{y}_5^1$	$\hat{y}_5^2$	$\hat{y}_5^3$	$\hat{y}_5^4$

TABLE 3 – Transformation du jeu de données suivant l'approche d'apprentissage CC.

Label Powerset (*LP*) (Tsoumakas and Katakis, 2007) transforme le problème d'apprentissage multi-label en un seul problème d'apprentissage mono-label à plusieurs classes. Elle considère chaque combinaison de labels présente dans l'ensemble d'apprentissage comme une classe et apprend ensuite un classifieur multi-classes h_M (Table 4). Pour un nouvel exemple (e.g. x_5), le classifieur retourne la classe (i.e. combinaison de labels) la plus probable. LP a une complexité de calcul de $\mathcal{O}(h_M(n, m, 2^q))$ pour l'apprentissage du modèle et $\mathcal{O}(h'_M(m, 2^q))$ pour la prédiction des labels d'un nouvel exemple. L'avantage principal de LP est sa faible complexité de calcul du modèle mais aussi son exploitation naturelle des corrélations entre labels. Néanmoins, quelques classes peuvent être difficiles à apprendre si le nombre de labels est important et le nombre d'exemples est faible. Le nombre de classes est au plus égal à min$(2^q, n)$.

Son autre inconvénient est qu'elle ne permet pas de bien généraliser : elle ne permet pas de prédire de nouvelles classes (combinaisons de labels) qui n'existent pas dans l'ensemble d'apprentissage.

x_i	f_1	...	f_m	y_i
x_1				$\lambda_{1,4}$
x_2				$\lambda_{3,4}$
x_3				λ_1
x_4				$\lambda_{2,3,4}$
x_5				?

TABLE 4 – Transformation du jeu de données suivant l'approche d'apprentissage LP.

Calibrated Label Ranking (*CLR*) (Fürnkranz et al., 2008) transforme le problème d'apprentissage multi-label en $\frac{q.(q+1)}{2}$ problèmes de classification ou de régression mono-label. Précisément, elle entraîne $\frac{q.(q-1)}{2}$ classifieurs binaires $h_B i$ pour l'apprentissage de chaque combinaison de labels de taille 2 et q autres classifieurs binaires pour l'apprentissage de chaque label (i.e. Binary Relevance). Pour l'apprentissage de chaque paire de labels (λ_i, λ_j) $(i, j \in \{1, 2, \ldots, q\}$ et $i \neq j)$, l'ensemble d'apprentissage est construit de telle sorte que les exemples positifs appartiennent au premier label λ_i et les exemples négatifs appartiennent au deuxième label λ_j (e.g. Table 5). Ainsi, le nombre d'exemples associés à chaque classifieur est réduit à un nombre plus petit que n. Pour séparer les labels positifs des labels négatifs, CLR introduit un label virtuel λ_0 et le combine avec chaque label λ_i $(1 \leq i \leq q)$. Pour apprendre ces q nouvelles paires de labels, l'approche BR est utilisée. Pour un nouvel exemple (e.g. x_5), les prédictions de tous les modèles sont agrégées et un ranking est associé à chaque label. Pour transformer le ranking en classification, CLR utilise le nombre de votes attribués à λ_0 comme seuil de séparation entre les labels positifs et les labels négatifs. CLR a une complexité de calcul de $\mathcal{O}(q^2 \,.\, h_B(n, m))$ pour l'apprentissage des modèles et $\mathcal{O}(q^2 \,.\, h'_B(m))$ pour la prédiction des labels d'un nouvel exemple. L'avantage de CLR est d'exploiter les corrélations entre les labels. Elle permet aussi d'atténuer l'influence négative du déséquilibre entre classes sur les performances des classifieurs binaires. Néanmoins, CLR n'est pas adaptée aux données avec un grand nombre de labels car son coût d'apprentissage est élevé. Par exemple, plus de 5000 modèles sont requis pour apprendre 100 labels.

Hierarchy Of Multi-label classifiERs (HOMER) (Tsoumakas et al., 2008) transforme le

x_i	f_1	...	f_m	$\lambda_1 - \lambda_2$
x_1				1
x_3				1
x_4				0
x_5				?

x_i	f_1	...	f_m	$\lambda_1 - \lambda_3$
x_1				1
x_2				0
x_3				1
x_4				0
x_5				?

x_i	f_1	...	f_m	$\lambda_1 - \lambda_4$
x_2				0
x_3				1
x_4				0
x_5				?

x_i	f_1	...	f_m	$\lambda_2 - \lambda_3$
x_2				0
x_5				?

x_i	f_1	...	f_m	$\lambda_2 - \lambda_4$
x_1				0
x_2				0
x_5				?

x_i	f_1	...	f_m	$\lambda_3 - \lambda_4$
x_1				0
x_5				?

x_i	f_1	...	f_m	$\lambda_1 - \lambda_0$
x_1				1
x_2				0
x_3				1
x_4				0
x_5				?

x_i	f_1	...	f_m	$\lambda_2 - \lambda_0$
x_1				0
x_2				0
x_3				0
x_4				1
x_5				?

x_i	f_1	...	f_m	$\lambda_3 - \lambda_0$
x_1				0
x_2				1
x_3				0
x_4				1
x_5				?

x_i	f_1	...	f_m	$\lambda_4 - \lambda_0$
x_1				1
x_2				1
x_3				0
x_4				1
x_5				?

TABLE 5 – Transformation du jeu de données suivant l'approche d'apprentissage CLR.

problème d'apprentissage multi-label en plusieurs problèmes de classification ou de régression mono-label. Elle construit une hiérarchie de classifieurs multi-label telle que chaque classifieur se focalise sur un ensemble de labels de petite taille comparée à q et sur un ensemble d'apprentissage plus équilibré dans l'espace de labels. Pour organiser automatiquement les labels dans une hiérarchie, HOMER utilise un algorithme de clustering avancé qui divise de manière récursive et uniforme l'ensemble de labels en k sous-ensembles de même taille. Pour l'appren-

tissage, chaque nœud filtre les exemples de son parent et conserve uniquement ceux qui sont annotés avec au moins l'un de ses labels. Un classifieur multi-label est ensuite entraîné dans chaque nœud (hormis les feuilles) en se basant sur l'approche LP. Pour la prédiction de labels d'un nouvel exemple x_i, HOMER interroge d'abord le classifieur du nœud racine puis transfère récursivement x_i vers le classifieur d'un noeud fils dont les labels associés font partie des prédictions du classifieur parent pour x_i. L'union des prédictions des modèles constitue la prédiction finale. Sa complexité de calcul est de $\mathcal{O}(C(q) \ + \ d \ . \ h_M(n, m, 2^{q/k}))$ où $C(.)$ et d dénotent respectivement la complexité de calcul de l'algorithme de clustering avancé et le nombre de nœuds dans la hiérarchie. Pour la prédiction de labels d'un nouvel exemple, il a une complexité de $\mathcal{O}(log_k(q) \ . \ h'_M(m, 2^{q/k}))$ où $log_k(q)$ représente la profondeur de la hiérarchie. HOMER est un bon candidat pour des problèmes d'apprentissage où le nombre de labels est important (à partir d'une centaine). En utilisant un algorithme de clustering avancé, il tire bien avantage des corrélations entre les labels et évite le problème du déséquilibre entre classes (0 et 1). Néanmoins, il suppose que les labels peuvent être organisés sous forme d'une hiérarchie. Or, cette hypothèse n'est pas toujours garantie. De plus, le coût de construction de la hiérarchie peut très vite devenir important si le nombre de labels et le nombre d'exemples sont grands.

Les méthodes d'apprentissage par transformation sont flexibles : elle peuvent utiliser n'importe quel classifieur mono-label. Néanmoins, la qualité de leurs performances prédictives dépend principalement du choix de ce classifieur de base. En général, deux classifieurs sont utilisés (Madjarov et al., 2012; Read, 2010; Tawiah and Sheng, 2013) : Support Vector Machine (SVM) (Cortes and Vapnik, 1995) et l'algorithme de construction d'arbre de décision C4.5 (Quinlan, 1993) avec un avantage au SVM en terme de performance prédictive. Une étude comparative de l'impact des classifieurs mono-label sur la performance prédictive des méthodes de transformation confirme l'effacité de ces classifieurs de base (Read, 2010).

3.2 Approches d'apprentissage par adaptation

Multi-Label kNN (*ML-kNN*) (Zhang and Zhou, 2007) est une méthode de type BR qui combine l'algorithme standard de kNN avec une inférence bayésienne. En phase d'apprentissage, ML-kNN estime les probabilités *a priori* et *a posteriori* de chaque label à partir des exemples d'apprentissage. Pour un nouvel exemple x_i, ML-kNN calcule ses k plus proches voisins puis

mesure la fréquence de chaque label dans ce voisinage. Cette fréquence est ensuite combinée avec les probabilités estimées dans la phase d'apprentissage pour déterminer son ensemble de labels en suivant le principe du maximum *a posteriori* (*MAP*). ML-kNN a l'avantage de tirer parti à la fois de l'apprentissage paresseux et du raisonnement bayésien : la frontière de décision peut être ajustée de manière adaptative dûe à la variabilité des voisins des nouveaux exemples et le problème du déséquilibre entre classes peut être largement atténué à l'aide des probabilités *a priori* estimées pour chaque label. Néanmoins, comme toute approche de type kNN, le temps de prédiction croît linéairement avec la taille de l'ensemble d'apprentissage.

Instance-Based Learning by logistic Regression (*IBLR_ML*) (Cheng and Hüllermeier, 2009) est une amélioration du ML-kNN qui combine un kNN avec une régression logistique. Elle permet d'enrichir la description de chaque exemple d'apprentissage x_i en lui ajoutant q nouveaux attributs tels que chaque attribut estime le fréquence d'un label dans le voisinage de x_i (ses k plus proches voisins). Une régression logistique est ensuite appliquée pour l'apprentissage de chaque label (approche de type BR). Pour un nouvel exemple, IBLR_ML combine les prédictions des q modèles. L'avantage de IBLR_ML est la modélisation des corrélations entre les labels mais aussi la réduction du temps de prédiction. Cependant, son temps d'apprentissage peut être important car elle fait appel à deux approches d'apprentissage : un KNN et une régression logistique.

ML-kNN et IBLR_ML partagent les mêmes complexités de calcul en apprentissage et en prédiction : $\mathcal{O}(n^2 \,.\, m \,+\, q \,.\, n \,.\, k)$ pour l'apprentissage et $\mathcal{O}(n \,.\, m \,+\, q \,.\, k)$ pour la prédiction des labels d'un nouvel exemple.

3.3 Approches d'apprentissage ensemble

RAndom k labEL sets (RAkEL) (Tsoumakas and Vlahavas, 2007) est une amélioration de l'approche LP. Elle entraîne un ensemble de N classifieurs multi-classes h_M suivant l'approche LP tel que chaque classifieur est construit sur un sous-ensemble de labels aléatoire de petite taille $k < q$. Pour l'apprentissage de chaque sous-ensemble de labels, seuls les exemples qui possèdent au moins l'un de ses labels sont considérés. Pour un nouvel exemple x_i, les prédictions des N modèles sont agrégées pour obtenir son ensemble de labels. RAkEL a une complexité de calcul de $\mathcal{O}(N \,.\, h_M(n, m, 2^k))$ pour l'apprentissage des modèles et $\mathcal{O}(N \,.\, h'_M(m, 2^k))$

pour la prédiction des labels d'un nouvel exemple. L'avantage de la méthode RAkEL est qu'elle permet de prédire de nouvelles combinaisons de labels qui n'existent pas dans l'ensemble d'apprentissage. De plus, elle permet d'exploiter naturellement les corrélations entre les labels. Néanmoins, elle n'explore pas suffisamment tous les sous-ensembles de labels pour capturer toutes les corrélations et se focalise uniquement sur l'apprentissage de quelques sous-ensembles de taille k.

Ensemble de Binary Relevance (EBR) et **Ensemble de Classifier Chains** (ECC) (Read et al., 2009) entraînent respectivement N classifieurs BR et N classifieurs CC. Pour diversifier les classifieurs, elles sous-échantillonnent de façon répétitive (avec remise) l'ensemble d'apprentissage (i.e. bagging (Breiman, 1996)). Par ailleurs, chaque classifieur CC est entraîné suivant un ordre de label différent défini aléatoirement. EBR et ECC ont une complexité de calcul de $\mathcal{O}(N\ .\ q\ .\ h_B(n,m))$ et $\mathcal{O}(N\ .\ q\ .\ h_B(n,m+q))$ respectivement pour l'apprentissage des modèles, et $\mathcal{O}(N\ .\ q\ .\ h'_B(m))$ et $\mathcal{O}(N\ .\ q\ .\ h'_B(m+q))$ respectivement pour la prédiction des labels d'un nouvel exemple. L'avantage de ces deux méthodes ensemble est qu'elles tentent d'améliorer les performances de leurs classifieurs de base (BR et CC) en multipliant les modèles. Néanmoins, leur complexité d'apprentissage croît linéairement avec le nombre de labels.

Random Forest of Predictive Clustering Trees (*RF-PCT*) (Kocev, 2011; Kocev et al., 2007) entraîne N arbres de décisions multi-label PCT. Un arbre PCT peut être construit à l'aide d'un algorithme standard d'induction descendante d'arbres de décision. L'heuristique utilisée pour sélectionner les attributs de coupure (tests) est la réduction de la variance associée au partitionnement des instances. Pour une tâche de classification, le critère de coupure correspond à la maximisation de la moyenne des Gini obtenus pour les labels. L'entropie ou le gain d'information peuvent aussi être utilisés. Les vecteurs prototypes dans les feuilles contiennent le vote majoritaire pour chaque label. Pour une tâche de régression, le critère de coupure correspond à la réduction de la somme des variances des labels. Les feuilles sont associées avec des vecteurs réels contenant le vote moyen pour chaque label. Pour introduire de la diversité, les arbres sont appris sur des sous-échantillons (crées de manière répétitive avec remise) de l'ensemble d'apprentissage (i.e. bagging (Breiman, 1996)) et en sélectionnant des sous-ensembles d'attributs aléatoires m' pour la découpe des nœuds où $m' << m$. Pour une nouvelle instance,

les prédictions de tous les arbres sont combinées à l'aide d'une approche de vote : prédictions majoritaires ou distribution des probabilités pour une classification et la moyenne des prédictions pour une régression. RF-PCT a une complexité de calcul de $\mathcal{O}(N \; . \; q \; . \; n \; . \; m' \; . \; log(n))$ pour l'apprentissage des modèles et $\mathcal{O}(N \; . \; log(n))$ pour la prédiction des labels d'un nouvel exemple. La force de cette approche réside dans la combinaison de classifieurs différents, de faibles performances et qui modélisent plus finement les corrélations entre les labels.

En général, toutes les approches d'apprentissage multi-label optimisent implicitement le critère Hamming Loss. Néanmoins, il y a des approches qui optimisent d'autres critères. Par exemple, LP et RAkEL optimisent implicitement le critère Exact match et CLR optimise explicitement le critère Ranking Loss. Ces critères sont définis dans la section suivante 4.

4 Évaluation multi-label

Pour évaluer la performance prédictive d'un classifieur multi-label, il existe un grand nombre de critères différents dans la littérature (Madjarov et al., 2012; Zhang and Zhou, 2013). La variété des critères d'évaluation est nécessaire pour fournir un aperçu global sur la performance prédictive de chaque classifieur. Ces critères peuvent être classés en deux familles :

1. **Critères d'évaluation de la classification** : ils permettent de comparer deux vecteurs binaires en fonction d'un critère spécifique. La plupart de ces critères sont utilisés dans l'évaluation binaire mono-label et ont été naturellement généralisés au cas multi-label. Ces critères se décomposent aussi en deux catégories :

   (a) **Critères d'évaluation de la classification par exemple** : ils évaluent, pour chaque exemple test $x_i \in \mathcal{S}$, la différence entre son vecteur des vrais labels y_i et son vecteur de prédictions $\hat{y}_i$. Ils calculent ensuite la moyenne sur tous les exemples de l'ensemble de test $\mathcal{S}$.

Méthode	**Idée principale**	**Modélisation des corrélations**	**Complexité [Train/Test]**	**Critère optimisé**
BR	apprend q classifieurs binaires	/	$\mathcal{O}(q \,.\, h_B(n,m))$ / $\mathcal{O}(q \,.\, h'_B(m))$	Hamming Loss
CC	apprend q classifieurs binaires dans une chaîne	$\star\star$	$\mathcal{O}(q \,.\, h_B(n,m+q))$ / $\mathcal{O}(q \,.\, h'_B(m+q))$	Hamming Loss
LP	apprend un seul classifieur multi-classes	$\star\star$	$\mathcal{O}(h_M(n,m,2^q))$ / $\mathcal{O}(h'_M(m,2^q))$	Exact match
CLR	apprend $\frac{q(q+1)}{2}$ classifieurs binaires	$\star$	$\mathcal{O}(q^2 \,.\, h_B(n,m))$ / $\mathcal{O}(q^2 \,.\, h'_B(m))$	Ranking Loss
HOMER	apprend une hiérarchie de classifieurs multi-classes	$\star\star$	$\mathcal{O}(C(q) \,+\, d \,.\, h_M(n,m,2^{q/k}))$ / $\mathcal{O}(log_k(q) \,.\, h'_M(m,2^{q/k}))$	Exact match
ML-kNN	combine kNN avec une inférence bayésienne	/	$\mathcal{O}(n^2 \,.\, m \,+\, q \,.\, n \,.\, k)$ / $\mathcal{O}(n \,.\, m \,+\, q \,.\, k)$	Hamming Loss
IBLR_ML	combine kNN avec une régression logistique	$\star\star$	$\mathcal{O}(n^2 \,.\, m \,+\, q \,.\, n \,.\, k)$ / $\mathcal{O}(n \,.\, m \,+\, q \,.\, k)$	Hamming Loss
RAkEL	apprend N classifieurs multi-classes	$\star\star$	$\mathcal{O}(N \,.\, h_M(n,m,2^k))$ / $\mathcal{O}(N \,.\, h'_M(m,2^k))$	Exact match
EBR	apprend N classifieurs BR	/	$\mathcal{O}(N \,.\, q \,.\, h_B(n,m))$ / $\mathcal{O}(N \,.\, q \,.\, h'_B(m))$	Hamming Loss
ECC	apprend N classifieurs CC	$\star\star$	$\mathcal{O}(N \,.\, q \,.\, h_B(n,m+q))$ / $\mathcal{O}(N \,.\, q \,.\, h'_B(m+q))$	Hamming Loss
RF-PCT	apprend N arbres de décisions multi-label	$\star\star$	$\mathcal{O}(N \,.\, q \,.\, n \,.\, m' \,.\, log(n))$ / $\mathcal{O}(N \,.\, log(n))$	Hamming Loss

TABLE 6 – Résumé des méthodes d'apprentissage multi-label représentatives de chaque famille.

(b) **Critères d'évaluation de la classification par label** : tout critère utilisé traditionnellement pour évaluer la performance prédictive d'un classifieur binaire peut être appliqué ici pour chaque label λ_i. Pour combiner les performances obtenues pour les différents labels, deux types de moyennes peuvent être utilisées selon le contexte d'application : la micro-moyenne ou la macro-moyenne.

2. **Critères d'évaluation du ranking** : ils permettent d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, la différence entre le vrai ranking y_i ($dom(y_i) \in \{0,1\}^q$) et le ranking prédit $\hat{y}_i$ ($dom(y_i) \in [0;1]^q$).

4.1 Critères d'évaluation de la classification

4.1.1 Critères d'évaluation de la classification par exemple

Accuracy : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, la similarité de *Jaccard* entre le vecteur des vrais labels y_i et le vecteur des labels prédits $\hat{y}_i$. Il est défini par :

$$Accuracy = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{|\hat{y_i}^+ \cap y_i^+|}{|y_i^+ \cup \hat{y_i}^+|}$$

Précision : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, la proportion des labels correctement prédits sur l'ensemble des labels prédits positifs, i.e. le taux de bonnes prédictions. Il est défini par :

$$Précision = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{\left|\hat{y_i}^+ \cap y_i^+\right|}{\left|\hat{y_i}^+\right|}$$

Rappel : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, la proportion des labels correctement prédits sur l'ensemble des vrais labels y_i. Il est défini par :

$$Rappel = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{\left|\hat{y_i}^+ \cap y_i^+\right|}{\left|y_i^+\right|}$$

F-mesure : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, la moyenne harmonique entre les critères de rappel et de précision. Il est défini par :

$$F_1 - score = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{2 \times |y_i^+ \cap \hat{y_i}^+|}{|y_i^+| + |\hat{y_i}^+|}$$

Hamming Loss : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, le nombre de labels mal classés. Il est défini par :

$$Hamming\text{-}Loss = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} |y_i \triangle \hat{y}_i|$$

où $\triangle$ est la différence symétrique entre l'ensemble des vrais labels y_i et l'ensemble des prédictions $\hat{y}_i$.

Exact match : il permet de vérifier, pour chaque exemple test $x_i \in \mathcal{S}$, si le vecteur des labels prédits $\hat{y}_i$ est identique au vecteur des vrais labels y_i. Il est souvent considéré comme un critère d'évaluation très strict parce qu'il punit sévèrement les prédictions du classifieur. Il est défini par :

$$Exact\ match = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} I[y_i = \hat{y}_i]$$

où I(vrai)=1 and I(faux)=0.

Tous ces critères sont définis dans l'intervalle [0..1] et leurs plus grandes valeurs indiquent les meilleures performances sauf pour le Hamming Loss où la plus petite valeur indique la meilleure performance.

4.1.2 Critères d'évaluation de la classification par label

C-macro : il utilise un critère d'évaluation de classification binaire C pour évaluer les prédictions obtenues pour chaque label λ_i. Il calcule ensuite la moyenne pour tous les q labels tels que tous les labels possèdent le même poids.

$$C_{macro} = \frac{1}{q} \sum_{i=1}^{q} C(\mathcal{S}, \lambda_i)$$

Exemple : Précision-macro

$$Précision\ macro = \frac{1}{q} \sum_{i=1}^{q} \frac{VP_i}{VP_i + FP_i}$$

C-Micro : il calcule, pour chaque label λ_i, la somme des exemples vrais positifs (VP_i), faux positifs (FP_i), vrais négatifs (VN_i) et faux négatifs (FN_i). Il calcule ensuite leurs moyennes sur l'ensemble des labels en considérant que chaque label a une contribution proportionnelle à son nombre d'exemples. En effet, un label dense compte davantage qu'un label rare.

$$C_{micro} = C\left(\sum_{i=1}^{q} VP_i, \sum_{i=1}^{q} FP_i, \sum_{i=1}^{q} VN_i, \sum_{i=1}^{q} FN_i\right)$$

Exemple : Précision-micro

$$\textit{Précision micro} = \frac{\frac{1}{q}\sum_{i=1}^{q} VP_i}{\frac{1}{q}\sum_{i=1}^{q} VP_i + \frac{1}{q}\sum_{i=1}^{q} FP_i}$$

4.2 Critères d'évaluation du ranking

On considère une fonction de tri r_i qui ordonne les labels de x_i dans un ordre descendant en fonction de leurs prédictions $\hat{y}_i$: $r_i(\lambda_a) = k, k \in \{1, 2, \ldots, q\}$, si $\hat{y}_i^a$ est la k^{e} grande valeur dans le vecteur $\hat{y}_i$.

Ranking Loss : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, le nombre de fois où un vrai label négatif est classé avant un vrai label positif. Il est défini par :

$$RL = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{|y_i^+| \times |y_i^-|}|(\lambda_a, \lambda_b) \in y_i^+ \times y_i^- : r_i(\lambda_b) < r_i(\lambda_a)|$$

One error : il permet de vérifier, pour chaque exemple test $x_i \in \mathcal{S}$, si le label prédit en tête de classement est un vrai positif. Il est défini par :

$$One - Error = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} I[(\lambda_j^* = \underset{\lambda_j \in y_i^+}{\operatorname{argmax}}\, \hat{y}_i^j) \notin y_i^+]$$

où I(vrai) = 1 and I(faux)=0.

Coverage : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, le nombre minimum de décalages nécessaires pour qu'un label positif arrive en tête de classement. Il est défini par :

$$Coverage = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\max_{\lambda_j \in y_i^+} r_i(\lambda_j) - 1$$

Average precision : il permet d'évaluer, pour chaque exemple test $x_i \in \mathcal{S}$, le nombre de

labels positifs $\lambda_k \in y_i^+$ se classant avant chaque label positif $\lambda_j \in y_i^+$. Il est défini par :

$$Average\ precision = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{|y_i^+|}\sum_{\lambda_j \in y_i^+}\frac{|w_i|}{r_i(\lambda_j)}$$

où $w_i = \{\lambda_k | r_i^k \leq r_i^j, \lambda_k \in y_i^+\}$.

Tous ces critères sont définis dans l'intervalle [0..1] sauf Coverage qui est défini dans l'intervalle $[0..q-1]$. Leurs plus petites valeurs indiquent les meilleures performances à l'exception de Average precision où la plus grande valeur indique la meilleure performance.

5 Données multi-label

Pour évaluer les performances prédictives ainsi que la vitesse de calcul des algorithmes d'apprentissage multi-label, il existe plusieurs jeux de données de référence. Nous présentons dans le tableau 7 17 jeux de données très utilisés dans la littérature de caractéristiques différentes et issus de divers domaines d'application (audio, biologie, musique, texte, image). Ces jeux de données sont classés en fonction de leur difficulté ($|\mathcal{F}| \times |\mathcal{L}|$). Les statistiques complémentaires données dans ce tableau montrent que ces jeux de données simulent un grand nombre de situations. En effet, leur nombre d'attributs varie de 71 à 49060, et leur nombre de labels varie de 6 à 101.

Emotions (Trohidis et al., 2008) est un petit jeu de données qui décrit des morceaux de musique par 71 attributs numériques. Ils peuvent être étiquetés par 6 émotions possibles : sad-lonely, angry-aggressive, amazed-surprised, relaxing-calm, quiet-still, et happy-pleased. ***Yeast*** (Elisseeff and Weston, 2001) est un jeu de données biologiques où les gènes sont décrits par 103 attributs numériques. Ils peuvent être associées avec 14 fonctions biologiques. ***Scene*** (Boutell et al., 2004) est un jeu de données où des images sont décrites par 294 attributs numériques. Elles peuvent être annotées avec 6 concepts au maximum : beach, sunset, field, fall-foliage, mountain, et urban. ***Birds*** (Briggs et al., 2013) est un petit jeu de données où des enregistrements audio de 10-secondes de sons d'oiseaux sont décrits par 260 attributs numériques. Ils peuvent être étiquetés avec 19 espèces d'oiseaux au maximum. ***Slashdot*** (Read et al., 2011) est un jeu

de données creux où les documents sont définis par 1079 attributs binaires. Ils peuvent être associés avec 20 catégories (e.g. linux, technology, science). ***IMDB*** (Read et al., 2011) est aussi un jeu de données creux où les films sont définis par 1001 attributs binaires. Ils peuvent être étiquetés avec 28 genres (e.g. Romance, Comedy, Drama). ***Genbase*** (Diplaris et al., 2005) est un autre jeu de données microbiologiques où les gènes sont décrits par 1186 attributs binaires. Ils peuvent être associées avec 27 fonctions biologiques.

TMC (Srivastava and Zane-Ulman, 2005) est un jeu de données creux de rapports de préparation et de divergence de vols. Ces rapports sont décrits par 49060 attributs binaires et peuvent être associés avec 22 labels représentant les problèmes décrits. ***Arts***, ***Business***, ***Health*** et ***Computers*** sont des jeux de données textuelles creuses qui décrivent des pages web de Yahoo. Dans ces jeux de données, les valeurs minimales (maximales) de $|\mathcal{L}|$ et $|\mathcal{F}|$ sont 24 (30) et 21924 (34096). Les cinq derniers jeux de données sont des sous-ensembles **Reuters S1-Reuters S5** du jeu de données *Reuters* (Lewis et al., 2004). Ces données représentent des articles de presse décrits par 47236 (47229) attributs au maximum (minimum). Ils peuvent être étiquetés avec 101 catégories (e.g. agriculture, pêche).

5.1 Distribution des labels

En plus des caractéristiques de base telles que le nombre d'attributs et le nombre de labels, on peut utiliser en apprentissage multi-label des critères qui permettent de mesurer la distribution des labels dans les données. Ces critères permettent d'approfondir la description des données et d'aider à l'interprétation des performances des classifieurs.

Cardinalité des labels (LCard) est sans doute le critère de distribution des labels le plus populaire (Tsoumakas and Katakis, 2007). Il permet d'évaluer le nombre de labels associé en moyenne aux exemples dans un jeu de données $\mathcal{D}$.

$$LCard(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} |y_i^+|$$

Densité des labels (LDens) est un critère relatif à *LCard* mais qui prend en compte en compte la taille de l'espace de labels (Tsoumakas and Katakis, 2007). Il est défini par le ratio du nombre moyen de labels dans un jeu de données $\mathcal{D}$ sur le nombre de labels q :

$$LDens(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\frac{|y_i^+|}{q} = \frac{LC}{q}$$

Jeu	Domaine	$\|\mathcal{F}\|$	$\|\mathcal{D}\|$	$\|\mathcal{L}\|$	*LCard*	*PUniq*	*LDens*
Emotions	Musique	71	592	6	1.86	0.05	0.31
Yeast	Biologie	103	2417	14	4.24	0.08	0.30
Scene	Image	294	2407	6	1.07	0.006	0.18
Birds	Audio	260	645	19	1.01	0.21	0.05
Slashdot	Texte	1079	3782	20	1.18	0.04	0.05
IMDB	Texte	1001	120919	28	2.00	0.04	0.07
Genbase	Biologie	1186	662	27	1.25	0.05	0.05
Arts	Texte	23146	7485	24	1.66	0.18	0.07
Business	Texte	21925	11213	28	1.55	0.07	0.05
Health	Texte	30605	9205	25	1.63	0.11	0.06
Computers	Texte	34097	12443	30	1.44	0.10	0.05
TMC	Texte	49060	28596	22	2.16	0.23	0.1
Reuters S4	Texte	47229	6000	101	2.48	0.14	0,03
Reuters S5	Texte	47235	6000	101	2.64	0.16	0,03
Reuters S1	Texte	47236	6000	101	2.88	0.17	0,03
Reuters S2	Texte	47236	6000	101	2.63	0.16	0,03
Reuters S3	Texte	47236	6000	101	2.61	0.16	0,03

TABLE 7 – Description des jeux de données multi-label de la litterature ($|\mathcal{F}|$: nombre d'attributs, $|\mathcal{D}|$: nombre d'exemples, $|\mathcal{L}|$: nombre de labels, *LCard* : Cardinalité des labels, *PUniq* : Proportion des ensembles de labels uniques, *LDens* : Densité des labels)

Proportion des ensembles de labels uniques (PUniq) est un nouveau critère de mesure de distribution des labels qui a été récemment introduit par (Read, 2010). Il mesure la régularité ou l'uniformité de l'étiquetage des données. Plus précisément, il permet de calculer le ratio du nombre d'ensembles de labels y_i uniques sur le nombre d'exemples dans un jeu de données $\mathcal{D}$:

$$PUniq(D) = \frac{|y_i|\exists!\ x_i : (x_i, y_i) \in \mathcal{D}|}{n}$$

Dans ces jeux de données, les valeurs *LCard* sont généralement plus petites ou égales à 2.0 à l'exception de Yeast et Reuters où les exemples sont associés en moyenne avec plus de 3 labels. Il n'est pas étonnant que la cardinalité des labels soit faible dans les données textuelles ou multimedia où la plupart des exemples possèdent un seul label. En effet, l'étiquetage multi-label a été utilisé seulement pour éviter les ambiguïtés. Les valeurs de *LDens* sont très faibles pour la

grande majorité des données car l'étiquetage est souvent très creux sauf pour Emotions et Yeast où 30 % des labels est associée en moyenne à chaque exemple. Les faibles valeurs de *PUniq* indiquent que l'étiquetage est généralement régulier sauf pour Birds, IMDB et TMC où plus de 20 % des exemples possèdent des ensembles de labels uniques (i.e. étiquetage irrégulier).

6 Discussion

En vue de la nécessité de l'étiquetage multi-label dans les applications récentes, de nombreuses approches ont été développées. Pour fournir un aperçu global sur les performances des approches d'apprentissage multi-label existantes et ainsi guider le développement des futures approches, plusieurs études comparatives ont été effectuées ces dernières années. Ces études évaluent essentiellement les performances prédictives des classifieurs dans divers contextes d'application. L'une des premières études est celle de (Li et al., 2006) qui compare six classifieurs pour neuf critères d'évaluation sur deux jeux de données. Les deux classifieurs : BR et ML_ADTree[1] sont recommandés à l'issue de cette étude. L'étude de (Nasierding and Kouzani, 2012) compare sept classifieurs pour quatre critères d'évaluation sur huit jeux de données. Elle préconise trois classifieurs : TREMLC[2], MLkNN et BR. L'étude la plus extensive est celle de (Madjarov et al., 2012) où douze classifieurs ont été comparés pour seize critères d'évaluation sur onze jeux de données. À l'issue de cette étude, quatre classifieurs obtiennent les meilleures performances : RF-PCT, HOMER, BR et CC. L'étude la plus récente est celle de (Tawiah and Sheng, 2013) où six classifieurs ont été comparés pour cinq critères d'évaluation sur onze jeux de données. Elle recommande les quatre classifieurs MLkNN, RAkEL, CC et BR. En raison de son exhaustivité en termes de classifieurs, de critères d'évaluation et de jeux de test, l'étude de (Madjarov et al., 2012) constitue notre point de référence pour nos expérimentations présentées dans la suite du manuscrit.

[1] Multi-Label Alternating Decision Tree
[2] Triple-Random Ensemble Multi-Label Classification

4

Apprentissage multi-label avec contraintes d'interactivité

Sommaire

1 Introduction

L'objectif final de l'entreprise dans laquelle s'est déroulée la thèse est d'intégrer une approche d'apprentissage multi-label dans un système de classification interactif pour permettre aux utilisateurs d'étiqueter des exemples avec plusieurs labels subjectifs et ainsi exprimer des requêtes de recherche complexes sur les données. Le premier problème majeur commun à tous les développeurs d'un tel système est : quel classifieur multi-label devrions-nous choisir ? Comme mentionné en introduction, l'efficacité d'un système de classification interactif dépend aussi d'autres facteurs tels que le support de visualisation des données et les mécanismes d'interaction mais nous nous focalisons ici sur la composante d'apprentissage.

Nous présentons dans ce chapitre la première étude extensive des comportements des algorithmes d'apprentissage multi-label dans un environnement interactif. Plus précisément, nous considérons ici les deux contraintes d'interactivité majeurs suivantes : l'apprentissage à partir de quelques exemples en un temps limité. Et nous étudions l'impact de ces contraintes sur les performances d'un grand ensemble de douze algorithmes représentatifs des trois grandes familles des approches multi-label : cinq méthodes d'apprentissage par transformation, deux méthodes d'apprentissage par adaptation et cinq méthodes ensemble. Les performances des classifieurs sont évaluées de manière « objective » avec un protocole expérimental indépendant de toute application cible sur un ensemble de douze jeux de données multi-label de tailles différentes provenant de divers domaines (musique, audio, image, biologie et texte). Nous nous limitons à l'évaluation des performances prédictives et des temps de calcul des classifieurs pendant que le nombre d'exemples d'apprentissage augmente régulièrement et nous nous focalisons sur le début de la tâche de classification où seulement quelques exemples sont disponibles.

Dans la littérature, plusieurs critères ont été proposés pour évaluer les performances prédictives des classifieurs (par exemple Tsoumakas et al. (2010); Zhang and Zhou (2013)) mais il

n'y a pas de consensus dans la sélection des « meilleurs » critères. Nous nous focalisons ici sur les propriétés essentielles dans une application de classification multi-label : tri des labels d'un exemple par pertinence, tri des exemples d'un label par pertinence, classification des labels d'un exemple. Ces propriétés désirées nous ont conduites à sélectionner quatre critères d'évaluation de la littérature (Ranking Loss, Accuracy, F-mesure et Balanced Error Rate (BER)) et de proposer une adaptation du Ranking Loss pour évaluer la qualité du tri des exemples d'un label (macro-averaged Ranking Loss). Pour la consolidation de nos conclusions, nous considérons cinq mesures supplémentaires classiques : Coverage, One Error, Average Precision, Hamming Loss et Exact match. La vitesse de calcul des classifieurs est mesurée à la fois par les temps observés dans les expérimentations et les complexités de calcul théoriques requis pour l'apprentissage des modèles et pour la prédiction des labels d'un nouvel exemple.

Les résultats expérimentaux montrent que les classifieurs ensemble RF-PCT (Random Forest of Predictive Clustering Trees, Kocev (2011)) et EBR (Ensemble of Binary Relevance, Read et al. (2011)) obtiennent les meilleures performances pour toutes les tailles d'ensemble d'apprentissage. Ils sont suivis par CLR (Calibrated Label Ranking, Fürnkranz et al. (2008)) et ML*k*NN (Multi-label *k*NN, Zhang and Zhou (2007)). En terme de vitesse de calcul, RF-PCT obtient également de bonnes performances et il peut être ainsi considéré comme le classifieur multi-label le plus efficace dans les différents contextes d'apprentissage.

2 Classification multi-label

Pour notre étude comparative, nous avons sélectionné douze approches très populaires et représentatives des trois familles des algorithmes de classification multi-label présentés dans le chapitre précédent. Parmi les classifieurs choisis, nous retrouvons les classifieurs recommandés à l'issue de l'étude expérimentale extensive de Madjarov et al. (2012). Ces algorithmes de classification sont décrits succintement dans la section 3 du chapitre 3. Leurs implémentations sont disponibles dans les librairies d'apprentissage multi-label suivantes qui sont largement utilisées par la communauté : *MeKA*[1], *MULAN*[2] et *CLUS*[3].

[1] meka.sourceforge.net
[2] mulan.sourceforge.net
[3] dtai.cs.kuleuven.be/clus

De la première famille d'approches (apprentissage par transformation), nous avons sélectionné cinq classifieurs : BR (Schapire and Singer, 2000), CC (Read et al., 2011), LP (Tsoumakas and Katakis, 2007), CLR (Fürnkranz et al., 2008), HOMER (Tsoumakas et al., 2008) et un classifieur Baseline qui évalue la fréquence de chaque ensemble de labels dans l'ensemble d'apprentissage et prédit l'ensemble de labels le plus fréquent pour tout nouvel exemple. Les méthodes de cette famille d'approches permettent d'apprendre avec tout classifieur mono-label de l'état de l'art. Une étude comparative préconise les classifieurs SVM et l'arbre de décision C4.5 comme les classifieurs de base les plus efficaces (Read, 2010). Ici, en raison de la forte contrainte en temps de calcul dans notre cadre d'apprentissage interactif, nous avons sélectionné l'arbre de décision C4.5 pour sa faible complexité de calcul lui permettant de traiter avec aisance de grandes volumétries. Contrairement au SVM, il ne nécessite qu'un sous-ensemble d'attributs pour construire un modèle prédictif.

De la deuxième famille d'approches (apprentissage par adaptation), nous avons sélectionné deux classifieurs : ML-kNN (Zhang and Zhou, 2007) et IBLR_ML (Cheng and Hüllermeier, 2009) et de la troisième famille d'approches (apprentissage ensemble), nous avons sélectionné cinq classifieurs : les deux variantes de RAkEL (RAkEL$_1$ et RAkEL$_2$), EBR, ECC et RF-PCT.

Référence	#Classifieurs	#Critères	#Jeux	Recommandation
(Li et al., 2006)	6	9	2	BR et ML_ADTree
(Nasierding and Kouzani, 2012)	7	4	8	TREMLC, MLkNN et BR
(Madjarov et al., 2012)	12	16	11	RF-PCT, HOMER, BR et CC
(Tawiah and Sheng, 2013)	6	5	11	MLkNN, RAkEL, CC et BR

TABLE 1 – Les études comparatives les plus pertinentes des algorithmes d'apprentissage multi-label avec ML_ADTree : Multi-Label Alternating Decision Tree et TREMLC : Triple-Random Ensemble Multi-Label Classification.

2.1 Études comparatives des classifieurs multi-label

Comme nous l'avons souligné dans la conclusion du chapitre 3, il existe dans la littérature des études comparatives des algorithmes d'apprentissage multi-label mais elles ne considèrent pas les contraintes d'interactivité dans l'évaluation des classifieurs. Nous résumons les études expérimentales principales dans le tableau 1 où chaque étude est décrite par son nombre de classifieurs (*#Classifieurs*), nombre de jeux d'évaluations (*#Jeux*), nombre de critères d'éva-

luation des performances prédictives (*#Critères*) et l'ensemble des classifieurs recommandés (*Recommendation*).

3 Critères d'évaluation

Dans cette section, nous précisons les critères d'évaluation des performances des classifieurs pour les deux contraintes d'interactivité majeures : apprendre à généraliser à partir de peu d'exemples et, apprendre et prédire en temps limité.

3.1 Contrainte 1 : Apprendre à généraliser à partir de peu d'exemples

Demander à l'utilisateur de fournir un grand nombre d'exemples pour expliquer un concept cible est une tâche laborieuse qu'il faut lui éviter. Pour l'évaluation du pouvoir de généralisation des classifieurs à partir d'un nombre d'exemples limité, nous sélectionnons cinq critères complémentaires adaptés aux propriétés nécessaires dans les applications d'apprentissage multi-label : tri des labels d'un exemple par pertinence, tri des exemples d'un label par pertinence et classification des labels d'un exemple. Nous consolidons les résultats obtenus avec cinq critères additionnels qui sont classiquement utilisés dans la littérature. Le tableau 2 présente un résumé de ces critères d'évaluation que nous appelons *critères de qualité* dans la suite.

3.1.1 Propriété 1 : Tri des labels d'un exemple par pertinence

En pratique, les utilisateurs sont généralement intéressés par un tri des labels d'un nouvel exemple. Lorsqu'un exemple est sélectionné, le modèle d'apprentissage doit présenter ses labels positifs en tête de la liste des prédictions. Pour évaluer la capacité des classifieurs à bien ordonner les labels des exemples, nous sélectionnons le critère classique : *Ranking Loss* (*RL*) (voir définition en section 4 du chapitre 3).

3.1.2 Propriété 2 : Tri des exemples d'un label par pertinene

Les utilisateurs peuvent être également intéressés par un tri d'exemples pour un ou plusieurs labels. Lorsqu'un label ou une combinaison est sélectionné(e), le modèle d'apprentissage doit présenter ses exemples positifs en tête de la liste des prédictions. Pour évaluer la capacité des

classifieurs à bien ordonner les exemples des labels, nous avons adapté la définition du critère RL : nous proposons le critère macro-averaged Ranking-Loss (*macro-RL*) qui permet de mesurer pour chaque label le nombre de fois où des exemples positifs et négatifs sont inversement ordonnés. Comme RL, macro-RL est défini dans l'intervalle [0..1] et ses plus faibles valeurs indiquent les meilleures performances. Formellement, soient γ_i^+ et γ_i^- respectivement l'ensemble des exemples positifs et négatifs du label λ_i avec $|\gamma_i^+|$ + $|\gamma_i^-|$ = $|\mathcal{S}|$, et $\hat{\gamma}_i$ ($dom(\hat{\gamma}_i) \in [0..1]^{|\mathcal{S}|}$) le vecteur réel qui décrit les probabilités d'appartenance des exemples $x_i \in \mathcal{S}$ au label λ_i. On considère une fonction de tri r'_i qui ordonne chaque vecteur $\hat{\gamma}_i$ dans un ordre descendant : $r'_i(x_a) = k, k \in \{1, 2, \ldots, |\mathcal{S}|\}$, si $\hat{\gamma}_i^a$ est la k^e plus grande valeur parmi les valeur de $\hat{\gamma}_i$. Le *macro-RL* d'un classifieur sur un ensemble de test $\mathcal{S}$ est défini par :

$$macro - RL = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} \frac{1}{|\gamma_i^+| \times |\gamma_i^-|} |(x_a, x_b) \in \gamma_i^+ \times \gamma_i^- : r'_i(x_b) < r'_i(x_a)|$$

3.1.3 Propriété 3 : Classification des labels d'un exemple

Un tri des labels pour de nouveaux exemples est essentiel dans un système d'apprentissage multi-label mais une classification des labels peut être parfois désirée par l'utilisateur. Lorsque un exemple est sélectionné, le modèle d'apprentissage doit présenter uniquement ses labels positifs. Pour évaluer la capacité des classifieurs à bien classer les labels des exemples, nous sélectionnons trois critères : *Accuracy*, *F-mesure* et l'adaptation multi-label du Balanced Error Rate (*BER*) (voir définitions de l'Accuracy et du F-mesure en section 4 du chapitre 4). Dans Madjarov et al. (2012), les critères *Accuracy* et *F-mesure* permettent de bien discriminer les classifieurs et le critère BER est adapté lorsque les données d'évaluation sont déséquilibrées (e.g. *Slashdot* (Read et al., 2011)).

Le ***BER*** (Chen and Lin, 2006) a généralement été utilisé pour l'évaluation de prédictions mono-label mais nous l'adaptons ici pour évaluer des prédictions multi-label. Il permet d'évaluer le ratio des labels mal classés où on rappelle que VP_i, VN_i, FP_i and FN_i dénotent le nombre de labels respectivement vrais positifs, vrais négatifs, faux positifs et faux négatifs d'un exemple x_i. Il prend ses valeurs dans l'intervalle [0..1] et sa plus faible valeur indique la meilleure performance. Le *BER* d'un classifieur sur un ensemble de test $\mathcal{S}$ est défini par :

$$BER = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{2}\times\left(\frac{FP_i}{FP_i+TN_i}+\frac{FN_i}{FN_i+TP_i}\right)$$

3.1.4 Critères de qualité additionnels

Pour limiter le biais induit par les critères de qualité choisis, nous ajoutons cinq nouveaux critères classiques issus des deux familles de critères présentés dans le chapitre 3 (section 4) : (i) Coverage, One-error et Average precision et (ii) Hamming loss et Exact match. Les critères issus du groupe (i) (resp. groupe (ii)) contribuent à l'évaluation de la propriété 1 (Section 3.1.1) (resp. propriété 3 (Section 3.1.3)).

Critères de qualité	Min/Max	Intervalle	Ranking/Classification
Ranking Loss	Min	[0..1]	Ranking
Macro-averaged Ranking Loss	Min	[0..1]	Ranking
Accuracy	Max	[0..1]	Classification
F-mesure	Max	[0..1]	Classification
Balanced Error Rate	Min	[0..1]	Classification
Coverage	Min	$[0..q-1]$	Ranking
One error	Min	[0..1]	Ranking
Average precision	Max	[0..1]	Ranking
Hamming Loss	Min	[0..1]	Classification
Exact match	Max	[0..1]	Classification

TABLE 2 – Un résumé des critères sélectionnés pour l'évaluation de la qualité des prédictions.

3.2 Contrainte 2 : Apprendre et prédire en temps limité

Dans un environnement interactif, le temps de réponse d'un système d'apprentissage doit être faible : lorsque un utilisateur ajoute de nouveaux exemples, le modèle d'apprentissage doit rapidement mettre à jour son modèle et lui fournir des prédictions le plus vite possible. En Interaction Homme-Machine, les systèmes interactifs sont souvent contraints à fournir une réponse aux utilisateurs en moins de 100 ms (Dabrowski and Munson, 2001). À notre connaissance, dans ce contexte cette contrainte est très forte et nous la réduisons ici à quelques secondes. Pour évaluer la vitesse de calcul des classifieurs, nous mesurons pour chaque classifieur le nombre de secondes requises pour apprendre un modèle à partir de peu d'exemples et le nombre de secondes nécessaires pour prédire l'ensemble de labels d'un nouvel exemple. Nous sommes

conscients que ces temps de calcul mesurés dépendent évidemment de la qualité des implémentations de ces classifieurs, et que les résultats obtenus fournissent uniquement des tendances sur leurs complexités de calcul théoriques.

4 Cadre expérimental

Nous décrivons d'abord le protocole expérimental proposé pour évaluer les performances des classifieurs dans un contexte interactif simplifié. Il permet de comparer les classifieurs dans les mêmes conditions. Ensuite, nous précisons les paramètres choisis pour les différents classifieurs et les jeux de données utilisés.

4.1 Protocole expérimental

L'évaluation de l'efficacité des classifieurs dans un cadre interactif est une tâche difficile qui suscite de nombreuses questions. En raison de la nouveauté de ce domaine de recherche, il n'existe pas encore un protocole d'évaluation standard ou largement utilisé pour comparer les performances des classifieurs. Dans les travaux présentés dans la section 2 du chapitre 2, l'évaluation des systèmes est essentiellement basée sur des retours de quelques utilisateurs.

Pour tirer des conclusions plus générales qui permettent de guider la sélection de l'algorithme d'apprentissage lors du développement d'un système de classification interactive multi-label, nous évaluons les performances des algorithmes pour des petits ensembles d'apprentissage de tailles croissantes. L'objectif derrière ce protocole est de détecter les classifieurs qui sont en mesure de bien apprendre de manière « continue » avec des ensembles d'apprentissage de tailles très limitées en un temps raisonnable. Plus précisément, nous nous focalisons ici sur le début de la tâche de classification qui -comme nous l'avons indiqué précédemment- est une étape clé pour conserver l'intérêt de l'utilisateur dans le système. Nous supposons que seulement quelques exemples (2 à 64 exemples) sont disponibles.

Pour éviter les biais dans les comparaisons, tous les classifieurs sont entraînés avec les mêmes exemples d'apprentissage. Le principe du protocole expérimental proposé est le suivant. Chaque jeu de données est divisé en un petit ensemble d'apprentissage et un grand ensemble de test. De chaque ensemble d'apprentissage, des sous-ensembles d'apprentissage im-

briqués de petites tailles sont successivement créés. Chaque classifieur est entraîné avec tous les sous-ensembles d'apprentissage crées et ses performances sont évaluées pour chaque taille de sous-ensemble d'apprentissage sur le même ensemble de test. Ce processus permet de suivre précisément l'évolution des performances des classifieurs pendant que la taille de l'ensemble d'apprentissage croît. Pour assurer des résultats statistiquement significatifs, il est répété plusieurs fois, comme décrit précisément ci-dessous.

1. Diviser chaque ensemble de données $\mathcal{D}$ en 5 sous-ensembles disjoints : Un sous-ensemble est utilisé pour l'apprentissage ($\mathcal{T}$ = 20% de $\mathcal{D}$) et les 4 autres pour le test ($\mathcal{S}$ = 80% de $\mathcal{D}$). Au total, 5 ensembles $\mathcal{T}_i$ (i = 1 à 5) sont utilisés pour l'apprentissage et 5 pour le test $\mathcal{S}_i$ (i = 1 à 5) (i.e. 5 validation-croisée).

2. À partir de chaque ensemble d'apprentissage $\mathcal{T}_{i\ (1\leq i\leq 5)}$, extraire s ensembles de p sous-ensembles imbriqués de taille 2^1, 2^2, ... jusqu'à 2^p.

3. Associer chaque classifieur avec les $5 \times s \times p$ sous-ensembles d'apprentissage ($5 \times s$ sous-ensembles d'apprentissage pour chaque taille).

4. Pour chaque taille de sous-ensemble d'apprentissage et pour chaque critère, évaluer sa performance moyenne sur les 5 ensembles de test.

Dans toutes les expérimentations, le paramètre s a été fixé à 10 et p à 6 ; ce qui correspond à 300 évaluations « apprentissage-test » pour la 5 validation croisée. Le seuil ($p = 6$) est conforme avec des expérimentations réelles. De notre expérience, nous supposons qu'un utilisateur ne va pas annoter plus de 64 exemples seul.

Cette approche d'apprentissage en-ligne est adaptée au début de l'interaction lorsque le modèle n'est pas encore stable et peut très probablement changer (i.e. dérive de concept). En pratique, les utilisateurs définissent leurs labels préférés en temps réel au moment où ils interagissent avec les données et le système d'apprentissage. En effet, ils n'ont pas en général une idée claire sur les concepts désirés -qu'ils ont en tête- avant que l'interaction commence (i.e. flexibilité du concept (Amershi, 2011)).

4.2 Paramètres des classifieurs et jeux de données

Pour paramétrer les classifieurs sélectionnés, nous suivons les recommandations de la littérature sauf pour les trois approches suivantes : HOMER, MLkNN et IBLR_ML. Afin de sélectionner le nombre de clusters approprié pour HOMER, plusieurs valeurs sont testées dans la littérature (Madjarov et al., 2012). Pour éviter donc un biais dans la comparaison des classifieurs, nous testons ici seulement la valeur par défaut (i.e. 3) dans la librairie. Pour MLkNN et IBLR_ML, nous fixons le nombre de voisins à 1 parce que le nombre d'exemples d'apprentissage est très faible. Les paramètres des classifieurs sont précisés dans le tableau 3.

Classifieur	paramètres	Libraire	Référence
LP	/	MeKA	/
CC	/	MeKA	/
ECC	$N = 10 \times q$	MeKA	(Read et al., 2011)
BR	/	MeKA	/
EBR	$N = 10 \times q$	MeKA	(Read et al., 2011)
RAkEL$_1$	$N = 10, k = q/2$	MULAN	(Tsoumakas and Vlahavas, 2007)
RAkEL$_2$	$N = 2 \times q, k = 3$	MULAN	(Tsoumakas and Vlahavas, 2007)
MLkNN	$k = 1$	MULAN	/
IBLR_ML	$k = 1$	MULAN	/
HOMER	$k = 3$	MULAN	par défaut
CLR	/	MULAN	/
RF-PCT	$N = 100$, $m' = 0.1 \times \lvert\mathcal{F}\rvert + 1$	CLUS	(Kocev, 2011)

TABLE 3 – Les paramètres en entrée de chaque classifieur multi-label où q est le nombre de labels, N est le nombre de classifieurs utilisés par les méthodes ensemble, m' est le nombre d'attributs sélectionnés pour chaque nœud par RF-PCT, et k peut représenter soit le nombre de clusters de labels, la taille des sous-ensembles de labels ou le nombre de voisins respectivement pour HOMER, RAkEL et les approches à base de kNN (i.e. MLkNN et IBLR_ML).

Pour l'évaluation, nous considérons douze jeux de données décrits dans le chapitre 3 (Tableau 7) : Emotions, Yeast, Scene, Birds, Slashdot, IMDB, Genbase, Arts, Business, Health, Computers et TMC. Dans un cadre de classification interactive, le modèle d'apprentissage est souvent appliqué sur une partie des données non-étiquetées seulement car il n'est pas possible de prédire les labels de tous les exemples en un temps raisonnable. Donc, pour obtenir des estimations des temps d'apprentissage et de prédiction moyens des classifieurs, nous conservons ici uniquement des échantillons de 1000 exemples sélectionnés aléatoirement à partir des données

d'évaluation originales sauf pour Emotions, Birds et Genbase qui sont à l'origine de petites tailles.

5 Résultats expérimentaux I : apprentissage à partir de peu d'exemples

Les tableaux 4-8 présentent les résultats obtenus pour les critères de qualité définis dans la section 3.1. La performance moyenne de chaque classifieur sur tous les jeux de données est donnée pour chaque taille d'ensemble d'apprentissage (de 2 à 64 exemples). Les performances prédictives des classifieurs s'améliorent considérablement lorsque le nombre exemples d'apprentissage augmente, en particulier lorsque la taille de l'ensemble d'apprentissage est supérieure à 8 -sauf pour le macro-RL. Notons qu'il n'est pas surprenant que tous les classifieurs obtiennent des performances très proches pour les ensembles d'apprentissage de taille 2 ; le nombre d'exemples fournis est simplement insuffisant pour leur permettre de bien apprendre. Par ailleurs, les différences entre les performances des classifieurs augmentent linéairement avec le nombre d'exemples. Ces différences sont confirmées par un test statistique de Friedman (avec un niveau de significativité de 5%) pour tous les critères hormis le BER.

Pour les critères de qualités majeurs (RL et macro-RL), la méthode ensemble RF-PCT surpasse tous les autres classifieurs pour toutes les tailles d'ensemble d'apprentissage (Tableaux 4 et 5). Elle est suivie de près par EBR, CLR et MLkNN qui obtiennent des performances très similaires et qui appartiennent aux trois familles d'approches d'apprentissage multi-label. Le test post-hoc de Nemenyi ne révèle aucune différence statistique entre ces classifieurs : ils sont tous au même niveau de performance. Les résultats détaillés obtenus pour trois tailles représentatives des ensembles d'apprentissage (4, 16 et 64 exemples) montrent que les meilleurs classifieurs restent toujours les mêmes quel que soit le jeu de données (voir l'Annexe 1 - Tableaux 1 - 6) ; cela est confirmé par un test statistique Friedman.

En considérant les autres critères de qualité (Accuracy, F-mesure et BER), le podium reste le même et RF-PCT demeure en tête de classement (Tableaux 6 - 8). Rappelons-nous que RF-PCT et CLR étaient déjà parmi les meilleurs classifieurs pour les critères RL, Accuracy et F-mesure dans l'étude comparative de Madjarov et al. (2012) qui ne prend en compte les contraintes

d'interactivité. Dans cette étude antérieure, EBR n'a pas été évaluée et MLkNN a obtenu des mauvais résultats pour l'Accuracy et le F-mesure. Les digrammes critiques[4] obtenus à l'issue des tests statistiques (Friedman et Nemenyi post-hoc) pour les principaux critères de qualité sont donnés en Figures 1-3.

La principale conclusion est que les capacités d'apprentissage des méthodes ensemble RF-PCT et EBR restent bonnes quelle que soit la taille de l'ensemble de l'apprentissage. Des détails complémentaires sont donnés ci-dessous pour chaque critère.

5.1 Ranking Loss (RL)

Pour toutes les tailles d'ensembles d'apprentissage, RF-PCT obtient les meilleures performances. Elle est suivie par CLR, EBR et MLkNN qui obtiennent des performances très similaires avec un très léger avantage à CLR qui optimise intrinsèquement ce critère ; cela est confirmé par un test statistique de Friedman. En revanche, BR et CC, qui étaient classifieurs favoris dans l'étude de Madjarov et al. (2012), perdent leur efficacité pour des ensembles d'apprentissage de petites tailles. En outre, lorsque le nombre d'exemples d'apprentissage augmente, tous les classifieurs ont réussi à améliorer la qualité de leurs tri des labels -sauf HOMER qui a également déjà fourni de mauvais résultats pour des ensembles d'apprentissage de grandes tailles (Madjarov et al., 2012). Il semble que HOMER soit plus adapté à l'apprentissage à partir de données où le nombre de labels est important (des centaines et plus) (Tsoumakas et al., 2008).

Quelques remarques peuvent aussi être ajoutées pour les différentes familles de classifieurs. Pour les méthodes d'apprentissage ensemble, EBR est meilleure que BR, ECC est légèrement meilleure que CC et RAkEL$_1$ surpasse légèrement RAkEL$_2$. Pour les méthodes d'apprentissage par transformation, CC ne surpasse pas BR et ils obtiennent des performances très proches comme dans Madjarov et al. (2012). Pour les méthodes d'apprentissage par adaptation, MLkNN surpasse IBLR_ML ; ce qui diffère des résultats précédents obtenus par (Cheng and Hüllermeier, 2009) pour des ensembles d'apprentissage de grandes tailles.

[4]Un diagramme critique représente une projection des rangs moyens des classifieurs sur un axe énuméré. Les classifieurs sont ordonnés de gauche (le meilleur) à droite (le pire) et une ligne épaisse relie les classifieurs dont les rangs moyens ne diffèrent pas significativement (pour le niveau de significativité de 5%).

	2	4	8	16	32	64
Baseline	$0,39 \pm 0,13$	$0,37 \pm 0,14$	$0,35 \pm 0,14$	$0,34 \pm 0,13$	$0,34 \pm 0,13$	$0,33 \pm 0,13$
LP	**0,34 ± 0,13**	$0,34 \pm 0,11$	$0,32 \pm 0,11$	$0,30 \pm 0,11$	$0,28 \pm 0,11$	$0,26 \pm 0,11$
CC	**0,34 ± 0,13**	$0,33 \pm 0,12$	$0,31 \pm 0,11$	$0,29 \pm 0,10$	$0,26 \pm 0,11$	$0,24 \pm 0,11$
RAkEL1	$0,35 \pm 0,13$	$0,34 \pm 0,12$	$0,32 \pm 0,12$	$0,29 \pm 0,11$	$0,26 \pm 0,11$	$0,24 \pm 0,11$
RAkEL2	$0,35 \pm 0,12$	$0,36 \pm 0,13$	$0,34 \pm 0,13$	$0,31 \pm 0,13$	$0,28 \pm 0,13$	$0,25 \pm 0,13$
MLkNN	**0,34 ± 0,13**	$0,32 \pm 0,13$	$0,28 \pm 0,11$	$0,24 \pm 0,09$	$0,20 \pm 0,08$	$0,18 \pm 0,08$
HOMER	$0,43 \pm 0,10$	$0,41 \pm 0,11$	$0,39 \pm 0,12$	$0,38 \pm 0,13$	$0,37 \pm 0,14$	$0,35 \pm 0,15$
IBLR-ML	**0,34 ± 0,13**	$0,32 \pm 0,13$	$0,30 \pm 0,11$	$0,26 \pm 0,10$	$0,23 \pm 0,09$	$0,20 \pm 0,08$
CLR	**0,34 ± 0,13**	**0,31 ± 0,13**	$0,28 \pm 0,12$	$0,24 \pm 0,10$	$0,20 \pm 0,07$	$0,17 \pm 0,07$
ECC	$0,37 \pm 0,13$	$0,33 \pm 0,13$	$0,31 \pm 0,13$	$0,28 \pm 0,13$	$0,25 \pm 0,12$	$0,22 \pm 0,11$
BR	**0,34 ± 0,13**	$0,33 \pm 0,12$	$0,31 \pm 0,11$	$0,28 \pm 0,10$	$0,25 \pm 0,10$	$0,23 \pm 0,10$
EBR	**0,34 ± 0,13**	$0,32 \pm 0,12$	$0,28 \pm 0,11$	$0,24 \pm 0,09$	$0,20 \pm 0,07$	$0,17 \pm 0,07$
RF-PCT	**0,34 ± 0,13**	**0,31 ± 0,12**	**0,27 ± 0,11**	**0,23 ± 0,08**	**0,18 ± 0,06**	**0,15 ± 0,06**

TABLE 4 – Les performances moyennes de chaque classifieur sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Ranking Loss (RL).

	2	4	8	16	32	64
Baseline	**0,49 ± 0,02**	$0,49 \pm 0,02$	$0,49 \pm 0,02$	$0,49 \pm 0,02$	$0,49 \pm 0,02$	$0,49 \pm 0,02$
LP	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,03$	$0,45 \pm 0,04$	$0,43 \pm 0,07$	$0,41 \pm 0,08$
CC	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,02$	$0,46 \pm 0,03$	$0,44 \pm 0,05$	$0,43 \pm 0,07$
RAkEL1	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,03$	$0,44 \pm 0,05$	$0,42 \pm 0,07$	$0,39 \pm 0,09$
RAkEL2	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,03$	$0,44 \pm 0,05$	$0,42 \pm 0,08$	$0,39 \pm 0,09$
MLkNN	**0,49 ± 0,02**	$0,49 \pm 0,02$	$0,47 \pm 0,03$	$0,45 \pm 0,06$	$0,43 \pm 0,08$	$0,42 \pm 0,10$
HOMER	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,03$	$0,45 \pm 0,05$	$0,44 \pm 0,07$	$0,43 \pm 0,08$
IBLR-ML	**0,49 ± 0,02**	$0,49 \pm 0,02$	$0,48 \pm 0,02$	$0,47 \pm 0,04$	$0,44 \pm 0,07$	$0,42 \pm 0,11$
CLR	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,03$	$0,45 \pm 0,05$	$0,42 \pm 0,08$	$0,38 \pm 0,10$
ECC	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,46 \pm 0,04$	$0,43 \pm 0,07$	$0,40 \pm 0,09$	$0,38 \pm 0,10$
BR	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,47 \pm 0,02$	$0,46 \pm 0,04$	$0,44 \pm 0,06$	$0,42 \pm 0,07$
EBR	**0,49 ± 0,02**	$0,48 \pm 0,02$	$0,46 \pm 0,04$	$0,43 \pm 0,07$	$0,40 \pm 0,09$	$0,38 \pm 0,10$
RF-PCT	**0,49 ± 0,02**	**0,46 ± 0,03**	**0,43 ± 0,06**	**0,39 ± 0,09**	**0,35 ± 0,11**	**0,32 ± 0,12**

TABLE 5 – Les performances moyennes de chaque classifieur sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère macro-averaged Ranking Loss (macro-RL).

5.2 Macro-averaged Ranking Loss (macro-RL)

À notre connaissance, la capacité des classifieurs multi-label à bien trier les exemples des labels n'a pas encore été évaluée dans la littérature. Le tableau 5 montre que la méthode ensemble RF-PCT est la meilleure pour toutes les tailles d'ensembles d'apprentissage -sauf pour la plus

petite taille où tous les classifieurs n'arrivent pas à apprendre et obtiennent exactement la même performance. Elle est suivie par les méthodes ensemble ECC et EBR ; ce résultat est confirmé par un test statistique de Friedman. Cependant, lorsque la taille de l'ensemble d'apprentissage augmente, tous les classifieurs ont du mal à améliorer leurs prédictions à l'exception de RF-PCT. Il semble que les algorithmes d'apprentissage multi-label soient naturellement conçus pour apprendre à ordonner des labels mais pas des exemples.

5.3 Accuracy, F-mesure et BER

Comme les critères de ranking sont prioritaires dans un cadre de classification interactive multi-label, nous retenons ici uniquement les meilleurs classifieurs selon RL et macro-RL pour approfondir l'analyse de leurs comportements pour d'autres critères de qualité (Accuracy, F-mesure et BER). Pour l'Accuracy et le F-mesure (tableaux 6 et 7), RF-PCT est toujours la plus efficace pour toutes les tailles d'ensembles d'apprentissage sauf la plus petite où CLR est légèrement plus efficace. Les tests post-hoc de Nemenyi confirment que RF-PCT est significativement meilleure que MLkNN pour le F-mesure et l'Accuracy. Cependant, pour le critère BER (tableau 8), les classifieurs obtiennent des performances très similaires pour toutes les tailles d'ensembles d'apprentissage ; cela est confirmé par un test statistique de Friedman.

	2	4	8	16	32	64
MLkNN	$0,17 \pm 0,10$	$0,20 \pm 0,10$	$0,24 \pm 0,11$	$0,28 \pm 0,13$	$0,31 \pm 0,13$	$0,33 \pm 0,16$
CLR	**0,19 ± 0,10**	$0,21 \pm 0,11$	$0,24 \pm 0,12$	$0,26 \pm 0,12$	$0,30 \pm 0,12$	$0,34 \pm 0,13$
EBR	$0,18 \pm 0,10$	$0,20 \pm 0,10$	$0,26 \pm 0,11$	$0,30 \pm 0,13$	$0,33 \pm 0,14$	$0,38 \pm 0,16$
RF-PCT	$0,18 \pm 0,10$	**0,24 ± 0,12**	**0,28 ± 0,13**	**0,33 ± 0,12**	**0,39 ± 0,13**	**0,44 ± 0,17**

TABLE 6 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Accuracy.

	2	4	8	16	32	64
MLkNN	$0,22 \pm 0,14$	$0,27 \pm 0,13$	$0,32 \pm 0,14$	$0,36 \pm 0,15$	$0,40 \pm 0,16$	$0,42 \pm 0,18$
CLR	**0,26 ± 0,14**	$0,28 \pm 0,15$	$0,31 \pm 0,15$	$0,35 \pm 0,15$	$0,40 \pm 0,15$	$0,43 \pm 0,15$
EBR	$0,24 \pm 0,13$	$0,27 \pm 0,13$	$0,33 \pm 0,14$	$0,37 \pm 0,15$	$0,40 \pm 0,16$	$0,45 \pm 0,18$
RF-PCT	$0,24 \pm 0,14$	**0,29 ± 0,14**	**0,34 ± 0,15**	**0,39 ± 0,14**	**0,45 ± 0,15**	**0,51 ± 0,17**

TABLE 7 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère F-mesure.

Nos conclusions sont consolidées par les résultats obtenus pour les critères de qualité supplémentaires définis dans la section 3.1.4. Les meilleurs résultats sont toujours obtenus par les méthodes ensemble RF-PCT et EBR (voir les tableaux 7-11 dans l'Annexe 2).

	2	4	8	16	32	64
MLkNN	$0,40 \pm 0,10$	$0,38 \pm 0,10$	**0,34 ± 0,10**	**0,31 ± 0,10**	$0,28 \pm 0,10$	$0,27 \pm 0,11$
CLR	**0,38 ± 0,11**	**0,37 ± 0,10**	**0,34 ± 0,10**	$0,32 \pm 0,09$	**0,28 ± 0,09**	$0,26 \pm 0,09$
EBR	$0,39 \pm 0,10$	**0,37 ± 0,09**	**0,34 ± 0,10**	$0,32 \pm 0,10$	$0,30 \pm 0,10$	$0,28 \pm 0,10$
RF-PCT	$0,39 \pm 0,10$	**0,37 ± 0,09**	$0,35 \pm 0,09$	$0,32 \pm 0,08$	**0,28 ± 0,08**	**0,25 ± 0,09**

TABLE 8 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Balanced Error Rate (BER).

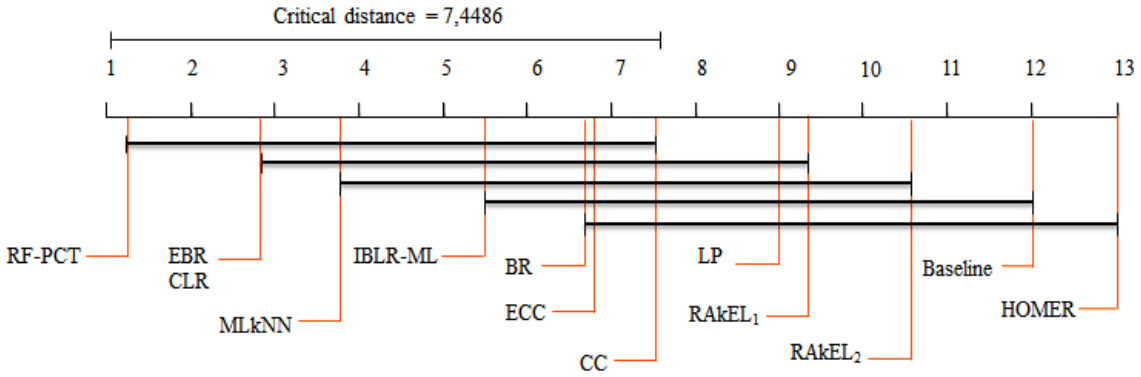


FIGURE 1 – Le diagramme critique pour le critère Ranking Loss pour toutes les tailles d'ensembles d'apprentissage.

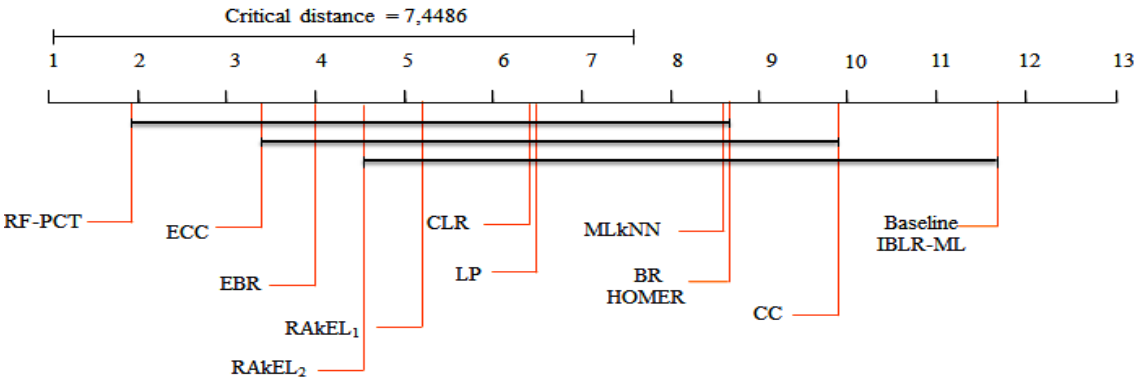
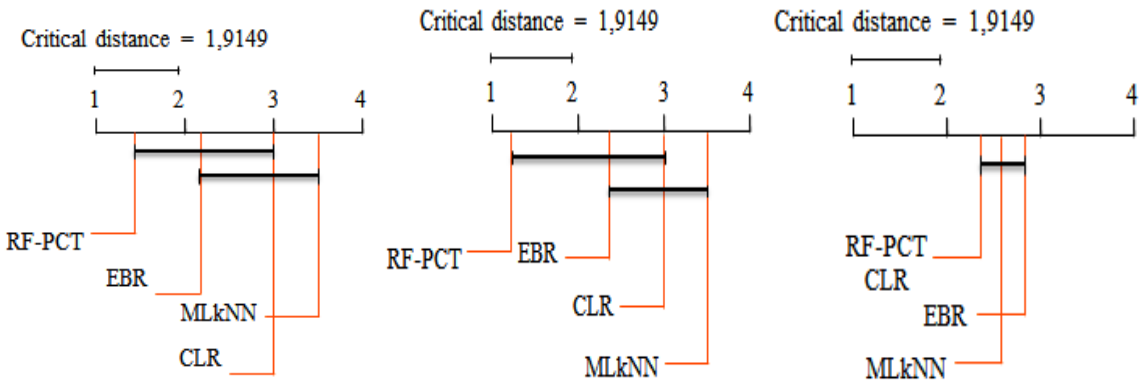


FIGURE 2 – Le diagramme critique pour le critère macro-averaged Ranking loss pour toutes les tailles d'ensembles d'apprentissage.



(a) F-mesure (b) Accuracy (c) BER

FIGURE 3 – Le diagrammes critiques pour les critères : (a) F-mesure, (b) Accuracy et (c) BER pour toutes les tailles d'ensembles d'apprentissage.

	2	4	8	16	32	64
Baseline	$0,55 \pm 0,68$	$0,56 \pm 0,69$	$0,54 \pm 0,67$	$0,53 \pm 0,67$	$0,54 \pm 0,67$	$0,53 \pm 0,66$
LP	$1,23 \pm 1,54$	$1,23 \pm 1,54$	$1,24 \pm 1,55$	$1,24 \pm 1,55$	$1,23 \pm 1,54$	$1,24 \pm 1,55$
CC	$8,90 \pm 11,3$	$8,92 \pm 11,4$	$8,91 \pm 11,4$	$8,81 \pm 11,2$	$8,80 \pm 11,2$	$8,83 \pm 11,3$
RAkEL1	$2,85 \pm 3,56$	$2,93 \pm 3,67$	$2,94 \pm 3,68$	$2,98 \pm 3,74$	$2,99 \pm 3,74$	$3,01 \pm 3,73$
RAkEL2	$5,27 \pm 6,62$	$5,96 \pm 7,55$	$6,59 \pm 8,26$	$7,26 \pm 9,10$	$7,66 \pm 9,58$	$7,91 \pm 9,90$
MLkNN	$2,66 \pm 3,34$	$3,01 \pm 3,83$	$3,66 \pm 4,82$	$4,67 \pm 6,67$	$6,67 \pm 10$	$10,54 \pm 17$
HOMER	$2,39 \pm 3$	$2,48 \pm 3,12$	$2,54 \pm 3,21$	$2,57 \pm 3,24$	$2,69 \pm 3,42$	$2,65 \pm 3,36$
IBLR-ML	$2,73 \pm 3,43$	$3,10 \pm 3,94$	$3,66 \pm 4,79$	$4,79 \pm 6,70$	$6,71 \pm 10,2$	$10,80 \pm 17,4$
CLR	$21,11 \pm 30,3$	$20,88 \pm 30,2$	$20,60 \pm 29,6$	$21,11 \pm 30,2$	$21,00 \pm 30$	$20,73 \pm 29,5$
ECC	$89,58 \pm 114$	$88,72 \pm 113$	$88,96 \pm 114$	$89,05 \pm 113$	$88,27 \pm 113$	$89,12 \pm 113$
BR	$17,29 \pm 22$	$17,21 \pm 22$	$17,32 \pm 22,1$	$17,08 \pm 21,8$	$17,17 \pm 21,9$	$17,26 \pm 22,1$
EBR	$166,92 \pm 213$	$167,77 \pm 214$	$167,31 \pm 213$	$166,86 \pm 213$	$167,02 \pm 213$	$167,66 \pm 214$
RF-PCT	**0,36 ± 0,39**	**0,34 ± 0,39**	**0,36 ± 0,42**	**0,41 ± 0,49**	**0,49 ± 0,62**	**0,66 ± 0,88**

TABLE 9 – Les temps de prédiction moyens de chaque classifieur sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage (en secondes).

6 Résultats éxperimentaux II : Apprendre et prédire en temps limité

Les tableaux 9 et 10 présentent les temps de calculs moyens (mesurés en secondes) de chaque classifieur sur tous les jeux de données en apprentissage et en prédiction en fonction pour chaque taille d'ensemble d'apprentissage (de 2 à 64 exemples). Pour le temps de prédiction des labels d'un nouvel exemple, quatre classifieurs se démarquent du reste : deux méthodes ensemble RF-PCT et RAkEL$_1$ et deux méthodes de transformation LP et HOMER. Pour le temps d'ap-

prentissage du modèle, trois méthodes de transformation LP, CC et BR obtiennent de bonnes performances. Elles sont suivies par la méthode ensemble RF-PCT.

Si les temps d'apprentissage observés sont globalement cohérents avec les complexités de calcul théoriques (Tableau 6 dans le chapitre 3), les temps de prédiction observés sont plus surprenants. En particulier, on s'attendait à ce que RF-PCT, qui requiert une centaine d'arbres de décision pour la prédiction, soit moins rapide que LP, HOMER et RAkEL$_1$ qui recourent à un plus petit nombre d'arbres de décision. Par ailleurs, quand le nombre d'exemples d'apprentissage augmente, les temps de prédiction de tous les classifieurs restent constants sauf pour les deux méthodes d'apprentissage par adaptation MLkNN et IBLR_ML qui calculent des plus proches voisins en phase de prédiction.

	2	4	8	16	32	64
Baseline	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$
LP	**0,04 ± 0,05**	**0,05 ± 0,06**	**0,07 ± 0,09**	**0,13 ± 0,17**	**0,37 ± 0,46**	**1,25 ± 1,52**
CC	$3,90 \pm 4,95$	$3,99 \pm 5,11$	$4,16 \pm 5,34$	$4,40 \pm 5,59$	$5,29 \pm 6,61$	$8,34 \pm 10,38$
RAkEL1	$37,49 \pm 73,2$	$37,72 \pm 73$	$37,76 \pm 74,5$	$38,24 \pm 72,7$	$40,25 \pm 72,1$	$48,39 \pm 81$
RAkEL2	$40,63 \pm 70$	$39,65 \pm 68,4$	$41,34 \pm 72$	$42,21 \pm 72,5$	$46,65 \pm 75,6$	$61,47 \pm 89,2$
MLkNN	$32,42 \pm 57,4$	$31,69 \pm 56,5$	$31,87 \pm 57$	$32,43 \pm 56,9$	$32,28 \pm 57,5$	$34,54 \pm 61,3$
HOMER	$35,34 \pm 61,7$	$35,50 \pm 62,4$	$35,12 \pm 62,8$	$36,07 \pm 62$	$36,08 \pm 62,1$	$39,16 \pm 65,3$
IBLR-ML	$31,33 \pm 55,8$	$32,17 \pm 56,5$	$33,03 \pm 57,9$	$33,73 \pm 62,6$	$34,11 \pm 61,9$	$35,82 \pm 64,6$
CLR	$61,24 \pm 108$	$61,18 \pm 109$	$60,88 \pm 110$	$61,88 \pm 110$	$62,74 \pm 111$	$68,04 \pm 115$
ECC	$39,84 \pm 50,8$	$39,78 \pm 50,6$	$40,28 \pm 51,4$	$42,41 \pm 53,9$	$47,46 \pm 60,0$	$63,00 \pm 78,4$
BR	$8,64 \pm 11$	$8,88 \pm 11,4$	$9,00 \pm 11,4$	$9,42 \pm 12$	$10,64 \pm 13,5$	$14,26 \pm 17,8$
EBR	$92,47 \pm 117$	$92,20 \pm 117$	$94,48 \pm 121$	$96,77 \pm 12$	$103,21 \pm 131$	$121,53 \pm 153$
RF-PCT	$3,35 \pm 4,55$	$3,61 \pm 4,88$	$4,31 \pm 5,72$	$5,84 \pm 7,47$	$9,67 \pm 11,8$	$20,25 \pm 24,3$

TABLE 10 – Les temps d'apprentissage moyens de chaque classifieur sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage (en secondes).

Ces résultats expérimentaux étonnants peuvent être expliqués en partie par l'hétérogénéité de la qualité du codage de ces algorithmes de classification issues de trois grandes libraires différentes (MULAN, Meka, CLUS). Une normalisation des implémentations sera nécessaire à l'avenir pour une analyse plus approfondie. Néanmoins, en combinant les complexités théoriques et les temps observés pendant les expérimentations, nous suggérons de retenir les méthodes de transformation LP, BR et CC et la méthode ensemble RF-PCT dont l'implémentation (fournie dans CLUS) est optimisée.

7 Discussion

L'intégration d'une approche de classification multi-label dans un système interactif est un problème de recherche prometteur qui a récemment été stimulé par des applications originales dans différents domaines réels. Dans la dernière décennie, de nombreuses approches de classification multi-label ont été développées mais une question importante est : quelle approche résiste le mieux aux contraintes d'interactivité ?

À notre connaissance, nous proposons dans ce chapitre la première étude comparative extensive des algorithmes d'apprentissage multi-label dans un environnement interactif. Nous avons comparé douze algorithmes représentatifs des trois familles d'approches (méthodes d'apprentissage par transformation, méthodes d'apprentissage par adaptation, méthodes d'apprentissage ensemble) sur douze jeux de données de tailles différentes provenant de divers domaines. La qualité de leurs prédictions a été évaluée pour cinq critères multi-label complémentaires auxquels nous avons ajouté cinq critères supplémentaires classiques de la littérature. Et la vitesse de calcul des classifieurs a été évaluée à la fois par leurs temps de calculs enregistrés et par leurs complexités de calcul théoriques pour l'apprentissage et pour la prédiction.

Notre comparaison montre que quatre classifieurs peuvent être distingués pour leur qualité de prédiction : RF-PCT (Random Forest of Predictive Clustering Trees), EBR (Ensemble of Binary Relevance), CLR (Calibrated Label Ranking) et MLkNN (Multi-label kNN) avec un avantage pour les deux premiers classifieurs ensemble. Par ailleurs, RF-PCT est également en concurrence avec les classifieurs les plus rapides qui obtiennent de mauvaises performances prédictives. En conséquence, nous concluons que RF-PCT, qui a déjà été distingué pour la classification multi-label classique (Madjarov et al., 2012), est tout aussi efficace pour une classification interactive multi-label. Néanmoins, ses temps de calcul et notamment ceux d'apprentissage -observées particulièrement pour les jeux de grande dimension qui s'approchent des tailles de données réelles rencontrées dans les applications réelles- ne sont pas négligeables : RF-PCT nécessite en moyenne entre 30 et 60 secondes pour construire un modèle prédictif à partir de 64 exemples. Pour réduire ce temps de calcul, une phase de réduction des dimensions est donc indispensable avant l'apprentissage du modèle.

5

Apprentissage multi-label à partir de données latentes

Sommaire

1 Introduction

À partir des résultats expérimentaux du chapitre 4, nous avons recommandé Random Forest of Clustering Trees (RF-PCT) (Kocev, 2011) pour être le classifieur le mieux adapté pour l'interaction. Or, les temps qu'il requiert pour fournir des prédictions et notamment pour apprendre à partir des données de grandes dimensions -même si le nombre d'exemples d'apprentissage est très faible- ne favorisent pas l'interactivité. Pour accélérer ses temps de calcul et rendre l'interaction plus fluide avec l'utilisateur, une approche de réduction des dimensions nous semble être une piste prometteuse avant de passer à l'apprentissage du modèle. Plusieurs approches de réduction de dimensions existent dans la littérature mais le problème majeur est la sélection d'une approche qui permette de compresser les données tout en conservant l'essentiel de l'information afin que RF-PCT puisse maintenir et même améliorer sa performance prédictive.

Nous proposons donc dans ce chapitre une nouvelle approche d'apprentissage multi-label hybride qui se compose de deux phases. Dans la première phase, on effectue une réduction de dimensions duale indépendante à l'aide d'un algorithme de factorisation rapide de matrice : il approxime chacune des matrices exemple-attribut et exemple-label en deux sous matrices rectangulaires. En combinant les représentations latentes des exemples dans les deux espaces latents, une représentation concise de l'ensemble d'apprentissage, où les corrélations entre les attributs et les corrélations entre les labels sont capturées, est créée. Dans la deuxième phase, on apprend un modèle de régression dans les espaces latents en utilisant le classifieur ensemble RF-PCT. Nous appelons notre approche FMDI-RF (Factorisation de Matrice duale Indépendante combinée avec le classifieur RF-PCT).

Notre analyse expérimentale se compose de deux parties : (*i*) évaluation du temps de calcul requis par l'algorithme de factorisation pour trouver la meilleure représentation latente des don-

nées et (*ii*) évaluation des performances prédictives et des temps d'apprentissage et de prédiction de RF-PCT à partir des données latentes. La qualité des prédictions est évaluée pour les cinq critères majeurs que nous avons sélectionnés dans le chapitre 4 : Ranking Loss, macro-Ranking Loss, Accuracy, F-mesure et BER auxquels nous ajoutons cinq critères supplémentaires pour consolider nos résultats : Exact match, Coverage, Average precision, Hamming Loss et One error. Pour illustrer l'efficacité de notre approche, nous avons sélectionné dix jeux de données de grandes tailles. Nous avons simplifié le protocole d'évaluation proposé dans le chapitre 4 pour bien illustrer le potentiel de notre approche en terme de précision mais surtout en terme de vitesse de calcul. Nous évaluons ses performances pour des petits ensembles d'apprentissage de taille 100. Les résultats obtenus montrent que FMDI-RF permet de réduire significativement la taille des données (réduction de 99% pour tous les jeux d'évaluation). Néanmoins, le temps requis à la recherche de la meilleure représentation latente possible est très couteux (entre 30 et 60 minutes). En terme de performance prédictive, FMDI-RF est aussi précise que RF-PCT -qui apprend à partir des données multi-dimensionnelles originales- avec parfois même un avantage à FMDI-RF pour quelques jeux de données. Cependant, les temps d'apprentissage et de prédiction de FMDI-RF sont significativement plus courts que ceux de RF-PCT. En moins d'une demi seconde, FMDI-RF peut construire un modèle d'apprentissage et fournir une prédiction pour un nouvel exemple (réduction de 99% par rapport aux temps de calcul mesurés pour RF-PCT). Ces résultats sont très prometteurs pour l'apprentissage mais le temps de calcul de la représentation latente étant très élevé. Nous avons donc tenté de le réduire.

En général, les jeux de données multi-label de grandes dimensions sont très creux : 1% des cellules de la matrice sont renseignées et le reste est inconnu ou nul. Dans le but de réduire le temps de la réduction de dimension, nous avons supposé qu'il est possible d'apprendre une bonne représentation latente à partir des cellules positives de la matrice seulement. Nous proposons ainsi une nouvelle approche FMDI-RF$^+$ relative à FMDI-RF où les facteurs attributs sont appris à partir des cellules positives de la matrice exemple-attribut. Pour évaluer la pertinence de cette hypothèse, nous avons mené une nouvelle analyse expérimentale. Les résultats obtenus valident notre hypothèse et indiquent qu'il n'y a aucune différence significative entre les performances prédictives de FMDI-RF et FMDI-RF$^+$ avec un avantage très significatif pour FMDI-RF$^+$ en temps de réduction : 2 minutes sont au maximum requises pour réduire la taille

de n'importe quel ensemble de données (soit une réduction de plus de 90%). En particulier, pour les plus grands jeux de données (sous-ensembles de Reuters), FMDI-RF$^+$ permet de réduire la dimension et d'apprendre un modèle prédictif en un temps 6 fois plus court que le temps d'apprentissage de RF-PCT ; ce qui lui permet de mieux résister aux contraintes d'interactivité -et d'être plus adaptée- que RF-PCT pour un système d'apprentissage interactif multi-label.

2 État de l'art : réduction des dimensions

En multi-label, les approches de réduction des dimensions se divisent en deux grandes familles : approches singulières et approches duales. Les approches singulières visent à réduire seulement l'un des deux espaces : espace d'attribut ou espace de labels. En revanche, les approches duales visent à réduire les deux espaces à la fois et se composent de deux groupes : les approches duales dépendantes et les approches duales indépendantes qui permettent de réduire les deux espaces de manière dépendante ou indépendante respectivement.

Tout récemment, une étude comparative des techniques de réduction de la taille des données multi-label a été proposée (Pacharawongsakda and Theeramunkong, 2013). Elle évalue l'impact de neuf approches de réduction (singulières et duales) sur les performances prédictives du classifieur multi-label "Binary Relevance" (BR) sur 5 jeux données différents pour quatre critères d'évaluation. Les résultats expérimentaux indiquent que les approches de réduction duale -impliquant un algorithme de Décomposition en Valeurs Singulières SVD- sont les plus efficaces. Une deuxième étude comparative plus fine qui concerne uniquement les approches duales montre que les approches indépendantes permettent à BR d'obtenir en moyenne les meilleures performances prédictives pour sept jeux de données et pour quatre critères d'évaluation (Pacharawongsakda and Theeramunkong, 2012). Néanmoins, le temps de la réduction des dimensions et les performances prédictives à partir des données latentes peuvent être davantage améliorés en utilisant respectivement une approche de réduction plus rapide que la SVD et un classifieur multi-label plus robuste que le classifieur de base BR.

3 Factorisation de matrice rapide

On se place ici dans le cadre général où $X = \{x_1, x_2, \ldots, x_n\}$ est un ensemble de n exemples x_i définis dans un espace multi-dimensionnel $D = \{d_1, d_2, \ldots, d_m\}$ de dimension m ($dom(x_i) \in$

$\mathcal{R}^m$) ; x_{ij} est la j^e caractéristique de x_i. Les algorithmes de factorisation de matrice rapide ont été popularisés par le concours organisé par *Netflix* entre 2006 et 2009 pour développer le meilleur algorithme de filtrage collaboratif qui permette de prédire au mieux les notes des utilisateurs pour des films (Bennett and Lanning, 2007). L'algorithme *Gravity* (Takacs et al., 2007) co-vainqueur du concours est un exemple représentatif de ces techniques. Plusieurs algorithmes de factorisation tels que Gravity traitent les données cellule par cellule plutôt que vecteur par vecteur (ligne par ligne). Soit $E = \{x_{ij} \mid x_i \in X\}$ l'ensemble des valeurs de la matrice X. Un exemple n'est donc pas un vecteur x_i comme dans le cas de l'apprentissage classique (e.g. SVM, Arbre de décision, etc) mais plutôt une cellule x_{ij} de la matrice. Comme le montre l'équation 5.1, l'idée sous-jacente aux techniques de factorisation de matrice est simple :

$$X \approx \quad P \;\; Q^T \tag{5.1}$$

Cette formule montre que la matrice $X^{n\times m}$ peut être approximée par le produit de deux sous matrices rectangulaires $P \in \mathcal{R}^{n\times k}$ et $Q \in \mathcal{R}^{m\times k}$ où les exemples x_i et les dimensions d_i originaux sont redéfinis dans des espaces latents différents A et B de taille k ($dom(x_i) \in \mathcal{R}^k$ et $dom(d_i) \in \mathcal{R}^k$). Ces espaces sont caractérisés respectivement par k facteurs : $A = \{a_1, a_2, \ldots, a_k\}$ et $B = \{b_1, b_2, \ldots, b_k\}$ avec $dom(a_i) \in \mathcal{R}$ et $dom(b_i) \in \mathcal{R}$ (voir Figure 1).

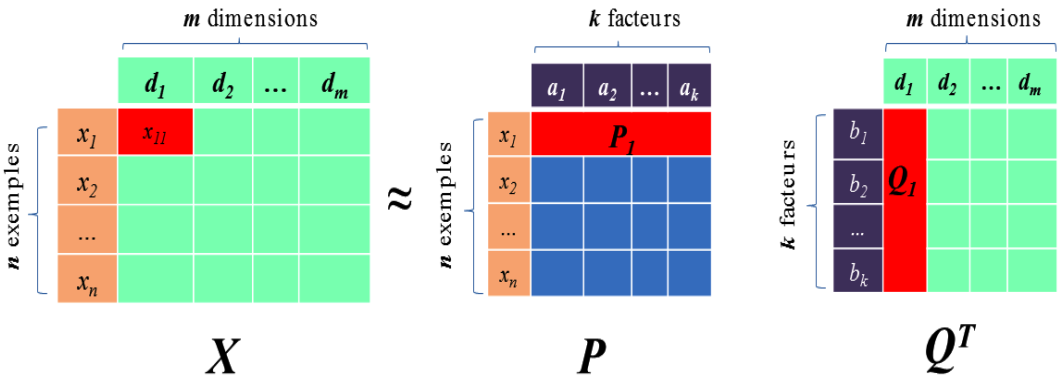


FIGURE 1 – Factorisation de la matrice $X^{n\times m}$ en sous matrices $P^{n\times k}$ et $Q^{m\times k}$.

À partir de la décomposition de la formule 5.1, la valeur de $x_{ij} \in E$ peut être approximée en multipliant la i^e ligne de la matrice P_i par la j^e colonne de la matrice Q_j^T (eq. 5.2) :

$$x_{ij} \approx \hat{x}_{ij} = P_i \times Q_j^T = \sum_{k=1}^{K} p_{ik} \times q_{kj} \tag{5.2}$$

Les matrices latentes $\mathcal{P}$ et $\mathcal{Q}$ sont apprises en minimisant un critère L_r. L'algorithme initial de factorisation rapide de matrice *Gravity* minimise l'erreur *Root Mean Squared Error* (*RMSE*). La *RMSE* calcule la racine carrée de l'erreur quadratique e_{ij}^2 moyenne commise par le modèle pour chaque cellule $x_{ij} \in E$. L'intérêt de la *RMSE* est -qu'élevée au carré- elle conduit à une fonction d'erreur quadratique, analytiquement pratique à dériver et pour laquelle une optimisation de type descente de gradient est efficace. La *RMSE* du modèle sur l'ensemble E est définie par :

$$RMSE_E = \sqrt{\frac{\sum_{\forall i,j/x_{ij} \in E} (x_{ij} - \hat{x}_{ij})^2}{|E|}} \tag{5.3}$$

De manière itérative, l'apprentissage des matrices P et Q est ainsi effectué en minimisant la moyenne des erreurs quadratiques entre les valeurs réelles x_{ij} et les valeurs prédites par le modèle $\hat{x}_{ij}$ (eq. 5.4) ; ce qui est équivalent à minimiser la RMSE.

$$\begin{aligned}
(P^*, Q^*) &= \underset{P,Q}{\operatorname{argmin}} L_r(X, P, Q) \\
&= \underset{P,Q}{\operatorname{argmin}} ||X - P\, Q^T||^2 \\
&= \underset{P,Q}{\operatorname{argmin}} \sum_{\forall i,j/x_{ij} \in E} e_{ij}^2 = \sum_{(i,j) \in R} (x_{ij} - \hat{x}_{ij})^2 \\
&= \underset{P,Q}{\operatorname{argmin}} \sum_{\forall i,j/x_{ij} \in E} \left(x_{ij} - \sum_{k=1}^{K} p_{ik} \times q_{kj} \right)^2
\end{aligned} \tag{5.4}$$

Pour chaque cellule d'apprentissage $x_{ij} \in E$, l'algorithme prédit une valeur $\hat{x}_{ij}$ en multipliant la i^{e} ligne de la matrice P_i par la j^{e} colonne de la matrice Q_j^T. Il calcule ensuite l'erreur constatée e_{ij} et rétro-propage le gradient de l'erreur dans les facteurs des matrices P et Q (eq. 5.5 et 5.6). Le gradient de l'erreur e_{ij}^2 relativement à chaque facteur ligne et chaque facteur

colonne est calculé comme suit dans les équations :

$$\frac{\sigma}{\sigma p_{ik}} e_{ij}^2 = -2 \times e_{ij} \times q_{kj} \tag{5.5}$$

$$\frac{\sigma}{\sigma q_{kj}} e_{ij}^2 = -2 \times e_{ij} \times p_{ik} \tag{5.6}$$

Pour faire décroître ces erreurs et mieux approximer les x_{ij}, les facteurs sont mis à jour dans la direction opposée du gradient où η est le pas d'apprentissage permettant de régler la vitesse d'apprentissage des facteurs (eq. 5.7 et 5.8). Le paramètre η prend généralement une petite valeur inférieure à 0.1.

$$p'_{ik} = p_{ik} + \eta \times 2 \times e_{ij} \times q_{kj} \tag{5.7}$$

$$q'_{kj} = q_{kj} + \eta \times 2 \times e_{ij} \times p_{ik} \tag{5.8}$$

Pour ne pas surapprendre les matrices P et Q et pour éviter que les facteurs divergent vers des valeurs élevées, une régularisation λ est nécessaire (eq. 5.9 et 5.10) :

$$p'_{ik} = p_{ik} + \eta \times 2 \times e_{ij} \times q_{kj} - \lambda \times p_{ik} \tag{5.9}$$

$$q'_{kj} = q_{kj} + \eta \times 2 \times e_{ij} \times p_{ik} - \lambda \times q_{kj} \tag{5.10}$$

À la fin de chaque itération d'apprentissage, la RMSE du modèle (P, Q) est calculée sur un petit ensemble de validation. Si au fil des itérations la RMSE ne réduit pas significativement, l'apprentissage s'arrête et les matrices P et Q courantes sont considérées optimales i.e. apprentissage par *early stopping* (Takacs et al., 2007). Une fois la phase d'apprentissage terminée, la valeur de n'importe quelle cellule x_{ij} peut être facilement obtenue à l'aide de l'équation 5.2. Si la matrice originale X contient des valeurs manquantes, celles-ci peuvent être prédites par le modèle (P et Q). L'algorithme de factorisation classique *Gravity* est résumé dans Algorithme

1.

Algorithme 1 : Algorithme de factorisation rapide de matrice.

Données : E : ensemble de cellules;

Entrées : k : nombre de facteurs, η : pas d'apprentissage, λ : régularisation, t : seuil de précision, $nbitmax$: nombre d'itérations maximal;

Initialisation aléatoire des matrices latentes P et Q dans [0 ; 0.1];

$rmseValid$ = 0 /* rmse de validation courante */;

$nbit$ = 0 /* nombre d'itérations courant */;

$nbitmax$ = 0 ;

$rmseValidMin$ = 1e6 /* rmse de validation précédente */;

tant que *($nbit$ < $nbitmax$)* **faire**

  pour *chaque cellule d'apprentissage* x_{ij} **faire**

    calculer l'erreur quadratique e_{ij} ;

    calculer le gradient de e_{ij}^2 ;

    mettre à jour la i^e ligne de P and et la j^e colonne de Q;

  Calculer la $rmseValid$ sur les cellules de validation;

  si *($rmseValidMin$ - $rmseValid$ < t)* **alors**

    $nbit$++;

  sinon

    $nbit$ = 0;

    $rmseValidMin = rmseValid$;

Résultat : P*, Q*

4 Apprentissage multi-label à partir de données latentes

4.1 Phase 1 : Réduction des dimensions par factorisation matricielle

Considérons la matrice des données $\mathcal{D}_{t_0}$ où seuls quelques exemples $x_i \in \mathcal{T}_{t_0}$ sont étiquetés (Figure 2). Notre algorithme de réduction des dimensions de $\mathcal{D}_{t_0}$ est décrit comme suit. Premièrement, $\mathcal{D}_{t_0}$ est divisée en deux matrices : une matrice exemple-attribut A et une matrice exemple-label E. La matrice A (en vert) contient les m attributs de tous les exemples (étiquetés ou pas) et la matrice E (en bleu) contient les q labels des exemples d'apprentissage uniquement. Nous supposons qu'en pratique les données non étiquetées sont toujours disponibles au moment de l'apprentissage. Pour réduire les deux espaces d'attributs et de labels, deux factorisations de matrices sont indépendamment appliquées aux matrices A et E. Chacune des deux matrices est approximée par le produit de deux sous-matrices rectangulaires : A par B et C, et E par F et G (eq. 5.11 et 5.12) :

$$A \approx B\ C^T \tag{5.11}$$

$$E \approx F\ G^T \tag{5.12}$$

Les matrices B et C décrivent respectivement les exemples (d'apprentissage et de test) et les attributs par k facteurs tandis que les matrices F et G décrivent respectivement les exemples d'apprentissage et les labels par k' facteurs (Figures 3 et 4). Comme la matrice A est factorisée en une seule fois, l'apprentissage des facteurs attributs des nouveaux exemples avant la prédiction est évité. Cela permet aussi de tirer avantage de l'information contenue dans les exemples non-étiquetés pour apprendre des facteurs plus précis. En combinant les deux matrices exemple-facteur B et F, un nouvel ensemble d'apprentissage est crée où les exemples sont décrits par k facteurs attributs (en rose) et étiquetés par k' labels latents (en vert) (Figure 5).

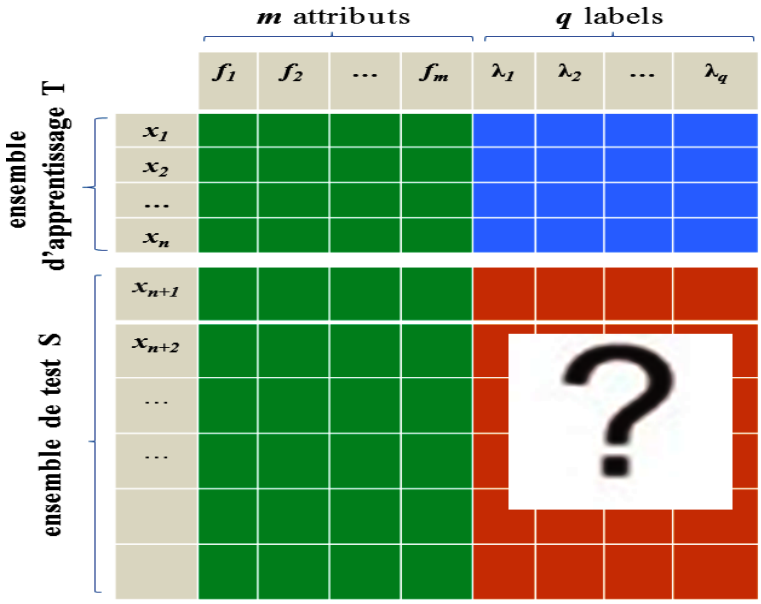


FIGURE 2 – Problème de classification multi-label.

4.2 Phase 2 : Apprentissage et prédiction multi-label dans les espaces latents

Pour apprendre un modèle prédictif à partir de cette nouvelle vue de l'ensemble d'apprentissage, nous utilisons le classifieur Random Forest of Predictive Clustering Tree (RF-PCT) (Ko-

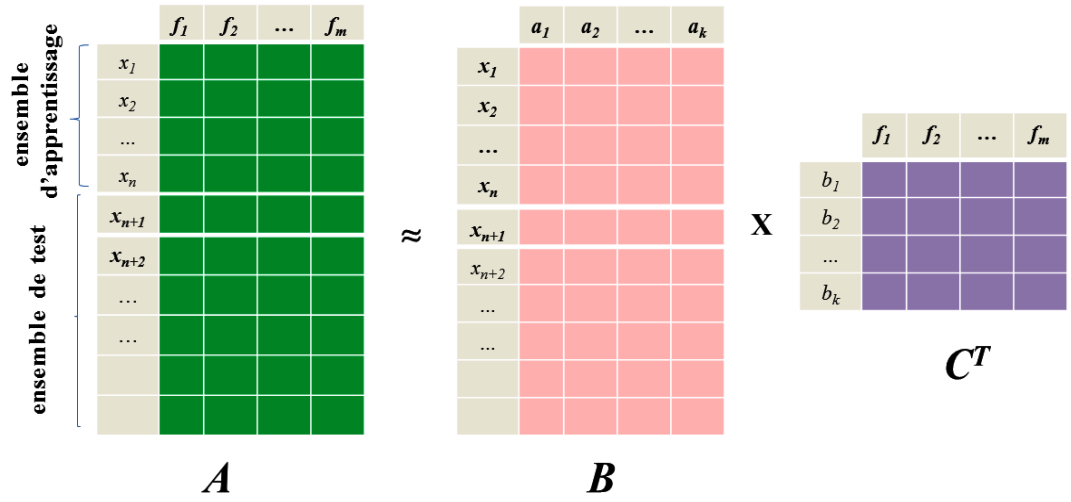


FIGURE 3 – Factorisation de la matrice exemple-attribut.

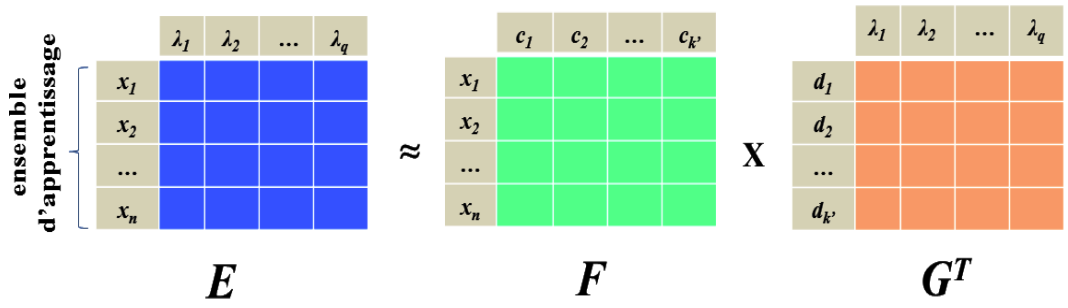


FIGURE 4 – Factorisation de la matrice exemple-label.

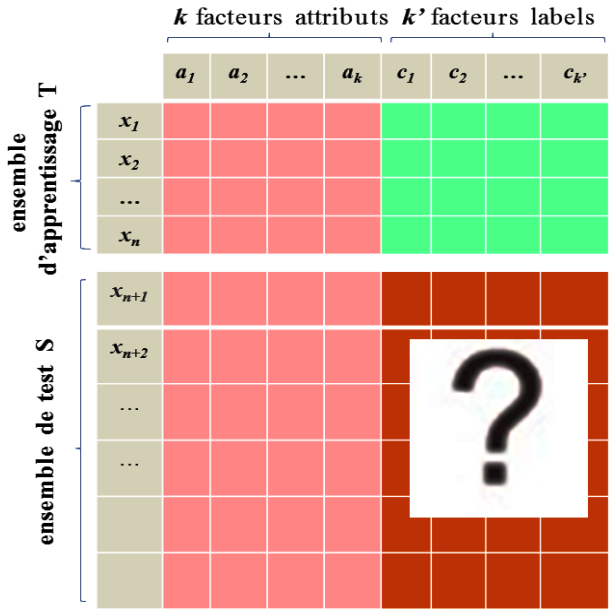


FIGURE 5 – Représentation latente de la matrice $\mathcal{D}_{t_0}$.

cev, 2011) que nous avons sélectionné dans le chapitre 4. Nous rappelons que RF-PCT est un classifieur ensemble qui combine les performances de 100 arbres de décision multi-label de type PCT (Predictive Clustering Trees). Chaque arbre de décision est appris sur un sous-échantillon

de l'ensemble d'apprentissage et pour la découpe de chaque nœud un nouveau sous-ensemble d'attributs est sélectionné aléatoirement. Pour notre tâche de régression, le critère de sélection des attributs est la réduction de la somme des variances des labels. Les feuilles sont associées avec des vecteurs réels contenant le vote moyen pour chaque label.

Pour la prédiction des labels d'un nouvel exemple, chaque arbre fournit une prédiction dans l'espace de labels latent en fonction de ses facteurs attributs appris dans la phase de factorisation de la matrice exemple-attribut. Les prédictions de tous les arbres sont ensuite moyennées pour chaque label latent. Pour ramener ce vecteur de prédictions dans l'espace de labels original, il est multiplié par la matrice label-facteur G^T.

5 Étude expérimentale

Dans cette section, nous présentons brièvement les données sélectionnées pour illustrer l'efficacité de notre approche proposée. Ensuite, nous définissons précisément notre protocole expérimental et rappelons les critères d'évaluation utilisés.

5.1 Implémentation

Pour la réduction de dimension, nous avons développé notre propre code de l'algorithme Gravity dans le langage Java et l'avons intégré dans la libraire multi-label CLUS. La représentation latente des données retournée par cet algorithme est ensuite fournie en entrée au classifieur RF-PCT qui est implémenté dans cette même libraire.

5.2 Données d'évaluation

Pour illustrer l'efficacité de notre approche proposée FMDI-RF et de son amélioration FMDI-RF$^+$, nous sélectionnons dix jeux de données multi-label de grandes dimensions qui se rapprochent des données d'un cas d'usage réel : ces données sont principalement issues du domaine de texte. Leurs caractéristiques de base et les mesures de distribution de leurs labels montrent qu'elles couvrent un grand nombre de situations diverses. En effet, leurs nombres d'attributs varient de 21925 à 49060 et leurs nombres de labels varient de 22 à 101. Nous avons supposé dans l'étude comparative du chapitre 4 que les utilisateurs ne définissent généralement qu'un nombre

limité de labels pour classer leurs données. Cependant, nous avons quand même souhaité évaluer notre approche sur des données avec un nombre de labels plus grand. Or, des données où le nombre d'attributs et le nombre de labels sont importants ne sont pas encore disponibles. Des descriptions succinctes de ces jeux de données sont fournies dans la section 5 du chapitre 3.

Dans un cadre de classification interactive, le modèle d'apprentissage est souvent appliqué sur une partie des données non-étiquetées seulement car il n'est pas possible de prédire les labels de tous les exemples en un temps raisonnable. Donc, pour obtenir des estimations des temps d'apprentissage et de prédiction moyens des classifieurs, nous conservons ici uniquement des échantillons de 1000 exemples sélectionnés aléatoirement à partir des données d'évaluation originales.

5.3 Protocole expérimental

Dans cette section, nous décrivons notre protocole d'évaluation de notre approche d'apprentissage hybride FMDI-RF, son amélioration FMDI-RF$^+$ et l'approche d'apprentissage de référence RF-PCT. La première étape consiste à diviser la matrice de données en deux sous-matrices : exemple-attribut et exemple-label. Ces deux sous-matrices sont ensuite factorisées de manière indépendante telle que chaque sous-matrice est approximée par le produit de deux nouvelles sous-matrices latentes. L'apprentissage des facteurs des matrices latentes s'effectue en plusieurs itérations en minimisant la RMSE en validation. En effet, si après 5 itérations d'apprentissage, cette erreur ne diminue pas de 0.01, nous arrêtons le processus d'apprentissage des facteurs et considérons que les matrices obtenus sont optimales (*early stopping*). En combinant les deux représentations latentes des exemples obtenues, une nouvelle vue de l'ensemble d'apprentissage est créée. Cependant, comme les deux espaces sont obtenus séparément, une question majeure se pose : quelle est la qualité de cette nouvelle représentation des données ? Plus précisément, les facteurs attributs ont-ils le pouvoir de prédire les facteurs labels ?

La qualité de la représentation latente apprise dépend principalement du nombre de facteurs mais aussi du pas d'apprentissage de l'algorithme et de la régularisation. Comme nous réduisons les deux espaces de manière indépendante, il y a donc six paramètres à estimer en tout. Mais nous les réduisons à 4 en sélectionnant le même pas d'apprentissage et la même régularisation pour les deux factorisations (attributs et labels). De notre expérience, nous considérons

qu'ils ont moins d'impact sur la qualité de la représentation que les nombres de facteurs. Ce problème d'estimation de paramètres peut être efficacement abordé par des heuristiques ou métaheuristiques d'optimisation combinatoire telles qu'une descente de gradient (Boyd and Vandenberghe, 2004) ou une recherche à voisinage variable (Mladenović and Hansen, 1997) mais nous le traitons avec une recherche exhaustive simple à partir d'une exploration d'un sous-ensemble de valeurs plausibles. Pour chaque paramètre, nous sélectionnons un petit ensemble de valeurs ou un intervalle. Pour estimer le nombre de facteurs attributs k, nous sélectionnons 4 valeurs $k \in \{16, 32, 64, 128\}$ et nous soulignons qu'au delà de 128 facteurs, les performances prédictives de RF-PCT restent stables. Pour estimer le nombre de facteurs labels k', nous sélectionnons aussi 4 valeurs $k' \in \{2, 4, 8, 16\}$. Les deux nombres de facteurs ne sont pas définis dans la même échelle de valeurs car le nombre d'attributs de nos jeux d'évaluations est largement plus grand que celui de labels. Ainsi, nous testons 16 combinaisons de valeurs de facteurs différentes. Le pas d'apprentissage et la régularisation sont considérés moins prioritaires que les deux premiers paramètres. Nous avons testé 4 valeurs -sélectionnées de manière aléatoire- pour chacun de ces paramètres dans les intervalles : [0.01 ; 0.1] pour le pas d'apprentissage et dans l'intervalle [0.001 ; 0.01] pour la régularisation. Pour chaque jeu de paramètres, une nouvelle représentation de données est créée et est fournie à RF-PCT pour l'apprentissage du modèle prédictif. De ces nouvelles données, 10 paires d'ensembles (apprentissage, test) de tailles 10% et 90% sont respectivement créées. Nous entrainons nos approches sur 5 ensembles d'apprentissage et nous évaluons la qualité des modèles appris sur des petits ensembles de validation -crées à partir de chaque ensemble d'apprentissage- pour le critère Accuracy. Le jeu de paramètres qui permet de maximiser l'Accuracy est conservé et ensuite utilisé pour mener une 10-validation croisée. Ce protocole expérimental est décrit étape par étape dans ce qui suit :

Protocole expérimental :

1. Chaque ensemble $\mathcal{D}$ de données est divisé en deux sous-matrices exemple-attribut A et exemple-label E.

2. Chaque sous-matrice est transformée en un ensemble de cellules (ligne, colonne, valeur).

3. De chaque ensemble de cellules, conserver 10% de cellules pour la validation des facteurs appris par le modèle.

4. Pour tout $k \in \{16, 32, 64, 128\}$ et pour tout $k' \in \{2, 4, 8, 16\}$ faire

   (a) Sélectionner aléatoirement une valeur dans l'intervalle $[0.01; 0.1]$ pour la régularisation η.

   (b) Sélectionner aléatoirement une valeur dans l'intervalle $[0.001; 0.01]$ pour le pas d'apprentissage λ.

   (c) À partir des deux ensembles de cellules, deux factorisations de matrices indépendantes sont effectuées : A est approximée par les deux sous-matrices latentes B et C, et E par F et G. À la fin de chaque itération d'apprentissage, les cellules conservées pour la validation sont utilisées pour évaluer la qualité des facteurs obtenus pour le critère RMSE. Si au bout de 5 itérations, cette erreur ne diminue pas de 0.01, l'apprentissage des facteurs est arrêté (*« early stopping »*) et les matrices obtenus sont considérés comme optimales.

   (d) Une fois que les matrices latentes sont obtenues, B et F sont combinées pour constituer le nouvel ensemble de données $\mathcal{D}'$.

   (e) $\mathcal{D}'$ est divisé en 10 ensembles disjoints tel que chaque ensemble (10% de $\mathcal{D}'$) est utilisé pour l'apprentissage et les autres ensembles (90% de $\mathcal{D}'$) pour le test. De chaque ensemble d'apprentissage, 10% des exemples sont conservés pour la validation du modèle d'apprentissage construit par RF-PCT.

   (f) Pour 5 ensembles d'apprentissage, RF-PCT apprend un modèle prédictif et l'évalue sur les 10% d'exemples de validation pour le critère Accuracy. L'Accuracy moyenne pour les 5 ensembles de validation est ensuite calculée.

   Les paramètres ayant permis de maximiser l'Accuracy sont utilisés pour évaluer les performances moyennes des approches avec une 10-validation croisée pour différents critères.

5.4 Critères d'évaluation

L'évaluation des performances de notre approche d'apprentissage et de son amélioration s'organise en deux parties. La première partie consiste à évaluer leurs taux de réduction des dimensions pour chaque jeu de données et le temps de calcul (en minutes) qu'elles requièrent pour trouver sa meilleure représentation dans les espaces latents. La deuxième partie consiste à évaluer leurs performances prédictives et leurs temps d'apprentissage et de prédiction pour chaque jeu de données. Nous reprenons les mêmes critères de qualité sélectionnés dans l'étude comparative du chapitre 4 en fonction des propriétés essentielles dans une application de classification interactive multi-label (section 3.1). Pour évaluer la capacité des classifieurs à bien ordonner les labels d'un exemple, nous sélectionnons le critère : *Ranking Loss* (RL). Pour évaluer leurs capacité à bien ordonner les exemples d'un ou plusieurs labels, nous avons adapté la définition du critère RL et nous proposons le critère macro-averaged Ranking-Loss (*macro-RL*). Et, pour évaluer la capacité des classifieurs à bien classer les labels d'un exemple, nous sélectionnons trois critères : *Accuracy*, *F-mesure* et l'adaptation multi-label du Balanced Error Rate (*BER*). Pour valider davantage nos conclusions, nous reprenons également les cinq critères supplémentaires utilisés dans le chapitre 4 : Exact match, Coverage, Average precision, Hamming Loss et One error. La vitesse de calcul des classifieurs est mesurée par les temps observés (en secondes) dans les expérimentations pour l'apprentissage des modèles et pour la prédiction des labels d'un nouvel exemple.

6 Résultats expérimentaux

Pour illustrer l'efficacité de notre approche proposée et de son amélioration, nous présentons dans le tableau de synthèse 4 les facteurs d'amélioration des performances prédictives pour les critères de qualité majeurs et les facteurs d'accélération des temps d'apprentissage et de prédiction par rapport à RF-PCT. Les facteurs supérieurs ou égaux à 1 sont présentés en gras.

6.1 Résultats expérimentaux I : Temps de la réduction des dimensions

Dans cette section, nous présentons les performances de notre approche de réduction des dimensions FMDI-RF et de son amélioration FMDI-RF$^+$ en termes de taux de réduction et de

temps de calcul. Dans le tableau 1, nous rappelons les caractéristiques originales de chaque jeu de données (nombre d'attributs (#nb attributs) et nombre de labels (#nb labels)) et nous présentons ses nouvelles caractéristiques après réduction des dimensions (nombre de facteurs attributs (#nb facteurs attributs) et nombre de facteurs labels (#nb facteurs labels)). Nous complétons le tableau avec le taux de réduction (Réduction %) et le temps de calcul en minutes (Temps) requis pour fournir la meilleure représentation latente de chaque jeu.

Jeu	**Approche**	**#nb labels**	**#nb attributs**	**#nb facteurs labels**	**#nb facteurs attributs**	**Réduction %**	**Temps (m)**
Arts	FMDI-RF	24	23146	8	16	> 99%	27,23
	FMDI-RF$^+$			4	128		**0,98**
Business	FMDI-RF	28	21924	16	16	> 99%	29,11
	FMDI-RF$^+$			16	16		**0,90**
Health	FMDI-RF	25	30605	16	32	> 99%	43,00
	FMDI-RF$^+$			8	64		**1,11**
Computers	FMDI-RF	30	34096	16	128	> 99%	42,67
	FMDI-RF$^+$			16	64		**1,37**
TMC	FMDI-RF	22	49060	2	64	> 99%	58,78
	FMDI-RF$^+$			4	16		**2,24**
Reuters S1	FMDI-RF	101	47236	16	64	> 99%	56,21
	FMDI-RF$^+$			16	16		**1,33**
Reuters S2	FMDI-RF	101	47236	8	16	> 99%	60,63
	FMDI-RF$^+$			8	16		**1,44**
Reuters S3	FMDI-RF	101	47236	8	16	> 99%	57,53
	FMDI-RF$^+$			8	32		**1,49**
Reuters S4	FMDI-RF	101	47229	8	64	> 99%	55,27
	FMDI-RF$^+$			8	16		**1,51**
Reuters S5	FMDI-RF	101	47235	16	64	> 99%	54,23
	FMDI-RF$^+$			2	32		**1,50**

TABLE 1 – Les performances de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF$^+$ en termes de taux de réduction et de temps de calcul en minutes.

Les deux approches permettent de réduire la taille de tous les jeux de données en très peu de dimensions et obtiennent ainsi un taux de réduction de 99% partout. Parmi les 16 combinaisons de facteurs testées, les résultats indiquent que la combinaison de 16 facteurs attributs et 8 facteurs labels est la plus appropriée : elle permet de réduire significativement les dimensions tout en conservant le maximum d'information. Les combinaisons de facteurs sélectionnées par

chaque approche pour chaque jeu de données sont différentes sauf pour les jeux *Business et Reuters S2* où elles restent inchangées. Cependant, la différence majeure entre nos deux approches réside dans le temps de calcul des facteurs due à la grande variance dans le nombre de cellules d'apprentissage : FMDI-RF utilise toutes les cellules de la matrice exemple-attribut alors que FMDI-RF$^+$ en exploite uniquement 1% (cellules positives). En effet, FMDI-RF nécessite entre moins de 30 minutes (pour Arts et Business) et au plus une heure pour tous les autres jeux de plus grandes tailles alors que FMDI-RF$^+$ requiert au maximum 2 minutes et quelques secondes pour compresser des données de n'importe quelle taille, soit une réduction de temps de plus de 96%.

6.2 Résultats expérimentaux II : Performance prédictive

Le tableau 2 présente les performances prédictives moyennes de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF$^+$ pour chaque jeux de données et pour chaque critère d'évaluation sélectionné : Ranking Loss (RL), macro-Ranking Loss (macro-RL), Accuracy, F-mesure et Balanced Error Rate (BER). Puisque les écarts-types de toutes ces moyennes de performances sont quasiment nuls, nous ne les présentons donc pas dans ce tableau. Les résultats montrent premièrement que FMDI-RF et RF-PCT obtiennent des performances similaires avec un avantage à RF-PCT pour la plupart des jeux de données sauf pour Computers où FMDI-RF est meilleure presque pour tous les critères : macro-RL, Accuracy, F-mesure et BER (en gras dans le tableau). Cependant, ces différences de performances ne sont pas significatives selon le test statistique de Wilcoxon avec un niveau de significativité de 95%.

Deuxièmement, les résultats montrent que FMDI-RF$^+$ et RF-PCT obtiennent également des performances du même ordre de grandeur. En moyenne, RF-PCT surclasse légèrement FMDI-RF$^+$ pour tous les critères d'évaluation. Néanmoins, FMDI-RF$^+$ surclasse RF-PCT sur trois jeux de données Business, Computers et Health pour les critères : Accuracy, F-mesure pour Business, macro-RL, Accuracy, F-mesure et BER pour Computers et macro-RL pour Health (en gras dans le tableau). Le test de Wilcoxon n'indique aucune différence de performance statistiquement significative entre les deux approches sauf pour le critère macro-RL. Ainsi, nous concluons à partir de nos expérimentations empiriques que l'apprentissage à partir de données latentes apprises avec toutes les cellules ou avec seulement les cellules positives est similaire

à l'apprentissage à partir des données originales multi-dimensionnelles. En effet, FMDI-RF et FMDI-RF$^+$ obtiennent en moyenne exactement les mêmes scores pour les critères RL et BER avec un léger avantage à FMDI-RF$^+$ pour les autres critères i.e. macro-RL, Accuracy, F-mesure. Nos conclusions sont confirmées par les résultats obtenus pour les cinq critères de qualité supplémentaires (Exact match, Coverage, Average precision, Hamming Loss et One error) dans l'Annexe 4. Par ailleurs, le test Wilcoxon indique que ce léger écart de performance n'est pas statistiquement significatif. Mais en combinant ces résultats avec les temps de calcul requis à la réduction, FMDI-RF$^+$ fournit le meilleur compris entre la performance prédictive et la vitesse de calcul.

6.3 Résultats expérimentaux III : Vitesse d'apprentissage et de prédiction

Le tableau 3 présente les temps moyens requis (en secondes) pour RF-PCT et nos deux approches FMDI-RF et FMDI-RF$^+$ pour apprendre un modèle à partir de 90 exemples de chaque jeux de données et de prédire les labels d'un nouvel exemple. Nous ne présentons pas les écarts-types de ces moyennes de temps car ils sont presque nuls. Les résultats montrent que la réduction des dimensions permet de réduire de manière très significative le temps d'apprentissage de RF-PCT ; la réduction est de plus de 99% pour tous les jeux de données et notamment pour les jeu les plus volumineux Reuters où nos approches montrent le meilleur de leurs performances. Pour ces jeux de données, le temps d'apprentissage de RF-PCT est passé de plus de 9 minutes à moins d'une demi seconde. Notons que FMDI-RF et FMDI-RF$^+$ ne partagent pas exactement les mêmes temps d'apprentissage car elles apprennent à partir de deux vues de tailles différentes de chaque jeu de données. Néanmoins, ils sont du même ordre de grandeur.

Même si les trois approches entraînent le même nombre d'arbres de décision, les résultats expérimentaux montrent que la réduction des dimensions permet également de réduire leur temps de prédiction de plus de 92% pour chaque jeu de donnée et en particulier pour TMC où le temps de prédiction de RF-PCT est réduit de 3.76 secondes à moins de 0.05 secondes, soit une réduction de plus de 99%. En effet, les arbres appris par FMDI-RF et FMDI-RF$^+$ à partir des données latentes sont de plus petites tailles que les arbres appris par RF-PCT à partir des données originales : un seul facteur attribut peut remplacer un ensemble d'attributs corrélés. À la différence des temps d'apprentissage, les temps de prédictions enregistrés pour les deux

approches sont très proches puisque elles construisent des arbres de faibles complexités.

		RL	macro-RL	Accuracy	F-mesure	BER
Arts	FMDI-RF$^+$	0,24	0,5	0,17	0,22	0,39
	FMDI-RF	0,24	0,49	0,17	0,22	0,4
	RF-PCT	0,19	0,4	0,25	0,3	0,35
Business	FMDI-RF$^+$	0,06	0,4	**0,65**	**0,73**	0,12
	FMDI-RF	0,06	0,41	0,63	0,72	0,12
	RF-PCT	0,06	0,36	0,64	0,72	0,12
Health	FMDI-RF$^+$	0,13	**0,37**	0,4	0,46	0,28
	FMDI-RF	0,14	0,39	0,36	0,43	0,29
	RF-PCT	0,10	0,38	0,44	0,51	0,24
Computers	FMDI-RF$^+$	0,13	**0,37**	**0,49**	**0,54**	**0,23**
	FMDI-RF	0,12	**0,37**	**0,48**	**0,54**	**0,24**
	RF-PCT	0,12	0,41	0,42	0,48	0,25
TMC	FMDI-RF$^+$	0,19	0,5	0,29	0,39	0,32
	FMDI-RF	0,19	0,5	0,28	0,39	0,32
	RF-PCT	0,13	0,25	0,4	0,51	0,25
Reuters S1	FMDI-RF$^+$	0,21	0,46	0,12	0,18	0,41
	FMDI-RF	0,22	0,46	0,11	0,18	0,42
	RF-PCT	0,13	0,28	0,23	0,32	0,32
Reuters S2	FMDI-RF$^+$	0,25	0,45	0,13	0,20	0,39
	FMDI-RF	0,23	0,47	0,13	0,20	0,39
	RF-PCT	0,15	0,32	0,21	0,29	0,34
Reuters S3	FMDI-RF$^+$	0,25	0,46	0,12	0,19	0,40
	FMDI-RF	0,24	0,43	0,13	0,20	0,40
	RF-PCT	0,15	0,33	0,20	0,28	0,34
Reuters S4	FMDI-RF$^+$	0,21	0,44	0,14	0,20	0,38
	FMDI-RF	0,21	0,45	0,14	0,21	0,38
	RF-PCT	0,13	0,31	0,21	0,28	0,33
Reuters S5	FMDI-RF$^+$	0,23	0,47	0,15	0,22	0,38
	FMDI-RF	0,23	0,44	0,16	0,24	0,37
	RF-PCT	0,15	0,30	0,20	0,29	0,34

TABLE 2 – Les performances prédictives moyennes de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF$^+$ pour les critères de qualité majeurs.

En pratique, le système d'apprentissage doit être prêt à fournir des prédictions personnalisées à tout moment : lorsque l'utilisateur ajoute de nouveaux exemples ou corrige des mauvaises prédictions, le modèle doit se mettre à jour au plus vite pour prendre en compte ces nouvelles informations et fournir de meilleurs résultats. En comprimant les données, les temps d'apprentissage et de prédiction de RF-PCT ont été significativement réduits. Ainsi, il devient possible

d'effectuer des mises à jour au modèle avec un faible coût et de fournir essentiellement des prédictions dans les conditions préconisées en IHM.

Jeu		**Temps d'apprentissage**	**Temps de prédiction**
Arts	FMDI-RF$^+$	**0,26**	**0,05**
	FMDI-RF	**0,39**	**0,05**
	RF-PCT	70,76	1,17
Business	FMDI-RF$^+$	**0,26**	**0,09**
	FMDI-RF	**0,27**	**0,1**
	RF-PCT	61,50	1,23
Health	FMDI-RF$^+$	**0,35**	**0,1**
	FMDI-RF	**0,26**	**0,09**
	RF-PCT	89,26	1,38
Computers	FMDI-RF$^+$	**0,37**	**0,13**
	FMDI-RF	**0,27**	**0,1**
	RF-PCT	124,09	1,65
TMC	FMDI-RF$^+$	**0,26**	**0,03**
	FMDI-RF	**0,27**	**0,04**
	RF-PCT	91,15	3,76
Reuters S1	FMDI-RF$^+$	**0,25**	**0,10**
	FMDI-RF	**0,24**	**0,04**
	RF-PCT	516,57	2,49
Reuters S2	FMDI-RF$^+$	**0,25**	**0,04**
	FMDI-RF	**0,38**	**0,09**
	RF-PCT	572,88	2,30
Reuters S3	FMDI-RF$^+$	**0,26**	**0,04**
	FMDI-RF	**0,27**	**0,04**
	RF-PCT	514,61	2,25
Reuters S4	FMDI-RF$^+$	**0,27**	**0,07**
	FMDI-RF	**0,25**	**0,06**
	RF-PCT	581,02	2,52
Reuters S5	FMDI-RF$^+$	**0,30**	**0,05**
	FMDI-RF	**0,31**	**0,10**
	RF-PCT	568,25	2,34

TABLE 3 – Les temps d'apprentissage et de prédiction moyens de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF$^+$ en secondes.

		RL	macro-RL	Accuracy	F-mesure	BER	T. Apprentissage	T. Prédiction
Arts	FMDI-RF$^+$	x 0,79	x 0,80	x 0,68	x 0,73	x 0,90	**x 272,15**	**x 23,40**
	FMDI-RF	x 0,79	x 0,82	x 0,68	x 0,73	x 0,88	**x 181,44**	**x 23,40**
Business	FMDI-RF$^+$	**x 1**	x 0,90	**x 1,02**	**x 1,01**	**x 1**	**x 236,54**	**x 13,67**
	FMDI-RF	**x 1**	x 0,88	x 0,98	**x 1**	**x 1**	**x 227,78**	**x 12,30**
Health	FMDI-RF$^+$	x 0,77	**x 1,03**	x 0,91	x 0,90	x 0,86	**x 255,03**	**x 13,80**
	FMDI-RF	x 0,71	x 0,97	x 0,82	x 0,84	x 0,83	**x 343,31**	**x 15,33**
Computers	FMDI-RF$^+$	x 0,92	**x 1,11**	**x 1,17**	**x 1,13**	**x 1,09**	**x 335,38**	**x 12,69**
	FMDI-RF	**x 1**	**x 1,11**	**x 1,14**	**x 1,13**	**x 1,04**	**x 459,59**	**x 16,50**
TMC	FMDI-RF$^+$	x 0,68	x 0,50	x 0,73	x 0,76	x 0,78	**x 350,58**	**x 125,33**
	FMDI-RF	x 0,68	x 0,50	x 0,70	x 0,76	x 0,78	**x 337,59**	**x 94**
Reuters S1	FMDI-RF$^+$	x 0,62	x 0,61	x 0,52	x 0,56	x 0,78	**x 2066,28**	**x 24,90**
	FMDI-RF	x 0,60	x 0,61	x 0,48	x 0,55	x 0,76	**x 2170,46**	**x 59,29**
Reuters S2	FMDI-RF$^+$	x 0,60	x 0,71	x 0,62	x 0,69	x 0,87	**x 2291,52**	**x 57,50**
	FMDI-RF	x 0,66	x 0,68	x 0,63	x 0,69	x 0,87	**x 1519,58**	**x 27,06**
Reuters S3	FMDI-RF$^+$	x 0,60	x 0,72	x 0,60	x 0,68	x 0,85	**x 1979,27**	**x 56,25**
	FMDI-RF	x 0,62	x 0,76	x 0,65	x 0,70	x 0,86	**x 1941,92**	**x 51,14**
Reuters S4	FMDI-RF$^+$	x 0,62	x 0,70	x 0,67	x 0,71	x 0,87	**x 2151,93**	**x 36**
	FMDI-RF	x 0,62	x 0,69	x 0,67	x 0,74	x 0,87	**x 2324,08**	**x 42**
Reuters S5	FMDI-RF$^+$	x 0,65	x 0,64	x 0,75	x 0,76	x 0,89	**x 1894,17**	**x 46,80**
	FMDI-RF	x 0,65	x 0,68	x 0,80	x 0,83	x 0,92	**x 1833,06**	**x 23,40**

TABLE 4 – Les facteurs (x) d'amélioration des performances prédictives pour les critères de qualité majeurs (Ranking Loss (RL), macro-Ranking Loss (macro-RL), Accuracy, F-mesure et Balanced Error Rate (BER)) et les facteurs d'accélération (x) des temps de calcul (Temps d'apprentissage (T. apprentissage) et Temps de prédiction (T. prédiction)) des approches FMDI-RF et FMDI-RF$^+$ par rapport à RF-PCT.

7 Conclusion

Nous avons proposé dans ce chapitre une nouvelle approche d'apprentissage multi-label hybride FMDI-RF qui permet d'effectuer une compression des données puis d'apprendre un modèle prédictif dans les espaces latents. Les deux espaces d'attributs et de labels latents sont appris de manière indépendante à l'aide d'un algorithme de factorisation de matrice efficace qui considère toutes les cellules des matrices exemple-attribut et exemple-label. À partir de la nouvelle représentation de l'ensemble d'apprentissage -obtenue à l'issue de la factorisation des deux matrices-, le classifieur ensemble RF-PCT apprend un modèle de régression et fournit des prédictions qui sont retransférées dans l'espace de labels original à l'aide de la matrice latente

des labels.

Les expérimentations se sont effectuées en deux phases pour dix jeux de données de grande dimension. Nous avons d'abord mesuré les temps de calcul (en minutes) de FMDI-RF pour fournir la meilleure représentation latente de chaque jeu de données. Nous avons ensuite évalué sa qualité de prédiction et sa vitesse de calcul lorsque le nombre d'exemples d'apprentissage est limité. La qualité des prédictions a été évaluée pour cinq critères majeurs : RL, macro-RL, Accuracy, F-mesure et BER et la vitesse de calcul a été évaluée pour deux deux critères : le temps d'apprentissage et le temps de prédiction en secondes. Les résultats obtenus montrent que FMDI-RF permet d'apprendre et de fournir des prédictions aussi précises (voir parfois meilleures pour quelques jeux de données) que celles de RF-PCT en un temps significativement plus court. Néanmoins, son temps de réduction des dimensions n'est pas négligeable pour un système interactif. Nous avons donc proposé une nouvelle approche FMDI-RF$^+$ (relative à FMDI-RF) qui présume qu'il possible d'apprendre de bons facteurs attributs à partir des cellules positives seulement de la matrice exemple-attribut. Nous avons confirmé notre hypothèse avec une deuxième étude expérimentale qui montre que les deux approches FMDI-RF et FMDI-RF$^+$ obtiennent des performances prédictives et des temps d'apprentissage et de prédiction très similaires avec clairement un avantage à FMDI-RF$^+$ qui permet de réduire les dimensions en un temps significativement plus faible. Pour les plus grands jeux d'évaluation (i.e. sous-ensembles de Reuters), FMDI-RF$^+$ permet de réduire les dimensions et d'apprendre un modèle en un temps significativement (de l'ordre de cinq fois) plus court que celui de RF-PCT qui apprend à partir des données originales. Au vu de nos expérimentations numériques, plus les dimensions des données sont grandes, plus FMDI-RF$^+$ semble se dégager en qualité de performance de RF-PCT. À notre connaissance, cette nouvelle approche hybride FMDI-RF+ est l'approche d'apprentissage multi-label qui paraît aujourd'hui la plus prometteuse pour les systèmes d'apprentissage multi-label interactifs.

6

Conclusion et perspectives

Sommaire

1 Résultats

La problématique de cette thèse s'est centrée sur la sélection d'un algorithme de classification multi-label pour un cadre d'apprentissage interactif. Si des études sérieuses ont été menées ces dernières années sur les performances des algorithmes de classification multi-label, la contrainte d'interactivité a été peu étudiée. Elle a été prise en compte dans le développement de systèmes applicatifs mais, à notre connaissance, sans analyse préalable approfondie du choix de l'algo-

rithme. La qualité des résultats fournie est pourtant un facteur important de l'appropriation du système par les utilisateurs et cette confiance doit être établie rapidement, c'est-à-dire dès les premières étapes de l'apprentissage.

Ainsi, dans cette thèse nous avons tenté de proposer un algorithme de classification multi-label qui puisse apprendre « rapidement » avec une croissance très limitée du nombre d'exemples. Outre la recherche bibliographique et la rédaction de l'état de l'art, notre travail s'est organisé en deux étapes principales. Tout d'abord, nous avons éprouvé les algorithmes de classification multi-label proposés dans la littérature dans un contexte simulé d'interactivité. Nous avons mené une comparaison expérimentale étendue (avec douze algorithmes, dix critères d'évaluation de la qualité des prédictions, deux critères d'évaluation de la vitesse de calcul et douze jeux de données) qui nous a permise de retenir le classifieur RF-PCT (Random Forest of Predictive Clustering Trees) initialement préconisé par Madjarov et al. (2012). Ce classifieur obtient de très bonnes performances prédictives tout en restant compétitif avec les classifieurs rapides –qui eux ont une qualité dégradée en contexte interactif-. En effet, RF-PCT doit sa force à l'union des performances d'une centaine d'arbres de décision multi-label complémentaires construits avec un faible coût.

Dans une seconde étape, nous avons cherché à améliorer l'approche pour maintenir un temps de calcul qui reste faible même pour des données de grande dimension. Notre proposition, appelée FMDI-RF (Factorisation de Matrice Duale Indépendante combinée avec le classifieur RF-PCT) est basée sur une combinaison d'une réduction de dimension de données avec un apprentissage par RF-PCT. Une première réduction de dimension a été effectuée par factorisation rapide de matrice -basée sur l'algorithme Gravity- qui nous permet d'apprendre dans les deux espaces latents calculés. Sur des jeux de données de grande dimension (entre 20.000 et 50.000 dimensions), la qualité des résultats obtenus est bonne mais le temps de la réduction de dimension reste trop élevé en pré-traitement. Nous avons donc proposé une autre approche de réduction basée uniquement sur les cellules non nulles de la matrice. Ce choix s'argumente par la composition des matrices de données qui sont -dans le cas des jeux de données multi-label de grande dimension- très creuses. Les résultats expérimentaux obtenus sont très encourageants : cette nouvelle approche (FMDI-RF+) permet de réduire la dimension en temps raisonnable (au maximum deux minutes) avec une qualité de prédiction comparable à RF-PCT et des temps de

prédiction et notamment d'apprentissage bien inférieurs.

2 Application : VIPE

Comme nous l'avons déjà annoncé en introduction, cette thèse se déroule dans un projet dont l'objectif est de développer un système de classification interactive multi-label. En complément de notre analyse du comportement des approches multi-label dans un contexte d'apprentissage interactif, un système prototype de classification interactive multi-label a été développé dés le début de la thèse pour tester essentiellement les différentes approches d'interaction et de visualisation des données. Nous l'appelons *VIPE* (*Visual Interactive and Personalized Exploration of data*). VIPE est multi-utilisateurs et est hébergé dans une application web accessible depuis n'importe quel navigateur. De manière interactive, l'utilisateur crée des concepts cibles et les explique avec un petit ensemble d'exemples et de contre-exemples qui les caractérisent. L'algorithme d'apprentissage intègre en quasi temps-réel ces exemples d'apprentissage et se corrige au fil des itérations. La boucle d'apprentissage se termine lorsque l'utilisateur juge que le moteur de VIPE est satisfaisant. Il peut dés lors l'utiliser pour classer les nouvelles données qui arrivent au quotidien. À Orange labs, deux cas d'utilisations réels sont ciblés pour VIPE : la classification interactive multi-label de films et la classification interactive multi-label de textes courts tels que les tweets (Figures 1 et 2).

Actuellement, VIPE est un démonstrateur : la version actuelle utilise un algorithme d'apprentissage par factorisation rapide de matrice Gravity (défini dans le chapitre 5 -section 3) qui n'est pas l'algorithme cible final. Une étude expérimentale de VIPE avec un groupe de test n'a pas encore été effectuée mais les premiers retours de plusieurs utilisateurs sont encourageants. Par ailleurs, une évaluation objective de son efficacité sur des données multi-label diverses a montré que Gravity est rapide et obtient de bonnes performances lorsque le nombre de dimensions est faible (voir Annexe 3). Cependant, sa capacité d'apprentissage se détériore pour des données de grande dimension. Pour améliorer la qualité de ses prédictions, nous l'avons donc couplé avec l'approche d'apprentissage multi-label la plus efficace RF-PCT. Dans le futur proche, nous envisageons d'intégrer notre nouvelle approche d'apprentissage hybride FMDI-RF+ dans le système VIPE et de mener une étude subjective pour analyser l'avis des utilisateurs. Nous présentons ci-dessous brièvement les deux cas d'utilisation de VIPE : la classification de

films (Figure 1) et la classification de tweets (Figure 2).

2.1 Classification interactive multi-label de films

Pour choisir un « bon » film pour la soirée, les utilisateurs qui n'ont pas de titre précis peuvent être perplexes face au nombre de films disponibles dans les catalogues actuels de *VoD*. Un utilisateur a donc besoin d'être assisté dans son choix de programmes télévisés pour lesquels il va consacrer du temps et dépenser de l'argent. Le système VIPE permet de créer un classifieur de films personnalisé avec l'objectif d'une proposition de films appropriés qui limite les efforts de l'utilisateur. Pour commencer, l'utilisateur définit son ensemble de labels préférés (e.g. Drôle, J'aime, Belle musique, Triste) puis il annote un petit ensemble de films pour transmettre ses préférences à l'algorithme d'apprentissage. Il annote par exemple le film Titanic en positif avec les trois labels : « J'aime », « Belle musique » et « Triste ». S'ajoute à cette annotation une annotation implicite en négatif avec le label « drôle ». En fonction de la requête, l'algorithme d'apprentissage peut prédire les labels les plus probables pour un film sélectionné ou les films les plus probables pour un label ou une combinaison sélectionné(e).

2.2 Classification interactive multi-label de tweets

Avec la popularisation des médias sociaux, l'analyse d'opinions est devenue un enjeu pour les entreprises qui souhaitent améliorer en permanence leur relation client. De nombreux efforts portent aujourd'hui sur le développement de traitements automatiques (e.g. (Liu, 2012)) et l'on voit émerger de nouveaux systèmes de la gestion de la relation client (Ajmera et al., 2013). Mais, comme l'indique une introduction sur la fouille de média sociaux (Gundecha and Liu, 2012), "analyser les opinions qui circulent à propos d'une entreprise [...] est un nouveau défi : les données issues des médias sociaux sont volumineuses, bruitées, distribuées, non-structurées et dynamiques". Ces caractéristiques se retrouvent notamment sur *Twitter* qui, malgré une baisse d'attractivité très récente, reste un site majeur pour la diffusion des opinions. En sus des approches d'analyse automatique, qui ont une capacité à détecter des grandes tendances dans des gros volumes mais plus de difficultés à y capter des « signaux faibles » d'une valeur ajoutée plus grande pour l'entreprise (Pepin et al., 2015), il est utile de développer des systèmes de

classification assistés qui aident un utilisateur –par exemple des agents du service marketing- à organiser les informations. Ainsi, le système VIPE a été aujourd'hui adapté pour aider ces agents à classifier les tweets en détectant les messages les plus pertinents pour leurs analyses. Pour ce faire, l'utilisateur définit d'abord son ensemble de labels désirés (Efficacité, Innovation, Problèmes, Négatif). Il procède ensuite à l'étiquetage d'un petit ensemble de tweets pour exprimer son point de vue. Par exemple, il annote le tweet "ça c'est de la #4G ! :D" en positif avec les deux labels : Efficacité et Innovation, et l'annote implicitement en négatif avec les deux autres labels : Problèmes et Négatif. En se basant sur cet ensemble de messages étiquetés, l'algorithme d'apprentissage assiste l'agent en lui prédisant les labels les plus probables pour un tweet sélectionné ou les tweets les plus probables pour un label ou une combinaison sélectionné(e).

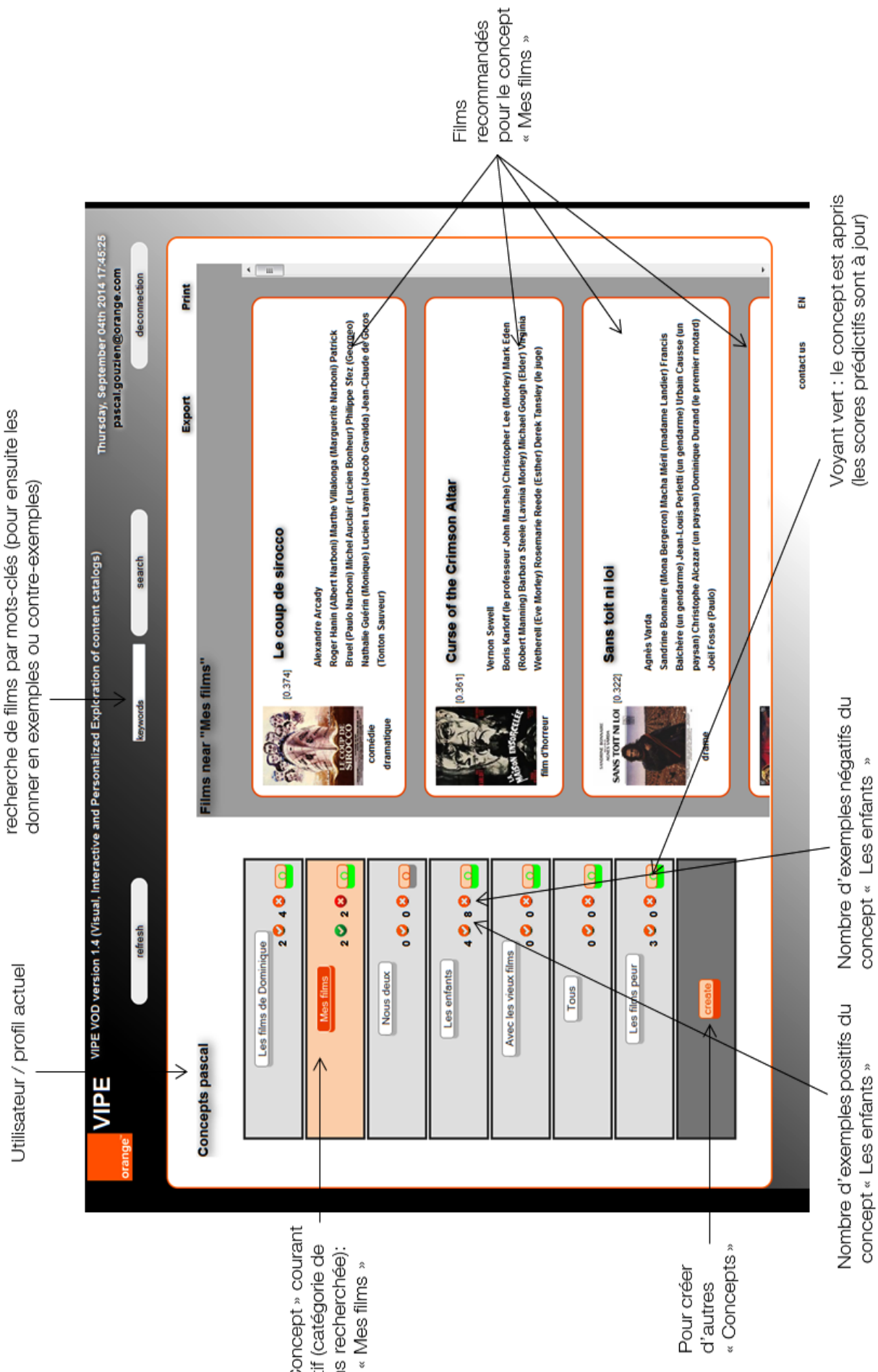


FIGURE 1 – *VIPE* pour la classification interactive multi-label de films.

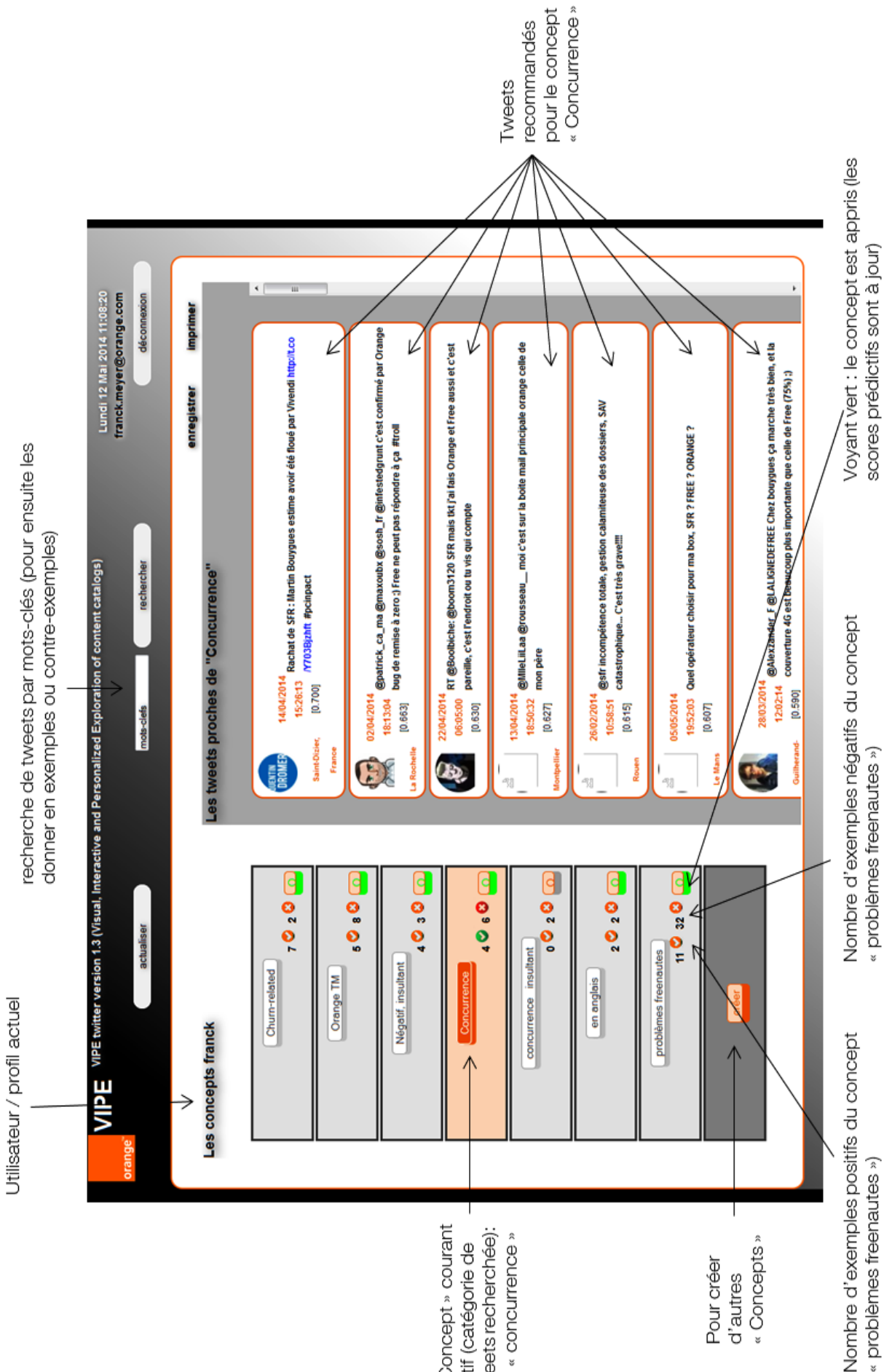


FIGURE 2 – *VIPE* pour la classification interactive multi-label de tweets.

3 Perspectives

Les résultats obtenus dans cette thèse et les retours applicatifs conduisent à de nouvelles recherches dans deux directions principales : l'amélioration de la nouvelle approche FMDI-RF+ pour la classification interactive multi-label et l'extension du protocole expérimental pour la comparaison des algorithmes.

Les derniers résultats que nous avons obtenus avec FMDI-RF+ sont prometteurs mais l'analyse des expérimentations nous donne l'intuition qu'il y a encore des possibilités d'amélioration en « jouant » sur la réduction de dimension. La réduction de dimension n'était pas la problématique centrale de la thèse et nous nous sommes donc appuyés pour choisir une méthode sur un état de l'art assez rapide et sur le savoir-faire de l'équipe de recherche de Orange Labs dans laquelle se sont effectuées ces recherches. Cependant, une étude comparative approfondie vient d'être mise à disposition de la communauté (Pacharawongsakda and Theeramunkong, 2013) et nous prévoyons à court terme de comparer nos résultats avec les meilleures approches de cette étude –notamment IDSR (Independent Dual Space Reduction)-. De plus, de façon générale, il est bien connu que la sélection de la meilleure combinaison de paramètres pour un algorithme de factorisation de matrices est un problème délicat. Nous l'avons simplifié ici en considérant un jeu restreint de combinaisons choisi selon nos premiers retours d'expérience. Cette question est aujourd'hui abordée sous la forme d'un problème d'optimisation pour lequel on peut appliquer des heuristiques de type descente de gradient (Boyd and Vandenberghe, 2004) ou recherche à voisinage variable (Mladenović and Hansen, 1997). Elle ouvre une discussion entre des chercheurs en sciences des données et en optimisation et le laboratoire académique auquel est associé cette thèse (Laboratoire d'Informatique de Nantes Atlantique) qui regroupe des chercheurs de ces deux domaines offre un cadre favorable pour aborder ce problème dans toute sa complexité. En sus de ces aspects, il nous paraît aussi intéressant de comparer la direction que nous avons choisie avec des approches d'apprentissage semi-supervisées. En effet, lorsque la taille de l'ensemble d'apprentissage est limitée, l'information induite à partir des données non étiquetées peut permettre aux classifieurs d'améliorer leurs performances prédictives (Chapelle et al., 2006). Des travaux récents ont permis d'identifier des approches semi-supervisées performantes pour des données multi-label (Kong et al., 2013). Et il nous semble qu'une comparaison

de FMDI-RF+ avec de telles approches aiderait à mieux comprendre le rôle joué par l'information ajoutée dans le processus d'apprentissage et ainsi, par rétroaction, à améliorer notre approche.

Le protocole expérimental que nous avons construit pour comparer les différents algorithmes est très simple. Il visait essentiellement à évaluer l'impact des deux contraintes majeures de l'interactivité sur les approches multi-label : apprentissage avec peu d'exemples et temps de calcul limité. L'analyse des systèmes applicatifs présentés dans la littérature, ainsi que nos propres retours d'usage sur les applications présentées ci-dessus, permettent d'identifier d'autres composantes à prendre en compte pour intégrer l'interactivité. Il est évident que les aspects relevant de l'ergonomie et du design de l'interface sont très importants pour l'adhésion des utilisateurs au système mais ils dépassent largement le cadre de cette thèse. Cependant, sans mettre en œuvre une IHM sophistiquée, il pourrait être possible d'introduire dans une simulation simple la tâche de correction de l'utilisateur et l'ajout de nouveaux labels cibles.

4 Liste des travaux

Revue internationale

Nair-Benrekia N. Y., Kuntz P., Meyer F. (2015). Learning from multi-label data with interactivity constraints : an extensive experimental study. *Expert Systems with Applications*, vol. 42, n° 13, pp. 5223-5736 (Impact Factor : 1.97 – revue classée #1 dans la catégorie Intelligence Artificielle de Google Scholar).

Conférence internationale avec actes

Nair-Benrekia N. Y., Kuntz P., Meyer F. (2014). Selecting a multi-label classification method for an interactive system. *In Data Science, Learning by Latent Structures, and Knowledge Discovery*, B. Lausen, S. Krolak-Schwerdt, M. Böhmer eds, Springer, p. 157-167 (post-actes sélectionnés de la conférence *European Conference on Data Analysis*, Luxembourg, 2013) ISSN 1431-8814.

Conférence nationale avec actes

Nair-Benrekia N. Y., Kuntz P., Meyer F. (2014). Sélection d'une méthode de classification multi-label dans un système intéractif. *Revue des Nouvelles Technologies de l'Information* (*Actes de la conférence Extraction et Gestion des Connaissances*, EGC'14), pp. 305-310.

Divers

Nair-Benrekia N. Y. (2014). **Apprentissage interactif multi-label : une étude expérimentale**. Article accepté pour la *14e Journée des doctorants* organisée par l'école doctorale 'ED STIM'.

Nair-Benrekia N. Y. (2014). Un système de classification intéractif pour l'aide à l'organisation personnalisée de données. Poster présenté à l'*École de Printemps sur l'Apprentissage arTificiel* (EPAT).

Bibliographie

Ajmera, J., Ahn, H.-i., Nagarajan, M., Verma, A., Contractor, D., Dill, S., and Denesuk, M. (2013). A crm system for social media : challenges and experiences. In *Proceedings of the 22nd international conference on World Wide Web*, pages 49–58. International World Wide Web Conferences Steering Committee. 96

Amershi, S. (2011). Designing for effective end-user interaction with machine learning. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, pages 47–50. ACM. 23, 61

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2015). Power to the people : The role of humans in interactive machine learning. *AI Magazine (Accepted and in press)*. 18, 23

Amershi, S., Fogarty, J., and Weld, D. (2012). Regroup : Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM. 9, 19, 24, 27, 29, 31

Amershi, S., Lee, B., Kapoor, A., Mahajan, R., and Christian, B. (2011). Cuet : human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 157–166. ACM. 9, 19, 24, 28, 30, 31

Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. 75

Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM. 18

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern recognition*, 37(9) :1757–1771. 20, 34, 49

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press. 83, 100

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2) :123–140. 43

Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S. F., Hadley, A., Betts, M., Fern, X. Z., et al. (2013). The 9th annual mlsp competition : New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–8. IEEE. 49

Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). *Semi-supervised learning*, volume 2. MIT press Cambridge. 100

Chen, Y.-W. and Lin, C.-J. (2006). Combining svms with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer. 58

Cheng, W. and Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3) :211–225. 42, 56, 64

Cheng, W., Hüllermeier, E., and Dembczynski, K. J. (2010). Graded multilabel classification : The ordinal case. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 223–230. 35

Clare, A. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *Principles of data mining and knowledge discovery*, pages 42–53. Springer. 20, 34

Cockton, G. (2007). Make evaluation poverty history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 31

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3) :273–297. 41

Dabrowski, J. R. and Munson, E. V. (2001). Is 100 milliseconds too fast ? In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pages 317–318. ACM. 59

Diplaris, S., Tsoumakas, G., Mitkas, P. A., and Vlahavas, I. (2005). Protein classification with multiple algorithms. In *Advances in Informatics*, pages 448–456. Springer. 50

Drucker, S. M., Fisher, D., and Basu, S. (2011). Helping users sort faster with adaptive machine learning recommendations. In *Human-Computer Interaction–INTERACT 2011*, pages 187–203. Springer. 9, 19, 24, 25, 27

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687. 49

Fails, J. A. and Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM. 23

Fiebrink, R., Trueman, D., and Cook, P. R. (2009). A metainstrument for interactive, on-the-fly machine learning. In *Proc. NIME*, volume 2, page 3. 9, 19, 24, 25, 26

Fogarty, J., Tan, D., Kapoor, A., and Winder, S. (2008). Cueflik : interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 29–38. ACM. 9, 19, 24, 29, 30, 31

Fürnkranz, J., Hüllermeier, E., Mencía, E. L., and Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2) :133–153. 39, 55, 56

Goodrich, M. A. and Schultz, A. C. (2007). Human-robot interaction : A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3) :203–275. 25

Greenberg, S. and Buxton, B. (2008). Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 111–120. ACM. 31

Gundecha, P. and Liu, H. (2012). Mining social media : a brief introduction. *Tutorials in Operations Research*, 1(4). 96

Katakis, I., Tsoumakas, G., and Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*. 20, 34

Kocev, D. (2011). *Ensembles for predicting structured outputs*. PhD thesis, University of Waikato. 20, 43, 55, 62, 72, 79

Kocev, D., Vens, C., Struyf, J., and Džeroski, S. (2007). Ensembles of multi-objective decision trees. In *Proceedings of the 18th European conference on Machine Learning*, pages 624–631. Springer-Verlag. 20, 43

Kong, X., Ng, M. K., and Zhou, Z.-H. (2013). Transductive multilabel learning via label set propagation. *Knowledge and Data Engineering, IEEE Transactions on*, 25(3) :704–719. 100

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1 : A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5 :361–397. 50

Li, T., Zhang, C., and Zhu, S. (2006). Empirical studies on multi-label classification. In *IcTAI*, volume 6, pages 86–92. 52, 56

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations : Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1) :76–80. 17

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., et al. (2008). Galaxy zoo : morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389(3) :1179–1189. 23

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1) :1–167. 96

Lo, H.-Y., Wang, J.-C., Wang, H.-M., and Lin, S.-D. (2011). Cost-sensitive multi-label learning for audio tag annotation and retrieval. *Multimedia, IEEE Transactions on*, 13(3) :518–529. 20, 34

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Dzeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9) :3084–3104. 20, 21, 34, 35, 41, 44, 52, 55, 56, 58, 62, 63, 64, 70, 94

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24(11) :1097 – 1100. 83, 100

Nasierding, G. and Kouzani, A. Z. (2012). Comparative evaluation of multi-label classification methods. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 679–683. IEEE. 52, 56

Nicolescu, M. N. and Mataric, M. J. (2001). Learning and interacting in human-robot domains. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 31 :419–430. 25

Ozonat, K. and Young, D. (2009). Towards a universal marketplace over the web : Statistical multi-label classification of service provider forms with simulated annealing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1295–1304. ACM. 20, 34

Pacharawongsakda, E. and Theeramunkong, T. (2012). Towards more efficient multi-label classification using dependent and independent dual space reduction. In *Proceedings of the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining-Volume Part II*, pages 383–394. Springer-Verlag. 74

Pacharawongsakda, E. and Theeramunkong, T. (2013). A comparative study on single and dual space reduction in multi-label classification. In *Looking into the Future of Creativity and Decision Support Systems Proceedings of the 8th International Conference on Knowledge, Information and Creativity Support Systems*. 74, 100

Pepin, L., Greffard, N., Kuntz, P., Blanchard, J., Guillet, F., and Suignard, P. (2015). Visual analytics for exploring the topic evolution of company targeted tweets. In *Proc. of the 45th Int. Conf. on Computers & Industrial Engineering CIE 45, to appear*. International World Wide Web Conferences Steering Committee. 96

Porter, R., Theiler, J., and Hush, D. (2013). Interactive machine learning in data exploitation. *Computing in Science & Engineering*, 15(5) :12–20. 23

Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann. 41

Rak, R., Kurgan, L., and Reformat, M. (2005). Multi-label associative classification of medical documents from medline. In *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on*, pages 8–pp. IEEE. 20, 34

Read, J. (2010). *Scalable multi-label classification*. PhD thesis, University of Waikato. 41, 51, 56

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases : Part II*, ECML PKDD '09, page 254–269, Berlin, Heidelberg. Springer-Verlag. 37, 43

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3) :333–359. 49, 50, 55, 56, 58, 62

Ritter, A. and Basu, S. (2009). Learning to generalize for complex selection tasks. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 167–176. ACM. 9, 19, 24, 27, 28, 31

Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5). 24

Schapire, R. E. and Singer, Y. (2000). Boostexter : A boosting-based system for text categorization. *Machine learning*, 39(2-3) :135–168. 20, 36, 56

Shilman, M., Tan, D. S., and Simard, P. (2006). Cuetip : a mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 323–332. ACM. 9, 24, 25, 26, 31

Snoek, C. G., Worring, M., Van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430. ACM. 20, 34

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*. 20, 34

Srivastava, A. N. and Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace Conference, 2005 IEEE*, pages 3853–3862. IEEE. 50

Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. (2007). Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91. ACM. 24

Takacs, G., Pilaszy, I., Nemeth, B., and Tikk, D. (2007). On the gravity recommendation system. In *Proceedings of KDD cup and workshop*, volume 2007. 75, 77

Tawiah, C. A. and Sheng, V. S. (2013). A study on multi-label classification. In *Advances in Data Mining. Applications and Theoretical Aspects*, pages 137–150. Springer. 41, 52, 56

Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. (2008). Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330. 49

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification : An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3) :1–13. 20, 34, 38, 50, 56

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. 39, 56, 64

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer. 20, 34, 35, 54

Tsoumakas, G. and Vlahavas, I. (2007). Random k-labelsets : An ensemble method for multilabel classification. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, page 406–417, Berlin, Heidelberg. Springer-Verlag. 42, 62

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584. 18

Ware, M., Frank, E., Holmes, G., Hall, M., and Witten, I. H. (2001). Interactive machine learning : letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3) :281–292. 23

Witten, I. H. and Frank, E. (2005). *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann. 25

Yu, K., Yu, S., and Tresp, V. (2005). Multi-label informed latent semantic indexing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265. ACM. 20, 34

Zhai, S. (2003). Evaluation is the worst form of hci research except all those other forms that have been tried. *CHI Place Essay*. 31

Zhang, M. and Zhou, Z. (2013). A review on multi-label learning algorithms. 20, 34, 35, 44, 54

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn : A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7) :2038–2048. 41, 55, 56

7

Annexes

Sommaire

1 Annexe 1

Les résultats détaillés obtenus pour les trois tailles représentatives des ensembles d'apprentissage (4, 16 et 64 exemples) pour les critères de qualité majeurs (Ranking Loss et macro-averaged Ranking Loss) sont présentés ci-dessous.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,45	**0,44**	0,46	0,45	0,45	0,50	0,45	0,50	0,48	0,48	0,45	0,47	0,46
Yeast	0,37	0,32	0,32	0,30	0,30	0,30	0,36	0,32	**0,28**	0,29	0,31	**0,28**	**0,28**
Scene	0,50	**0,49**	0,50	0,50	0,50	0,50	0,51	0,50	0,50	0,50	0,50	0,50	**0,49**
Birds	0,26	0,25	0,24	0,25	0,25	0,24	0,26	0,24	0,24	0,25	0,24	0,24	**0,23**
Slashdot	0,57	0,49	0,49	0,52	0,56	0,49	0,59	0,49	**0,48**	0,49	0,49	**0,48**	**0,48**
IMDB	0,55	0,41	0,38	0,45	0,49	0,36	0,55	0,36	0,36	0,39	0,38	**0,35**	**0,35**
Genbase	0,29	**0,27**	0,28	0,29	0,32	0,29	0,42	0,29	0,28	**0,27**	**0,27**	**0,27**	**0,27**
Arts	0,40	0,40	0,39	0,40	0,42	0,38	0,46	**0,37**	**0,37**	0,39	0,39	**0,37**	**0,37**
Business	0,11	0,12	0,10	0,11	0,11	0,09	0,20	0,09	**0,08**	0,09	0,10	0,09	**0,08**
Health	0,28	0,26	0,24	0,26	0,27	0,22	0,34	0,22	**0,21**	0,24	0,25	0,22	**0,21**
Computers	0,23	0,25	0,27	0,25	0,27	0,23	0,33	0,23	**0,22**	0,23	0,27	0,23	0,23
TMC	0,42	0,33	0,31	0,32	0,35	**0,28**	0,42	**0,28**	**0,28**	0,29	0,31	**0,28**	**0,28**

TABLE 1 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 4 pour le critère Ranking Loss et pour chaque jeu de données.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,46	0,38	0,38	0,34	0,34	0,38	0,39	0,40	0,33	0,31	0,37	0,31	**0,29**
Yeast	0,31	0,33	0,40	0,29	0,29	0,25	0,35	0,34	0,26	0,27	0,35	0,27	**0,24**
Scene	0,49	0,44	0,41	0,39	0,39	0,36	0,45	0,40	0,44	0,39	0,41	0,37	**0,34**
Birds	0,26	0,24	0,23	0,23	0,23	**0,20**	0,26	0,24	0,21	0,22	0,23	0,21	**0,20**
Slashdot	0,53	0,40	0,37	0,48	0,53	0,36	0,55	0,37	0,35	0,51	0,37	0,35	**0,34**
IMDB	0,46	0,39	0,29	0,42	0,50	**0,24**	0,55	0,27	**0,24**	0,41	0,28	0,25	**0,24**
Genbase	0,27	0,15	0,16	0,17	0,18	0,15	0,22	0,19	0,21	**0,11**	0,16	0,14	0,18
Arts	0,39	0,41	0,36	0,38	0,40	**0,27**	0,51	0,29	0,28	0,38	0,36	0,30	**0,27**
Business	0,10	0,10	0,10	0,10	0,10	**0,07**	0,17	0,08	**0,07**	0,09	0,10	0,08	**0,07**
Health	0,25	0,24	0,22	0,23	0,24	**0,15**	0,39	0,16	**0,15**	0,21	0,22	0,16	**0,15**
Computers	0,21	0,22	0,22	0,22	0,23	**0,17**	0,35	0,18	**0,17**	0,21	0,23	0,19	**0,17**
TMC	0,38	0,32	0,31	0,29	0,30	0,22	0,41	0,25	0,22	0,26	0,32	0,24	**0,21**

TABLE 2 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 16 pour le critère Ranking Loss et pour chaque jeu de données.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,46	0,33	0,35	0,25	0,25	0,29	0,34	0,28	0,25	0,23	0,32	0,23	**0,19**
Yeast	0,28	0,31	0,42	0,25	0,25	0,21	0,33	0,26	0,23	0,23	0,34	0,23	**0,20**
Scene	0,49	0,30	0,33	0,24	0,23	0,25	0,35	0,25	0,21	0,22	0,34	0,22	**0,14**
Birds	0,26	0,20	0,21	0,21	0,21	0,16	0,24	0,22	0,16	0,18	0,21	0,17	**0,15**
Slashdot	0,52	0,34	0,26	0,41	0,46	0,27	0,56	0,28	0,24	0,41	0,26	0,24	**0,22**
IMDB	0,41	0,40	0,25	0,39	0,48	**0,20**	0,57	0,23	**0,20**	0,39	0,25	0,22	0,21
Genbase	0,27	**0,03**	0,04	0,05	0,05	**0,03**	0,05	0,04	0,06	**0,03**	**0,03**	**0,03**	**0,03**
Arts	0,39	0,36	0,28	0,34	0,36	0,22	0,49	0,26	0,22	0,31	0,27	0,23	**0,20**
Business	0,10	0,09	0,08	0,09	0,09	**0,06**	0,16	0,07	**0,06**	0,08	0,09	**0,06**	**0,06**
Health	0,25	0,21	0,17	0,20	0,20	0,12	0,37	0,15	0,12	0,18	0,18	0,13	**0,11**
Computers	0,21	0,21	0,18	0,20	0,21	**0,13**	0,38	0,15	**0,13**	0,19	0,20	0,14	**0,13**
TMC	0,36	0,29	0,29	0,24	0,25	0,18	0,38	0,22	0,17	0,21	0,29	0,18	**0,15**

TABLE 3 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 64 pour le critère Ranking Loss et pour chaque jeu de données.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,45	0,49	0,45	0,45	0,50	0,47	0,52	0,48	0,48	0,48	0,48	**0,40**
Yeast	**0,48**	0,49	**0,48**	0,49	0,49	0,49	0,49	0,50	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Scene	0,50	0,49	0,50	0,49	0,49	0,49	0,50	0,51	0,50	0,50	0,50	0,50	**0,48**
Birds	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,50	0,49	0,49	0,49	0,49	**0,48**
Slashdot	0,46	0,46	0,46	0,46	0,46	0,46	0,47	0,47	0,46	0,46	0,46	0,46	**0,44**
IMDB	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Genbase	0,48	0,47	0,48	0,47	0,47	0,46	0,47	0,52	0,48	0,44	0,48	0,44	**0,41**
Arts	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,49	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Business	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**	0,45	0,45	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**
Health	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	**0,48**
Computers	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**
TMC	0,51	0,51	0,51	0,51	0,51	0,51	0,50	0,51	0,51	0,51	0,51	0,51	**0,49**

TABLE 4 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 4 pour le critère macro-averaged Ranking Loss et pour chaque jeu de données.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,41	0,41	0,34	0,34	0,41	0,41	0,41	0,32	0,30	0,40	0,30	**0,23**
Yeast	0,48	0,48	0,47	0,48	0,47	0,48	0,49	0,49	0,47	0,46	0,47	0,46	**0,45**
Scene	0,50	0,43	0,42	0,37	0,37	0,37	0,45	0,38	0,41	0,35	0,42	0,35	**0,25**
Birds	0,49	0,46	0,48	0,45	0,46	0,46	0,43	0,50	0,47	0,44	0,48	0,44	**0,40**
Slashdot	0,46	0,43	0,45	0,44	0,45	0,46	0,45	0,47	0,44	0,44	0,45	0,44	**0,39**
IMDB	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,50	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Genbase	0,48	0,35	0,40	0,38	0,39	0,32	0,32	0,40	0,40	0,32	0,40	0,32	**0,29**
Arts	0,48	0,48	0,48	0,48	0,48	0,48	0,50	0,50	0,48	0,48	0,48	0,48	**0,46**
Business	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,46	0,44	0,44	0,44	0,44	**0,42**
Health	0,49	0,48	0,48	0,47	0,48	0,48	0,48	0,49	0,48	0,47	0,48	0,47	**0,45**
Computers	0,50	0,50	0,50	0,50	0,50	0,50	0,49	0,50	0,50	0,50	0,50	0,50	**0,48**
TMC	0,51	0,50	0,50	0,48	0,48	0,50	0,50	0,50	0,49	0,49	0,50	0,49	**0,43**

TABLE 5 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 16 pour le critère macro-averaged Ranking Loss et pour chaque jeu de données.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,35	0,37	0,26	0,25	0,32	0,37	0,29	0,25	0,23	0,34	0,23	**0,18**
Yeast	0,48	0,46	0,46	0,44	0,43	0,45	0,47	0,46	0,43	0,42	0,46	0,42	**0,39**
Scene	0,50	0,31	0,34	0,23	0,23	0,28	0,35	0,25	0,19	0,22	0,34	0,22	**0,10**
Birds	0,49	0,39	0,45	0,39	0,40	0,40	0,46	0,50	0,41	0,36	0,45	0,36	**0,26**
Slashdot	0,46	0,40	0,43	0,40	0,41	0,46	0,44	0,46	0,38	0,40	0,43	0,40	**0,32**
IMDB	0,48	0,48	0,48	0,48	0,48	0,48	0,49	0,49	0,48	0,48	0,48	0,48	**0,47**
Genbase	0,48	0,22	0,27	0,25	0,25	0,20	0,22	0,22	0,25	0,19	0,26	0,19	**0,17**
Arts	0,48	0,47	0,47	0,46	0,46	0,48	0,48	0,49	0,45	0,45	0,47	0,45	**0,42**
Business	0,44	0,43	0,44	0,43	0,43	0,44	0,46	0,47	0,43	0,42	0,44	0,42	**0,37**
Health	0,49	0,46	0,47	0,44	0,44	0,48	0,47	0,49	0,44	0,43	0,47	0,43	**0,38**
Computers	0,50	0,49	0,50	0,48	0,48	0,50	0,49	0,49	0,49	0,48	0,49	0,48	**0,43**
TMC	0,51	0,47	0,46	0,43	0,43	0,49	0,48	0,47	0,39	0,42	0,46	0,42	**0,29**

TABLE 6 – Les performances moyennes de chaque classifieur pour tous les ensembles d'apprentissage de taille 64 pour le critère macro-averaged Ranking Loss et pour chaque jeu de données.

2 Annexe 2

Les performances prédictives moyennes des $top(4)$ classifieurs pour chaque taille d'ensemble d'apprentissage pour les critères de qualité supplémentaires : One-error, Hamming loss, Average precision, Exact match et Coverage sont présentées ci-dessous.

	2	4	8	16	32	64
MLkNN	$0,73 \pm 0,20$	$0,68 \pm 0,23$	**0,60 ± 0,22**	**0,54 ± 0,23**	**0,50 ± 0,25**	$0,48 \pm 0,26$
CLR	**0,69 ± 0,22**	$0,66 \pm 0,23$	$0,62 \pm 0,23$	$0,58 \pm 0,22$	$0,52 \pm 0,22$	$0,48 \pm 0,23$
EBR	$0,72 \pm 0,19$	$0,67 \pm 0,21$	**0,60 ± 0,22**	$0,55 \pm 0,23$	$0,51 \pm 0,24$	$0,47 \pm 0,24$
RF-PCT	$0,72 \pm 0,19$	**0,65 ± 0,22**	$0,61 \pm 0,23$	$0,56 \pm 0,22$	**0,50 ± 0,21**	**0,45 ± 0,23**

TABLE 7 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère One-error.

	2	4	8	16	32	64
MLkNN	**0,19 ± 0,12**	$0,20 \pm 0,12$	$0,17 \pm 0,11$	$0,15 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$
CLR	**0,19 ± 0,12**	$0,18 \pm 0,12$	$0,16 \pm 0,11$	$0,16 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$
EBR	**0,19 ± 0,12**	$0,20 \pm 0,11$	$0,16 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$	$0,12 \pm 0,08$
RF-PCT	**0,19 ± 0,12**	**0,16 ± 0,11**	**0,14 ± 0,10**	**0,13 ± 0,08**	**0,11 ± 0,07**	**0,10 ± 0,07**

TABLE 8 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Hamming loss.

	2	4	8	16	32	64
MLkNN	$0,42 \pm 0,16$	$0,45 \pm 0,16$	**0,52 ± 0,15**	**0,57 ± 0,16**	$0,60 \pm 0,17$	$0,63 \pm 0,18$
CLR	**0,44 ± 0,16**	**0,47 ± 0,17**	$0,50 \pm 0,16$	$0,54 \pm 0,15$	$0,59 \pm 0,15$	$0,63 \pm 0,16$
EBR	$0,42 \pm 0,16$	$0,46 \pm 0,16$	**0,52 ± 0,16**	$0,56 \pm 0,16$	$0,60 \pm 0,17$	$0,63 \pm 0,17$
RF-PCT	$0,43 \pm 0,16$	**0,47 ± 0,16**	**0,52 ± 0,16**	$0,56 \pm 0,15$	**0,62 ± 0,15**	**0,66 ± 0,16**

TABLE 9 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Average precision.

	2	4	8	16	32	64
MLkNN	$0,04 \pm 0,03$	$0,04 \pm 0,03$	$0,06 \pm 0,05$	$0,06 \pm 0,05$	$0,07 \pm 0,05$	$0,09 \pm 0,09$
CLR	**0,05 ± 0,03**	$0,04 \pm 0,03$	$0,05 \pm 0,04$	$0,04 \pm 0,04$	$0,06 \pm 0,04$	$0,08 \pm 0,05$
EBR	$0,04 \pm 0,03$	$0,04 \pm 0,04$	$0,10 \pm 0,07$	$0,13 \pm 0,11$	$0,16 \pm 0,12$	$0,20 \pm 0,13$
RF-PCT	**0,05 ± 0,03**	**0,10 ± 0,08**	**0,13 ± 0,09**	**0,17 ± 0,11**	**0,21 ± 0,14**	**0,26 ± 0,18**

TABLE 10 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Exact match.

	2	4	8	16	32	64
MLkNN	$8,32 \pm 3,92$	$7,74 \pm 3,35$	$6,89 \pm 2,81$	$6,04 \pm 2,48$	$5,33 \pm 2,36$	$4,79 \pm 2,32$
CLR	**8,26 ± 3,93**	**7,61 ± 3,34**	$6,89 \pm 2,76$	$6,13 \pm 2,42$	$5,41 \pm 2,29$	$4,80 \pm 2,31$
EBR	$8,30 \pm 3,92$	$7,66 \pm 3,31$	$6,91 \pm 2,85$	$6,17 \pm 2,65$	$5,46 \pm 2,57$	$4,91 \pm 2,54$
RF-PCT	$8,31 \pm 3,92$	$7,62 \pm 3,32$	**6,81 ± 2,78**	**5,94 ± 2,47**	**5,12 ± 2,36**	**4,48 ± 2,38**

TABLE 11 – Les performances moyennes des $top(4)$ classifieurs sur tous les jeux de données pour chaque taille d'ensemble d'apprentissage pour le critère Coverage.

3 Annexe 3

Les performances moyennes du classifieur Gravity pour chaque taille d'ensemble d'apprentissage pour les critères de qualité majeurs sont présentées ci-dessous. Les expérimentations sont effectuées sur cinq jeux de données de tailles croissantes : Emotions, Yeast, Scene, Slashdot, IMDB et Arts. Pour toutes ces évaluations, le nombre de facteurs a été fixé à 16, le pas d'apprentissage à 0.03 et la régularisation à 0.008. En effet, plusieurs combinaisons de paramètres ont été testées et ce jeu de paramètres nous permet d'obtenir les meilleures performances.

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,48	0,49	0,22	0,29	0,48	0,12	0,00
4	0,47	0,48	0,24	0,31	0,47	0,12	0,00
8	0,45	0,45	0,26	0,33	0,45	0,14	0,00
16	0,36	0,35	0,33	0,42	0,39	0,24	0,01
32	0,27	0,26	0,42	0,51	0,32	0,43	0,01
64	0,23	0,23	0,46	0,55	0,29	0,67	0,01

TABLE 12 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données Emotions pour les cinq critères de qualité majeurs : RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,35	0,53	0,32	0,44	0,38	0,21	0,01
4	0,28	0,52	0,36	0,49	0,35	0,23	0,01
8	0,26	0,52	0,39	0,51	0,33	0,23	0,01
16	0,24	0,50	0,41	0,54	0,31	0,38	0,01
32	0,22	0,46	0,44	0,56	0,29	0,66	0,01
64	0,20	0,42	0,47	0,59	0,27	1,57	0,01

TABLE 13 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données Yeast pour les cinq critères de qualité majeurs : RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,50	0,52	0,12	0,14	0,50	0,31	0,01
4	0,48	0,47	0,16	0,18	0,48	0,36	0,01
8	0,40	0,36	0,23	0,26	0,43	0,51	0,01
16	0,27	0,22	0,34	0,37	0,35	0,82	0,01
32	0,20	0,17	0,41	0,45	0,30	1,26	0,01
64	0,17	0,15	0,46	0,50	0,27	1,85	0,01

TABLE 14 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données Scene pour les cinq critères de qualité majeurs : RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,49	0,43	0,02	0,02	0,50	0,87	0,02
4	0,48	0,44	0,02	0,03	0,50	0,91	0,02
8	0,48	0,42	0,02	0,03	0,50	1,01	0,02
16	0,46	0,44	0,02	0,03	0,50	1,26	0,02
32	0,43	0,46	0,02	0,03	0,49	1,73	0,02
64	0,37	0,46	0,04	0,05	0,48	2,86	0,02

TABLE 15 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données Slashdot pour les cinq critères de qualité majeurs : RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,56	0,47	0,01	0,02	0,50	0,79	0,02
4	0,53	0,47	0,01	0,02	0,50	0,86	0,02
8	0,50	0,49	0,01	0,02	0,50	0,94	0,02
16	0,41	0,47	0,03	0,04	0,48	1,17	0,02
32	0,32	0,48	0,05	0,07	0,47	1,65	0,02
64	0,23	0,47	0,11	0,15	0,42	2,72	0,02

TABLE 16 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données IMDB pour les cinq critères de qualité majeurs : RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

	RL	macro-RL	Accuracy	F-mesure	BER	T. App	T. Test
2	0,50	0,50	0,01	0,02	0,50	16,61	0,66
4	0,50	0,50	0,01	0,02	0,50	19,64	0,66
8	0,50	0,50	0,02	0,02	0,50	22,50	0,67
16	0,50	0,50	0,02	0,03	0,50	29,49	0,66
32	0,50	0,50	0,02	0,03	0,50	44,35	0,66
64	0,50	0,50	0,03	0,04	0,50	71,72	0,66

TABLE 17 – Les performances moyennes de Gravity pour chaque taille d'ensemble d'apprentissage du jeu de données Arts pour les cinq critères de qualité majeurs :RL, macro-RL, F-mesure et BER, et pour les deux critères de vitesse : Temps d'apprentissage et de prédiction (T. App et T. Test).

4 Annexe 4

Les performances prédictives moyennes de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF+ pour les critères de qualité supplémentaires : One-error, Hamming loss, Average precision, Exact match et Coverage sont présentées ci-dessous.

		Exact match	**Coverage**	**Average P**	**H Loss**	**One error**
Arts	FMDI-RF$^+$	0,06	7,61	0,40	0,11	0,77
	FMDI-RF	0,06	7,62	0,41	0,11	0,77
	RF-PCT	0,10	6,24	0,48	0,10	0,68
Business	FMDI-RF$^+$	0,41	2,90	0,87	0,04	0,12
	FMDI-RF	0,37	2,96	0,87	0,04	0,12
	RF-PCT	0,40	2,74	0,87	0,04	0,12
Health	FMDI-RF$^+$	0,24	4,78	0,60	0,08	0,47
	FMDI-RF	0,18	4,71	0,61	0,08	0,47
	RF-PCT	0,26	3,94	0,66	0,07	0,44
Computers	FMDI-RF$^+$	0,36	5,63	0,60	0,05	0,47
	FMDI-RF	0,36	5,64	0,60	0,05	0,47
	RF-PCT	0,27	5,23	0,62	0,05	0,46
TMC	FMDI-RF$^+$	0,03	7,44	0,55	0,12	0,44
	FMDI-RF	0,03	7,44	0,55	0,12	0,44
	RF-PCT	0,09	5,52	0,63	0,09	0,41
Reuters s_1	FMDI-RF$^+$	0	44,36	0,25	0,04	0,75
	FMDI-RF	0	40,69	0,24	0,05	0,78
	RF-PCT	0,03	27,25	0,41	0,04	0,60
Reuters s_2	FMDI-RF$^+$	0	44,36	0,25	0,04	0,75
	FMDI-RF	0	39,72	0,25	0,04	0,75
	RF-PCT	0,02	29	0,38	0,04	0,69
Reuters s_3	FMDI-RF$^+$	0	45,63	0,24	0,04	0,77
	FMDI-RF	0	43,56	0,26	0,04	0,76
	RF-PCT	0,03	29,30	0,38	0,04	0,69
Reuters s_4	FMDI-RF$^+$	0	36,82	0,27	0,04	0,78
	FMDI-RF	0	36,84	0,28	0,04	0,76
	RF-PCT	0,04	25,73	0,40	0,03	0,71
Reuters s_5	FMDI-RF$^+$	0	41,90	0,28	0,04	0,76
	FMDI-RF	0	41,38	0,29	0,04	0,75
	RF-PCT	0,02	29,32	0,38	0,04	0,68

TABLE 18 – Les performances prédictives de RF-PCT et de nos deux approches FMDI-RF et FMDI-RF$^+$ pour les critères de qualité suppleméntaires : Exact match, Coverage, Average Precision, Hamming Loss et One error

l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Noureddine Yassine NAIR BENREKIA

Classification interactive multi-label pour l'aide à l'organisation personnalisée des données

Interactive multi-label classification for the personalized organisation of data

Résumé

L'importance croissante donnée actuellement aux contenus personnalisés a conduit au développement de plusieurs systèmes de classification interactive pour diverses applications originales. Néanmoins, tous ces systèmes recourent à une classification mono-label des items qui limite fortement l'expressivité de l'utilisateur. Le problème majeur commun à tous les développeurs d'un système de classification interactif et multi-label est : quel classifieur multi-label devrions-nous choisir ? Les évaluations expérimentales des systèmes d'apprentissage interactifs récents sont essentiellement subjectives. L'importance de leurs conclusions est donc limitée. Pour tirer des conclusions plus générales qui permettent de guider la sélection de l'algorithme d'apprentissage approprié lors du développement d'un tel système, nous étudions de manière approfondie l'impact des contraintes d'interactivité majeures (apprentissage à partir de peu d'exemples en un temps limité) sur les performances prédictives et les temps de calcul des classifieurs. Les expérimentations mettent en évidence le potentiel d'une approche d'apprentissage ensemble *Random Forest of Predictive Clustering Trees* (RF-PCT). Cependant, la forte contrainte sur le temps de calcul posée par l'interactivité, nous a conduits à proposer une nouvelle approche d'apprentissage hybride FMDI-RF$^+$ qui associe RF-PCT avec une approche de factorisation de matrice efficace pour la réduction de dimensions. Les résultats expérimentaux indiquent que FMDI-RF$^+$ est aussi précise que RF-PCT dans les prédictions avec clairement un avantage à FMDI-RF$^+$ pour la vitesse de calcul.

Mots clés

Apprentissage interactif, apprentissage multi-label, étude comparative, réduction des dimensions.

Abstract

The growing importance given today to personalized contents led to the development of several interactive classification systems for various novel applications. Nevertheless, all these systems use a single-label item classification which greatly constrains the user's expressiveness. The major problem common to all developers of an interactive multi-label system is: which multi-label classifier should we choose? Experimental evaluations of recent interactive learning systems are mainly subjective. The importance of their conclusions is consequently limited. To draw more general conclusions for guiding the selection of a suitable learning algorithm during the development of such a system, we extensively study the impact of the major interactivity constraints (learning from few examples in a limited time) on the classifier predictive and time-computation performances. The experiments demonstrate the potential of an ensemble learning approach *Random Forest of Predictive Clustering Trees* (RF-PCT). However, the strong constraint imposed by the interactivity on the computation time has led us to propose a new hybrid learning approach FMDI-RF$^+$ which associates RF-PCT with an efficient matrix factorization approach for dimensionality reduction. The experimental results indicate that RF-FMDI$^+$ is as accurate as RF-PCT in the predictions with a significant advantage to FMDI-RF $^+$ for the speed of computation.

Key Words

Interactive learning, multi-label learning, comparative study, dimensionality reduction.