

THÈSE

En vue de l'obtention du Diplôme de Doctorat

Présenté par : Melle. BENYETTOU Assia

Intitulé

Contribution en apprentissage semi-supervisé
sous contexte multi-label

Faculté : Mathématiques et Informatique

Département : Informatique

Spécialité : Informatique

Option : Reconnaissance des formes et intelligence Artificielle

Devant le Jury Composé de :

<i>Membres de Jury</i>	<i>Grade</i>	<i>Qualité</i>	<i>Domiciliation</i>
<i>Mr. DJEBBAR Bachir</i>	<i>Prof.</i>	<i>Président</i>	<i>USTO-MB</i>
<i>Mr. BENYETTOU Abdelkader</i>	<i>Prof.</i>	<i>Encadrant</i>	<i>USTO-MB</i>
<i>Mr. BENNANI Younès</i>	<i>Prof.</i>	<i>Co-Encadrant</i>	<i>Univ. Paris13-LIPN</i>
<i>Mr. CHIKH Mohammed Amine</i>	<i>Prof.</i>		<i>Univ. Tlemcen</i>
<i>Mr. RAHMOUN Abdellatif</i>	<i>Prof.</i>	<i>Examineurs</i>	<i>ESI. Sidi Bel-Abbès</i>
<i>Mr. RAHAL Sid-Ahmed</i>	<i>Prof.</i>		<i>USTO-MB</i>

Année Universitaire : 2017-2018

Remerciements

Je remercie Dieu tout puissant pour la santé, la volonté, le courage et la patience qu'Il m'a donnés durant ces années d'étude.

Je souhaite exprimer toute ma reconnaissance à mes parents qui m'ont soutenue tout au long de ces années et m'ont toujours encouragée à faire ce que je souhaitais et à donner le meilleur de moi-même.

Le travail de recherche exposé dans ce mémoire a été réalisé conjointement au sein du Laboratoire Signal-Image-Parole (SIMPA) de l'USTO-MB et au sein du Laboratoire d'Informatique de Paris Nord -LIPN, Paris 13.

Je tiens tout d'abord à remercier mon co-directeur de thèse, Prof. Younès BENNANI, Laboratoire d'Informatique de Paris Nord -LIPN, Paris 13. Tout au long de ces années, j'ai pu apprécier sa vision, son intégrité, sa disponibilité, sa générosité à partager ses connaissances. Je voudrais lui témoigner ici toute ma gratitude pour m'avoir guidée durant ces années et laissée entrevoir ce que le mot recherche veut dire. Qu'il trouve ici l'expression de ma profonde reconnaissance pour avoir su inspirer et diriger avec un intérêt constant ce travail.

Je remercie chaleureusement Prof. Abdelkader BENYETTOU pour avoir accepté d'être mon encadrant de cette thèse, pour son soutien, ses nombreux conseils et l'intérêt qu'il a porté à mon travail tout au long de ces années. Je lui suis très reconnaissante de sa patience tout au long de ces années de thèse.

Je suis profondément reconnaissante à Mr. Prof. Bachir DJEBBAR, Doyen de la Faculté des Mathématiques et d'Informatique, pour l'honneur qu'il me fait en acceptant de présider ce jury ainsi que pour son soutien de toujours.

Je présente mes respects et mes remerciements aux membres du jury qui m'ont fait l'honneur d'évaluer ce travail, les Professeurs MM. Mohammed Amine CHIKH de l'Université Abou-Bakr Belkaid, Abdellatif RAHMOUN de l'Ecole Supérieure d'Informatique –ESI de Sidi Bel-Abbès et Sid-Ahmed RAHAL de l'USTO-MB.

Plus largement, je remercie tous mes enseignants qui ont contribué de près ou de loin à l'aboutissement de mon parcours d'étudiante. Qu'ils trouvent ici mes sincères vœux.

Mes sincères remerciements aussi à toute l'équipe LIPN de l'université de Paris 13, qui m'a chaleureusement accueillie pendant mes stages. Spécialement, Monsieur Guénaël CABANES pour sa générosité, sa disponibilité et ses encouragements.

Merci à mon équipe du laboratoire SIMPA, à mes collègues doctorants et plus précisément à Monsieur BENDAHDANE Abderrahmane, qui était aussi l'un de mes enseignants, pour l'exemple qui nous donne tous à travers son partage, sa bonté, son optimisme et son sérieux, je tiens à lui présenter ici ma gratitude et mon estime.

Merci aussi à ma sœur Souhila et à mes frères Youcef et Houcine qui m'ont toujours soutenue avec une grande détermination.

Merci à mes amis, Yacine, Nada, Wafàa et Karima qui m'ont été d'un soutien moral tout au long de ces longues années d'amitié.

A

Mon père, Mon encadrant de thèse et de vie,

Ma mère, Mon amie et mon remède,

Vous êtes Source de ma Force et de ma Joie, je ne vous remercierai jamais assez ...

Résumé

La classification multi-label est de plus en plus répandue en tant que technique de fouille de données. Son objectif est de classer les modèles dans plusieurs groupes non exclusifs et est appliqué dans des domaines tels que la catégorisation des nouvelles, l'étiquetage des images et la classification de la musique, entre autres. Notre contribution est d'utiliser le paradigme de l'apprentissage actif avec le pouvoir topologique de la carte SOM pour la classification semi-supervisée multi-label, en tenant compte des informations multi-labels, et en sélectionnant des données non étiquetées qui peuvent conduire à la plus grande réduction attendue de la perte de modèle.

Ce travail de thèse concerne principalement la classification multi-label semi-supervisée par l'apprentissage actif topologique et traite de divers ensembles de données multi-label en présentant dans l'apprentissage actif, un ensemble de résultats allant de:

- 1) Classifieur transductif TSVM avec des méthodes d'échantillonnage pertinentes pour les données multi-étiquetées dans différents domaines d'application;
- 2) Classifieur proposé semi-supervisé Act-SOM dans l'apprentissage actif multi-label, en adoptant une stratégie relative à l'évaluation par l'incertitude des labels.

Act-SOM basé sur l'apprentissage actif sélectionne les données les plus incertaines tout en améliorant nettement le taux de test avec moins de 30% des instances marquées ajoutées, ce qui constitue notre principale contribution. Nous présentons les résultats des tests statistiques à l'aide de diagrammes critiques. Ainsi, le potentiel de la méthode de classification multi-label proposée est démontré, principalement en raison des propriétés concurrentielles avec la cohérence globale de l'Act-SOM semi-supervisée par l'apprentissage actif topologique.

Mots-clés: Apprentissage multi-label, apprentissage actif, SOM, TSVM, Stratégie d'incertitude sur les labels



Abstract

Multi-label classification is becoming increasingly widespread as a data mining technique. Its objective is to categorize models in several non-exclusive groups, and is applied in such areas as news categorization, image labeling and music classification, among others. Our contribution is to use the paradigm of active learning with the topological power of the Act-SOM for semi-supervised multi-label classification, taking into account the multi-label information, and selecting unlabeled data which can lead to the largest reduction of the expected model loss.

This work of thesis mainly concerns semi-supervised multi-label classification through topological active learning and deals with various multi-label datasets by presenting in active learning, a set of results ranging from:

- 1) Transductive classifier TSVM with relevance sampling methods for multi-labeled data in various application domains;
- 2) Proposed semi-supervised classifier Act-SOM in multi-label active learning, adopting a strategy relative to the evaluation by the uncertainty of the labels.

Act-SOM based on Active learning selects the most uncertain data while clearly improving the test rate with less than 30% of labeled instances added, which is our main contribution. We present the results from the statistical tests using critical diagrams. Thus, potential of the proposed multi-label classification method is demonstrate, due mainly to the competitive properties with global consistency of the semi-supervised Act-SOM through topological active learning.

Keywords: Multi-label Learning, Active Learning, SOM, TSVM, Label Uncertainty Strategy

Table des matières

Liste des figures.....	i
Liste des tables.....	ii
Abréviations.....	iii
1 Introduction générale	1
1 Présentation et motivation de l'étude.....	2
2 Problématiques et contributions de l'étude.....	2
3 Structure de la thèse.....	3
2 L'apprentissage multi-label	6
1 Introduction.....	7
2 Définition du problème et approches d'apprentissage multi-label.....	9
2.1 Définition formelle.....	9
2.2 Approches d'apprentissage multi-label.....	10
3 Approches d'apprentissage par transformation.....	11
3.1 Méthode de pertinence binaire BR.....	11
3.2 Méthode de chaîne de classification CC.....	11
3.3 Méthodes d'étalonnage des étiquettes LP.....	12
3.4 Méthodes par paires.....	12
4 Approches d'apprentissage par adaptation.....	13
4.1 k -Voisins les plus proches.....	13
4.2 Arbres de décision.....	13
4.3 Réseaux de neurones.....	14
4.4 Machines à vastes marges.....	14
4.5 Autres modèles développés.....	15
4.5.1 Modèles génératifs et probabilistes.....	15
4.5.2 Classification associative.....	15
4.5.3 Approches bio-inspirées.....	15
5 Approches d'apprentissage ensemble.....	16
5.1 RAKEL.....	16
5.2 ECC.....	17
6 Mesures d'évaluation.....	19
6.1 Mesures basées sur les exemples.....	20
6.2 Mesures basées sur les étiquettes.....	21
6.3 Mesures basées sur le classement.....	22
7 Bases de données.....	23
8 Conclusions.....	27
3 L'apprentissage semi-supervisé et l'apprentissage actif	29
1 Introduction.....	29
2 Classification semi-supervisée.....	32
2.1 Auto-apprentissage.....	32
2.2 Modèles de reproduction.....	34
2.3 Co-apprentissage.....	36
2.4 Autres méthodes d'apprentissage semi-supervisé.....	37
3 Clustering semi-supervisé.....	38
3.1 Le k -means clustering.....	38

3.1.1	Méthodes des <i>k</i> -means.....	39
3.1.2	<i>k</i> -means clustering avec contraintes.....	41
3.2	Les cartes topologiques SOM.....	42
3.2.1	Apprentissage compétitif SOM.....	45
3.2.2	La méthode SOM_Y (SOM dédiée label).....	46
3.2.3	La carte topologique des données mixtes (SOM-mixte).....	48
4	Les classifieurs SVM.....	50
4.1	Les SVM inductifs.....	50
4.2	Les SVM transductifs.....	56
5	Apprentissage actif.....	63
5.1	Concepts et définitions.....	65
5.2	Principaux scénarios.....	67
6	Conclusion.....	68
4	Apprentissage semi-supervisé actif multi-label	70
1	Introduction.....	70
2	Travaux connexes relatifs à la classification multi-label.....	72
3	Apprentissage semi-supervisé multi-label.....	74
3.1	Classification <i>k</i> NN multi-label (ML- <i>k</i> NN).....	75
3.2	Classification SOM multi-label.....	76
3.3	Classification SVM multi-label.....	77
3.3.1	Les SVM probabilistes.....	79
3.3.2	L'espace de versions.....	80
4	Apprentissage actif multi-label.....	82
4.1	Travaux connexes sur l'apprentissage actif multi-label.....	82
4.2	Stratégies de sélection d'échantillons avec SVM multi-label.....	85
4.2.1	Estimation de la réduction de perte.....	86
4.2.2	Prédiction des étiquettes.....	87
5	Classifieur d'apprentissage actif topologique proposé.....	91
5.1	Formulations mathématiques.....	94
5.2	Stratégies de sélection d'échantillons avec <i>Act-SOM</i> multi-label.....	96
6	Conclusion.....	98
5	Cadre expérimental et résultats	100
1	Introduction.....	100
2	Caractéristiques des datasets multi-label et métriques associées.....	102
2.1	Caractéristiques des datasets multi-label.....	103
2.2	Les métriques d'évaluation adoptées.....	105
2.3	Score de complexité théorique (TCS).....	108
3	Classifieurs de base multi-label.....	110
4	Résultats de TSVM actif en multi-label.....	114
5	Résultats de l'approche proposée de SOM en apprentissage actif multi-label ...	119
6	Conclusions.....	125
	Conclusion générale	126
	Liste des travaux.....	129
	Références.....	130

Liste des figures

2.1 Les principales méthodes d'apprentissage multi-label utilisées.....	18
2.2 Les principales méthodes d'apprentissage multi-label divisées en groupes selon l'algorithme d'apprentissage machine de base qu'ils utilisent	18
2.3 Catégorisation globale des mesures d'évaluation pouvant être utilisées pour évaluer la performance prédictive des méthodes d'apprentissage multi-label	20
3.1 Principaux contextes en apprentissage semi-supervisé.....	30
3.2 Diagramme de Voronoï : chaque sous-ensemble de la partition est associé à un des référents ...	40
3.3 Architecture des cartes auto-organisatrices SOM	42
3.4 Architecture détaillée d'un réseau neuronal SOM	43
3.5 Méthode de SOM-Y (label clustering)	47
3.6 Fonction de voisinage de type gaussien, l'influence entre deux neurones dépend de leur distance et de la température	49
3.7 Transformation d'un problème non linéairement séparable en un problème linéairement séparable	52
3.8 Un SVM multi-classes en mode un contre tous permet intuitivement de faire une classification multi-labels	56
3.9 Illustration de l'apprentissage SVM transductif	57
3.10 Problématique du T-SVM	60
3.11 Quelles données pour quel type d'apprentissage	64
4.1 Exemples d'espace de versions.....	81
4.2 Organigramme de la méthode proposée	93
5.1 Répartition du dataset pour l'apprentissage semi-supervisé multi-label.....	103
5.2 Exemple de résultats ML-kNN obtenus pour la base <i>medical</i> (accuracy, Hamming loss)	112
5.3 Exemples de résultats obtenus pour la base <i>medical</i> (accuracy, Hamming loss) pour TSVM ...	114
5.4 Les performances de TSVM actif pour les trois stratégies pour différents MLDatasets.....	118
5.5 Résultats sur les six Datasets pour les trois mesures (Accuracy, F-mesure et Hamming loss)....	121
5.6 Les diagrammes critiques (CD) pour les trois évaluations des quatre approches de classification	124

Liste des tables

2.1 Résumé des méthodes selon le type d'apprentissage multi-label associé.....	17
2.2 Description des benchmark de référence.....	26
3.1 Comparaison entre SOM clustering vs k-means clustering	48
5.1 Description des benchmark de référence selon le score de leur complexité théorique TCS.....	108
5.2 Les résultats obtenus par les différentes méthodes sur les six MLDatasets	111
5.3 Résultats de classification obtenus par ML-kNN pour les trois principales métriques relevés de [Herrera et al., 2016] à titre de comparaison	112
5.4 Résultats de classification obtenus par TSVM active	118
5.5 Performance de l'apprentissage actif multi-label sur les six MLDatasets	122

Abréviations

AL	Active Learning
BinMin	Binary Minimum
BR	Binary Relevance
CC	Chain Classification
CLR	Calibrated Label Ranking
LP	Label Powerset
MLD	Multi-Label Datasets
MLC	Multi-Label Classification
MI- k NN	Multi-label- k Nearest Neighbors
MMC	Loss reduction with Minimum Confidence
MML	Mean Max Loss
SSL	Semi-Supervised Learning
S3VM	Semi-Supervised Support Vector Machines
SOM	Self Organized Map
SVM	Support Vector Machines
TCS	Theoretical Complexity Score
TSVM	Transductive Support Vector Machines

Chapitre 1

Introduction générale

Sommaire

1	Présentation et motivation de l'étude.....	2
2	Problématiques et contributions de l'étude.....	2
3	Structure de la thèse	3

Chapitre 1 : Introduction générale

Les progrès technologiques des dernières années ont propulsé la disponibilité de quantités de données, 90% des données dans le monde ont été créées au cours des deux dernières années seulement. Les bigdata ou mégadonnées, connues pour ses 4V (volume, vitesse, variété et véracité) traitent des volumes de données croissants de tous types, analysent au fil de l'eau, à mesure que les données sont collectées afin d'en tirer le maximum de valeur, et d'identifier les fraudes potentielles.

De nouvelles connaissances sont issues de l'analyse collective de ces données et tirer parti de la croissance de 80% du volume de données image, vidéo et documentaires pour améliorer la satisfaction client. Pour la véracité, comment pouvons-nous nous appuyer sur l'information si nous n'avons pas confiance en elle pour prendre nos décisions? Etablir la confiance dans les Big Data représente un défi d'autant plus important que la variété et le nombre de sources ne font qu'augmenter.

Les capacités de stockage et de communication ont effectivement augmenté de façon exponentielle, augmentant ainsi le besoin de traiter automatiquement toutes ces données. De ce fait, les techniques de l'apprentissage machine (machine learning) ont acquis une importance significative et la classification de tout type d'information numérique, y compris les textes, les photos, la vidéos, est en demande croissante. La classification multi-label est le domaine approprié où des méthodes dédiées sont étudiées et proposées pour remplir cette tâche d'étiquetage des ressources en plusieurs catégories.

L'absence de tout contrôle éditorial solide sur la publication sur le Web est probablement l'un de ses principaux avantages qui contribue à son succès mais aussi l'un de ses principaux écueils exigeant des efforts de validation coûteux afin d'assurer la qualité des informations récupérées. Mais le Web n'est pas la seule source d'informations précieuses largement utilisées pour le compte d'une meilleure qualité de vie. Dans notre vie quotidienne, les données sont produites, collectées, traitées et utilisées par un nombre toujours croissant d'équipements et de systèmes (appareils mobiles et systèmes de communication en général et de surveillance financière, bancaire et de la santé, la distribution d'énergie, etc.).

L'information est un atout essentiel pour améliorer la qualité de vie et le progrès social [Atkinson and Castro, 2008]. Toutefois, étant donné que le bénéfice potentiel de l'information augmente avec la quantité d'informations disponible, la capacité d'utiliser efficacement l'information est un enjeu clé pour l'innovation et l'amélioration de notre vie quotidienne. La disponibilité de l'information est une condition nécessaire mais pas suffisante. L'utilisation efficace de l'information dépend de l'organisation de ses besoins spécifiques. Il doit y avoir une cartographie étroite entre les concepts d'organisation et de leur conformité à des objectifs spécifiques.

La classification automatique peut être utile après la construction d'un modèle de classification qui est aligné sur les classes cibles spécifiques de la tâche de classification et servant à attribuer des catégories à de nouvelles instances remplaçant l'expert du domaine à

un coût inférieur (avec une certaine perte d'exactitude). La classification automatique a un coût qui est lié à l'effort de récupération et proportionnel au montant pré-libellé requis pour construire le modèle de classification.

1. Présentation et motivation de l'étude

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.

Une classification efficace requiert un ensemble d'instances pré-libellées couvrant le domaine cible en largeur - toutes les classes cibles sont représentées, ce qui contribue à la complétude de la classe - et en profondeur - les instances pré-étiquetées de chaque classe cible constituent un échantillon représentatif solide contribuant ainsi à la précision. Le travail présenté dans cette thèse est lié à la construction de modèles de classification, abordant l'intégralité et l'exactitude de la classe à faible coût. Le problème spécifique à résoudre est la classification efficace des collections d'objets. L'efficacité dans ce sens signifie la sensibilisation de toutes les classes cibles et l'intégralité de la classe signifie la précision et le faible coût. Notre objectif est de construire des modèles de classification efficaces qui permettent d'organiser des collections d'objets selon des besoins spécifiques à faible coût en l'absence de toute description du concept cible.

Nous supposons qu'il existe seulement un pool pré-libellé d'instances à partir duquel on peut initialiser la classification sans frais et notre objectif est de construire une classification exacte et complète des modèles des classes multi-label, indépendamment de la répartition des classes et avec le moins d'efforts possible et à faible coût, c'est-à-dire avec le moins d'instances non-libellées à libeller. Nos classifieurs sont initialisés avec un minimum d'instances pré-libellées sélectionnées à partir de l'initial pool au hasard. Pendant le processus d'apprentissage actif, nous continuerons à vouloir choisir ces instances pour construire des classifieurs précis complets par classe.

2. Problématiques et contributions de l'étude

On verra, au cours de ce travail de recherche, des procédures de classification multi-label qui est de plus en plus répandue comme technique d'exploration des données. Son objectif est de catégoriser les modèles dans plusieurs groupes non exclusifs, et elle est appliquée dans des domaines tels que la catégorisation des nouvelles (news), l'étiquetage d'image et la classification musicale, entre-autres, comme on le verra par la suite. Comparativement parlant, la classification multi-label est une tâche plus complexe que la classification multi-classe et binaire, puisque le classifieur doit apprendre la présence de différentes sorties à la fois à partir du même ensemble de variables prédictives.

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 multi-media tels que 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., 2008], 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, 2014] 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, d'une part, les contraintes d'apprentissage à la fois combinées de semi-supervisé et de multi-labelling pour des applications diverses, d'autre part, les cartes topologiques SOM sont habituellement utilisées comme réductrices de l'espace des données à l'entrée et non pas comme classifieurs à part entière, au même titre que les ML-kNN ou les SVM.

La dernière 11^{ième} Conférence internationale sur les systèmes hybrides de l'Intelligence artificielle [Charte et al., 2016] qui a regroupé les chercheurs et praticiens impliqués dans le développement et l'application de techniques symboliques et sous-symboliques visant à la construction de techniques de résolution de problèmes robustes et fiables et à apporter les réalisations les plus pertinentes dans ce domaine, souligne la prise de conscience croissante que des combinaisons pluridisciplinaires synergiques se comportent souvent mieux que des techniques individuelles telles que la neuro-computation, les systèmes flous, les algorithmes évolutifs, les systèmes multi-agents, etc. Cette série HAIS (Hybrid Artificial Intelligence Systems) de conférences, et son succès croissant est la preuve de la vitalité de ce champ d'investigation. <http://www.springer.com/series/1244>.

Le travail de cette thèse est axé sur la classification multi-label et des sujets connexes. La classification multi-label est un type particulier de classification, la classification étant l'une des tâches habituelles dans le champ data mining. L'exploration de données elle-même peut être considérée comme une étape dans un vaste processus, la découverte de nouvelles connaissances à partir des bases de données.

L'objectif des deux premiers chapitres de ce travail est d'introduire tous ces concepts, visant à définir le contexte de travail en donnant un aperçu global de l'ensemble de la découverte du processus des bases de données en multi-label. On présentera les tâches essentielles et les différents styles d'apprentissage actuellement utilisés en semi-supervisé et enfin l'aspect multi-label est introduit par rapport à d'autres types traditionnels de classement.

3. Structure de la thèse

Le reste de cette thèse est organisée en 4 chapitres :

- Le chapitre 2 (*l'apprentissage multi-label*) se réfère à l'état de l'art du champ de l'apprentissage multi-label et nous examinons ce domaine de l'apprentissage et son évolution sans cesse croissante car son impact sur les data sciences et leur modélisation est très apprécié de nos jours. Dans ce chapitre 2, nous présentons la tâche de l'apprentissage multi-label et les

méthodes pour la résoudre, tant au niveau de la classification qu'au niveau du classement et donnons un aperçu des méthodes les plus récentes d'apprentissage multi-label pour aboutir à la fin à un choix non biaisé des méthodes ainsi que des critères de mesure, pour une mener une évaluation expérimentale sur diverses bases de données.

Ces récentes années, de nombreuses approches diverses ont été développées pour résoudre des problèmes d'apprentissage multi-label. Il convient de noter qu'il n'existe pas pour le moment de méthode générique adéquate et performante et qu'un nombre significatif de mesures a également été proposé pour évaluer la performance des méthodes multi-label, qui peuvent concerner la classification et/ou le classement du problème.

Ce chapitre a pour but de synthétiser les méthodes d'apprentissage multi-label avec un aperçu sur leur prédiction, actuellement disponibles dans la littérature surtout en contexte supervisé. S'agissant de bases de données multi-label, la complexité est encore croissante de la problématique en mode semi-supervisé.

- Le chapitre 3 (***Apprentissage semi-supervisé et apprentissage actif***): Dans de nombreuses applications, les données multi-label sont non étiquetées ou l'étiquetage est coûteux ou peu pratique. Ainsi, le chapitre 3 se consacre aux efforts qui ont également été axés sur des études semi-supervisées (utilisant de grandes quantités de données non labellisées pour augmenter les données labellisées limitées) et sur l'apprentissage actif (l'algorithme demande itérativement des exemples d'étiquetage soigneusement choisis dans le but de minimiser l'effort d'étiquetage).

Formellement, l'apprentissage semi-supervisé vise la construction d'une fonction classificatrice à partir d'un ensemble fini d'échantillons d'entrée partiellement classifiés, ensuite à attribuer des étiquettes de classes aux échantillons d'entrée, ainsi il vise à généraliser le processus de telle sorte que des sous-ensembles d'échantillons d'entrée cohérents avec des étiquettes de classes identiques, appelés clusters émergent.

L'apprentissage actif est un cadre d'apprentissage adéquat pour nos objectifs, il construit l'ensemble pré-libellé itérativement comme il apprend le modèle de classification. Les instances à libeller sont sélectionnées à chaque itération du processus d'apprentissage sur la base des preuves actuelles qui sont incorporées à chaque itération. Il sélectionne les instances les plus informatives, compte tenu des données pour réduire le nombre d'instances pré-libellées requises pour construire le modèle de classification. Le critère de sélection à utiliser peut être adapté à des fins spécifiques.

Une préoccupation fondamentale dans l'apprentissage actif concerne le compromis entre l'exploration et l'exploitation qui est implémenté par chaque critère de sélection de requête. L'équilibrage entre l'exploration et l'exploitation a un impact sur la performance de la stratégie d'apprentissage et constitue de ce fait, un cadre adéquat étant donné notre objectif - construire des modèles de classification précis en étant conscient de tous les concepts cibles à faible coût, c'est-à-dire exigeant un effort réduit d'étiquetage.

Il ressort clairement que l'apprentissage semi-supervisé guidé par un apprentissage actif aboutit à de meilleurs résultats et qu'à l'avenir ces deux modes d'apprentissage seront fortement liés.

- Le chapitre 4 (***Apprentissage semi-supervisé actif multi-label***): est au cœur de cette thèse en élaborant une description formelle du problème général étudié, pour promouvoir la discussion et les développements ultérieurs. Nous décrivons plus en détail les deux aspects de l'apprentissage semi-supervisé actif et celui multi-label, en examinant les principales techniques inhérentes à ces deux aspects combinés, bien que malgré l'importance du problème, la recherche actuelle sur l'apprentissage actif pour la classification multi-label reste à l'état préliminaire. On essayera aussi, de présenter dans le cadre des stratégies d'échantillonnage d'incertitude multi-label à partir des perspectives de prédiction d'étiquette, la façon dont l'évaluation est effectuée et qui est également une caractéristique importante pour le travail connexe. Nous passerons en revue les principales contributions et décrirons les possibilités de recherches ultérieures découlant de notre travail.
- Le chapitre 5 (***Cadre expérimental et résultats***): constitue le cadre applicatif avec les protocoles associés, les expérimentations ont été menées sur six datasets qu'on a sélectionnés provenant de trois domaines: deux en biologie (*yeast*, *genbase*), deux en multimedia (*emotions*, *scene*), deux en texte (*medical* et *tmc2007*).

A partir de la littérature et des observations de résultats déjà obtenus, on peut déduire que, plus le dataset multi-label est complexe, est plus les méthodes de classification multi-label deviennent moins performantes. Comme d'habitude lors de l'évaluation d'une tâche de classification multi-label, la performance dépend non seulement de l'algorithme sélectionné, mais aussi des traits et spécificités du dataset multi-label et des mesures d'évaluation choisies qui doivent être prises en compte.

L'apprentissage semi-supervisé multi-label est mené sur trois fronts: l'apport des principaux classifieurs de base en multi-label, l'apport de l'apprentissage actif pour les S3VM (Semi-Supervised SVM) et surtout l'apport des cartes topologiques SOM et ses variantes en apprentissage actif qui constitue notre réelle contribution. Chaque fois que c'est possible, on essaiera de mener une comparaison avec les résultats de la littérature, mais à notre connaissance, l'apprentissage multi-label avec une étude expérimentale aussi variée des datasets de différents domaines impliquant des cartes topologiques dans ce sens en mode semi-supervisé actif semble être notre pierre d'achoppement.

Nous concluons ce manuscrit avec un bilan sur les résultats obtenus dans cette thèse et une discussion sur les perspectives à court et moyen terme qui s'orientent autour de deux directions possibles: l'amélioration de la nouvelle approche d'apprentissage SOM actif pour la classification multi-label et la mise à l'échelle pour la préservation de la topologie du réseau pour des datasets volumineux.

Chapitre 2

L'apprentissage multi-label

Sommaire

1	Introduction	7
2	Définition du problème et approches d'apprentissage multi-label	9
2.1	Définition formelle.....	9
2.2	Approches d'apprentissage multi-label.....	10
3	Approches d'apprentissage par transformation	11
3.1	Méthode de pertinence binaire BR.....	11
3.2	Méthode de chaîne de classification CC.....	11
3.3	Méthodes d'étalonnage des étiquettes LP.....	12
3.4	Méthodes par paires.....	12
4	Approches d'apprentissage par adaptation	13
4.1	k -voisins les plus proches.....	13
4.2	Arbres de décision.....	13
4.3	Réseaux de neurones.....	14
4.4	Machines à vastes marges.....	14
4.5	Autres modèles développés.....	15
4.5.1	Modèles génératifs et probabilistes.....	15
4.5.2	Classification associative.....	15
4.5.3	Approches bio-inspirées.....	15
5	Approches d'apprentissage ensemble	16
5.1	RAkEL.....	16
5.2	ECC.....	17
6	Mesures d'évaluation	19
6.1	Mesures basées sur les exemples.....	20
6.2	Mesures basées sur les étiquettes.....	21
6.3	Mesures basées sur le classement.....	22
7	Bases de données	23
8	Conclusions	27

Chapitre 2 : L'apprentissage multi-label

Résumé :

Dans ce chapitre, nous présentons la tâche de l'apprentissage multi-label et les méthodes pour la résoudre, tant au niveau de la classification qu'au niveau du classement. Nous commençons par une définition formelle de la tâche de l'apprentissage multi-label. Nous présentons ensuite un aperçu des méthodes les plus récentes d'apprentissage multi-label pour aboutir à la fin à un choix non biaisé des méthodes ainsi que des critères de mesure, pour une évaluation expérimentale sur diverses bases de données.

1. Introduction

L'apprentissage multi-label concerne l'apprentissage à partir d'exemples, où chaque exemple est associé à plusieurs étiquettes. Ces étiquettes multiples appartiennent à un ensemble prédéfini d'étiquettes. Selon l'objectif, nous distinguons deux types de tâches: la classification multi-label et le classement multi-label. Dans le cas de la classification multi-label, l'objectif est de construire un modèle prédictif qui fournira une liste d'étiquettes pertinentes pour un exemple donné, précédemment non vu. D'autre part, l'objectif dans la tâche de classement multi-label est de construire un modèle prédictif qui fournira, pour chaque exemple non vu, une liste de préférences (c'est-à-dire un classement) des étiquettes à partir de l'ensemble d'étiquettes possibles.

L'apprentissage multi-label a reçu une attention significative dans la communauté de la recherche au cours des dernières années: cela a permis de développer une variété de méthodes d'apprentissage multi-label. Dans ce chapitre, nous présentons les méthodes concurrentes en fonction de leur utilisation antérieure par la communauté, de la représentation de différents groupes de méthodes et de la variété des méthodes fondamentales d'apprentissage machine sous-jacentes. De même, nous énumérons d'une façon globale, les mesures d'évaluation pour pouvoir évaluer le comportement des méthodes à partir d'une variété de points de vue.

Le problème de la classification mono-étiquette porte sur l'apprentissage à partir d'exemples, où chaque exemple est associé à un seul label y_i à partir d'un ensemble fini d'étiquettes disjointes $L = \{y_1, y_2, \dots, y_Q\}$, $Q > 1$. Pour $Q > 2$, le problème d'apprentissage est appelé classification multi-classe. D'autre part, la tâche d'apprendre une cartographie à partir d'un exemple $x \in X$ (X désigne le domaine des exemples) à un ensemble d'étiquettes $Y \subseteq L$ est désigné sous le nom de classification multi-label. Contrairement à la classification multi-classe, on ne suppose pas que les variantes dans la classification multi-label soient mutuellement exclusives: plusieurs étiquettes peuvent être associées à un seul exemple, c'est-à-dire que chaque exemple peut être membre de plus d'une classe. Les étiquettes de l'ensemble Y sont appelées pertinentes, alors que les étiquettes dans l'ensemble $\bar{Y} = L \setminus Y$, désignant le complément théorique de Y dans L , ne sont pas pertinentes pour un exemple donné.

Outre le concept de la classification multi-label, l'apprentissage multi-label introduit le concept du classement multi-label [Brinker et al., 2006]. Le classement multi-label peut être

considéré comme une généralisation de la classification multi-classe, où au lieu de prédire un seul label (l'étiquette supérieure), il prédit le classement de toutes les étiquettes. En d'autres termes, le classement multi-label est compris comme l'apprentissage d'un modèle qui associe un exemple de requête x à la fois avec un classement de l'ensemble complet d'étiquettes et une bipartition de cet ensemble dans des étiquettes pertinentes et non pertinentes.

Ces récentes années, de nombreuses approches différentes ont été développées pour résoudre des problèmes d'apprentissage multi-label. Dans [Tsoumakas and Katakis, 2007], les auteurs les résument en deux grandes catégories: (a) les méthodes d'adaptation d'algorithmes et (b) les méthodes de transformation des problèmes. Les méthodes d'adaptation d'algorithmes renferment les algorithmes d'apprentissage spécifiques pour traiter directement les données multi-label. On peut citer l'apprentissage paresseux [Zhang and Zhou, 2007 ; Wieczorkowska et al., 2006; Spyromitros et al., 2008], les réseaux de neurones [Crammer and Singer, 2003; Zhang and Zhou, 2006], le boosting (la stimulation) [Schapire and Singer, 2000; De Comité et al., 2003], les règles de classification [Thabtah et al., 2004], les arbres de décision [Clare and King, 2001; Blockeel et al., 1998], etc.

Les méthodes de transformation des problèmes, d'autre part, transforment le problème d'apprentissage multi-label en un ou plusieurs problèmes de classification mono-étiquette. Les problèmes de classification mono-label sont résolus avec une approche de classification uni-label parmi celles connues et couramment utilisées et la sortie est transformée en retour en une représentation multi-label. Une approche commune à la transformation de problème consiste à utiliser des méthodes de binarisation de classe, c'est-à-dire décomposer le problème en plusieurs sous-problèmes binaires qui peuvent alors être résolus en utilisant un classifieur de base binaire. Les stratégies les plus simples dans le cadre du multi-label sont les stratégies un contre tous et un contre un, respectivement appelées méthode de pertinence binaire (binary relevance) [Tsoumakas and Katakis, 2007] et méthode par paires (pairwise) [Fürnkranz, 2002; Wu et al., 2004].

Cette catégorisation des méthodes multi-label s'étend aussi à un troisième groupe de méthodes, à savoir les méthodes d'ensemble. Ce groupe de méthodes consiste en des méthodes qui utilisent des ensembles pour faire des prédictions multi-label et leurs classifieurs de base appartiennent soit à la transformation de problème, soit à des méthodes d'adaptation d'algorithme. Les méthodes appartenant à ce groupe sont RAKEL [Tsoumakas et al., 2010], ensembles de chaînes de classification (ECC) [Read et al, 2009], les forêts aléatoires d'arbres de clustering prédictif [Kocev et al., 2009; Kocev 2011] et les forêts aléatoires d'arbres C4.5 multi-label [Clare and King, 2001].

Ces nouvelles méthodes d'apprentissage multi-label ont été proposées et expérimentalement comparées aux méthodes déjà existantes [Madjarov et al., 2012] où l'évaluation expérimentale typique a porté sur quelques ensembles de données. Les méthodes ont été comparées selon les performances en termes d'une ou de quelques métriques d'erreur et la comparaison a montré généralement qu'une méthode proposée peut surpasser les autres méthodes sur certains des ensembles de données et métriques considérés. Il convient de noter qu'il n'existe pas pour le moment de méthode générique adéquate et performante et qu'un

nombre significatif de mesures a également été proposé pour évaluer la performance des méthodes multi-label, qui peuvent concerner la classification ou la variante de classement du problème.

Le nombre de méthodes, de jeux de données et de métriques proposés pour l'apprentissage multi-label augmente constamment. À mesure que le domaine de recherche de l'apprentissage multi-label arrive à maturité, il est impératif de disposer d'un aperçu complet des méthodes et des mesures. La nécessité d'une comparaison expérimentale plus étendue et non biaisée des méthodes d'apprentissage multi-label est encore plus forte. Les premiers tutoriels reflétant les travaux sur l'apprentissage multi-label, ont été publiés à ses stades antérieurs [De Carvalho and Freitas, 2009; Tsoumakas and Katakis, 2007]. Ils ont introduit la définition de l'apprentissage multi-label et compilé les contributions et les méthodes développées jusqu'en 2009. Ensuite, dans [Tsoumakas et al., 2010], une synthèse, qui est devenue une référence pour la communauté de l'apprentissage multi-label, a été publiée. Elle inclut les principales propositions développées jusqu'en 2008 ainsi que la description des principales métriques d'évaluation.

Il convient également de citer deux autres articles récents. Tout d'abord, dans [Madjarov et al., 2012], une comparaison expérimentale de 12 méthodes d'apprentissage multi-label bien connues a été effectuée en utilisant 16 mesures d'évaluation sur 11 bases de données de référence. Son but était de mieux comprendre la performance de ces méthodes. Deuxièmement, les auteurs de [Zhang and Zhou, 2014] ont publié un article dont l'objectif principal est de décrire de manière élaborée et formelle les paramètres, les méthodes d'évaluation et huit algorithmes d'apprentissage multi-label les plus représentatifs.

En raison du nombre élevé de propositions développées ces dernières années (environ 700 nouvelles œuvres seulement de 2009 à 2012), une étude récente [Gibaja and Ventura, 2014] a été de vouloir combler cette lacune avec une mise à jour pour le sujet. Cette étude a présenté une vue d'ensemble à jour sur l'apprentissage multi-label dans le but de trier et de décrire les principales approches développées jusqu'à présent. La définition formelle du paradigme, l'analyse de son impact sur la littérature, ses principales applications, les travaux développés, les lignes directrices, ainsi que les recherches en cours ont été présentées.

Dans ce chapitre, nous reportons l'essentiel des 12 méthodes connues pour l'apprentissage multi-label, comprenant trois méthodes d'adaptation d'algorithme, cinq méthodes de transformation de problème et quatre méthodes d'ensemble, ainsi que les 16 mesures d'évaluation (six mesures basées sur des exemples, six mesures basées sur l'étiquette et de quatre mesures basées sur le classement). Cette vaste gamme des méthodes d'apprentissage multi-label, tirée de la littérature, nous permettra d'une part, de connaître les tendances actuelles et d'autre part, de pouvoir, par la suite, situer nos travaux par rapport à l'existant et ainsi de pouvoir évaluer notre contribution en ce sens sur 6 bases de données de référence provenant de différents domaines d'application : classification des images, prédiction de la fonction des gènes, classification des textes, classification vidéo (*yeast*, *emotion*, *scene*, *genbase*, *medical*, *tmc2007*).

En outre, nous évaluerons l'efficacité des méthodes choisies en procédant par la suite à trois mesures (*Hamming-loss*, *accuracy* (exactitude), *precision*, *recall* (rappel), *F-measure* et *micro-F₁*) en permettant de tirer des conclusions plus générales et d'effectuer une évaluation non biaisée de la performance prédictive des méthodes multi-label choisies.

L'apprentissage multi-label est un paradigme d'apprentissage supervisé très récent. Dans notre travail, on s'intéresse à l'apprentissage semi-supervisé où dans de nombreuses applications, les données sont non étiquetées ou l'étiquetage est coûteux ou peu pratique. Ce fait est encore plus difficile dans l'apprentissage multi-label où l'effort d'étiquetage devrait être moindre avec à l'appui un choix rigoureux des exemples à étiqueter parmi de grandes quantités de données non étiquetées.

2. Définition du problème et approches d'apprentissage multi-label

2.1. Définition formelle

Nous définissons la tâche de l'apprentissage multi-label comme suit: Etant donné:

- un espace d'exemples X qui consiste en des vecteurs de valeurs de types de données primitives (booléennes, discrètes ou continues), c'est-à-dire $\forall \mathbf{x}_i \in X = (x_{i1}, x_{i2}, \dots, x_{iD})$, où D est la taille du vecteur (ou nombre d'attributs descriptifs);
- un espace d'étiquettes $L = \{y_1, y_2, \dots, y_Q\}$, qui est un vecteur de Q variables discrètes (avec valeurs 0 ou 1);
- un ensemble d'exemples E , où chaque exemple est une paire de vecteurs de l'espace des exemples et de celui des étiquettes, respectivement, c'est-à-dire $E = \{(\mathbf{x}_i, Y_i) \mid \mathbf{x}_i \in X, Y_i \in L, 1 \leq i \leq N\}$ et N est le nombre d'exemples de E ($N = |E|$); et
- un critère de qualité q , qui récompense des modèles à haute précision prédictive et faible complexité.

Si la tâche à accomplir est, d'une part, la *classification multi-label*, l'objectif est de trouver une fonction $h : X \rightarrow 2^L$ telle que h maximise q .

La classification multi-label consiste à définir une fonction $h(\mathbf{x}_i)$ qui renvoie l'ensemble des étiquettes pertinentes. Alors pour tout $\mathbf{x} \in X = X_1 \times \dots \times X_D$, un espace d'entrée de dimension D de caractéristiques numériques ou catégorielles, on a une bipartition $(Y, \bar{Y})$ de l'ensemble des étiquettes L , où $Y = h(\mathbf{x})$ est l'ensemble des étiquettes pertinentes et $\bar{Y}$ est l'ensemble de celles non pertinentes. Ici, $\bar{Y} = L \setminus Y$ désigne le complément théorique de Y dans L . La classification multi-classe et binaire peut être considérée à la fois comme un cas particulier d'apprentissage multi-label où $h : X \rightarrow L$ et $h : X \rightarrow \{0, 1\}$. Un classifieur multi-label peut être dérivé d'un modèle de classement en utilisant une fonction à seuil. Les stratégies de seuillage peuvent être trouvées dans les références [Barutcuoglu and Schapire, 2006; Tang et al., 2009;] et [Fan et al., 2007; Ioannou et al., 2010; Montejo-Râez and Ureña López, 2006; Yang, 2001].

D'autre part, si la tâche est le *classement multi-label*, alors l'objectif est de trouver une fonction $f : X \times L \rightarrow R$, telle que f maximise q , où R est le classement des étiquettes pour un exemple donné.

Le classement des étiquettes consiste à produire la fonction f qui induit un ordre de toutes les étiquettes possibles qui expriment la pertinence des étiquettes pour une instance donnée $\mathbf{x}_i$. Ainsi, l'étiquette y_1 est considérée comme étant supérieure à y_2 si $f(\mathbf{x}_i, y_1) > f(\mathbf{x}_i, y_2)$. Pour chaque instance, $\mathbf{x}_i \in X$, une fonction de rang, $rank_f(\mathbf{x}_i, y): Y \rightarrow \{1, 2, \dots, Q\}$, peut être définie en utilisant la valeur réelle de sortie du classifieur f , telle que si $f(\mathbf{x}_i, y_1) > f(\mathbf{x}_i, y_2)$ alors $rank_f(\mathbf{x}_i, y_1) < rank_f(\mathbf{x}_i, y_2)$. Plus la valeur est basse, meilleure est la position dans le classement.

Le classement multi-label est une généralisation de la classification multi-label et du classement des étiquettes consistant à produire, en même temps, à la fois une bipartition et un classement cohérent. En d'autres termes, si Y est l'ensemble des étiquettes associées à une instance, alors, dans un classement cohérent, les étiquettes dans Y auront un rang plus élevé que les étiquettes dans $\bar{Y}$.

2.2. Approches d'apprentissage multi-label

Comme méthodes d'apprentissage multi-label, dans [Tsoumakas and Katakis, 2007], est présenté un premier aperçu des méthodes pour l'apprentissage multi-label où ces méthodes sont divisées en deux catégories: adaptation d'algorithme et méthodes de transformation des problèmes. Ainsi, trois méthodes de transformation des problèmes ont été évaluées sur une petite étude empirique (trois ensembles de données). Dans [Madjarov et al., 2012], nous retrouvons les classifieurs recommandés à l'issue d'une étude expérimentale extensive.

Dans ce chapitre, on présentera les trois catégories de méthodes pour l'apprentissage multi-label: l'adaptation de l'algorithme, la transformation du problème et les méthodes d'ensemble en donnant un aperçu global des 12 méthodes dédiées généralement pour l'apprentissage multi-label en mode supervisé. On s'intéressera tout particulièrement à six benchmark multi-label de données (trois de petite taille: *genbase*, *emotions* et *medical*; deux de taille moyenne: *yeast* et *scene*; et une de taille large: *tmc2007*) parmi les 11 datasets connus en utilisant 6 mesures d'évaluation ou critères adoptés généralement en premier pour toute évaluations objective, qu'on a testées par la suite, dans le contexte semi-supervisé.

En résumé, les approches d'apprentissage multi-label peuvent être organisées en trois grandes familles [Madjarov et al., 2012; Tsoumakas et al., 2010; Zhang and Zhou, 2014] :

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 la Table 2.1 Les approches d'apprentissage par transformation utilisent toutes un classifieur de base binaire ou multi-classes pour l'apprentissage du ou des modèles (algorithme de construction d'un arbre de décision C4.5 ou un SVM (Support Vector Machine)).

3. Approches d'apprentissage par transformation

Les méthodes de transformation du problème sont des méthodes d'apprentissage multi-label qui transforment le problème d'apprentissage multi-label en un ou plusieurs problèmes de classification ou de régression à un seul marqueur d'étiquettes. Pour les petits problèmes d'étiquettes uniques, il existe une pléthore d'algorithmes d'apprentissage automatique. Les méthodes de transformation des problèmes peuvent être regroupées en trois catégories: la pertinence binaire, l'étiquetage des étiquettes et les méthodes en paire.

3.1 Méthode de pertinence binaire *BR*

La stratégie la plus simple pour la transformation des problèmes consiste à utiliser la stratégie un contre tous, bien connue pour convertir le problème multi-label en plusieurs problèmes de classement binaires. Cette approche est connue sous le nom de méthode de pertinence binaire (*Binary Relevance : BR*) [Tsoumakas et al., 2010; Schapire and Singer, 2000]. *BR* 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.

Elle traite du problème d'apprentissage multi-label en apprenant un classifieur pour chaque étiquette, en utilisant tous les exemples étiquetés avec ce label comme exemples positifs et tous les autres exemples comme négatifs. Lors d'une prédiction, chaque classifieur binaire prédit si son étiquette est pertinente pour l'exemple donné ou non, ce qui donne à la fin, un ensemble d'étiquettes pertinentes. Dans le scénario de classement, les étiquettes sont classées selon la probabilité associée à chaque étiquette par le classifieur binaire respectif.

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 un très bon candidat 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.

Le principal problème de *BR* est l'hypothèse d'une indépendance d'étiquette qui ignore les relations entre les étiquettes [Zhou et al., 2012] et peut conduire à ne pas prédire les combinaisons d'étiquettes ou le classement des étiquettes [Tsoumakas et al., 2009]. Néanmoins, il est simple du point de vue calcul, linéairement avec le nombre d'étiquettes et peut être parallélisé [Read et al., 2011]. Certaines approches ont été développées afin de surmonter l'hypothèse d'indépendance d'étiquette de *BR* tout en maintenant une complexité raisonnable. Ils sont décrits ci-dessous.

3.2 Méthode de chaîne de classification *CC*

Une méthode étroitement liée à la méthode *BR* est la méthode de la chaîne de classification (*CC*) proposée par Read et al. [Read et al., 2009; Read et al., 2011]. *CC* 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 $[1.., j, ..Q]$ tel que chaque classifieur binaire h_j apprenant un label y_j ajoute tous les labels associés aux classifieurs qui le précèdent dans la chaîne (i.e. $y_1, \dots, y_{j-1}$) dans son espace d'attributs. Comme *BR*, pour un nouvel exemple, *CC* retourne l'ensemble des prédictions générées par l'ensemble des classifieurs.

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.

3.3. Méthodes d'étalonnage des étiquettes *LP*

Une deuxième méthode de transformation du problème est la méthode de combinaison d'étiquettes, ou méthode d'étalonnage des étiquettes (*Label Powerset : LP*) [Boutell et al., 2004; Tsoumakas et al., 2010], qui a fait l'objet de plusieurs études [Read, 2008; Tsoumakas and Katakis, 2007]. *LP* transforme le problème d'apprentissage multi-label en un seul problème d'apprentissage mono-label à plusieurs classes. *LP* considère chaque combinaison de labels présente dans l'ensemble d'apprentissage comme une classe et apprend ensuite un classifieur multi-classes h .

Pour un nouvel exemple, le classifieur retourne la classe (i.e. combinaison de labels) la plus probable. 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 $\min(2^Q, N)$ où Q nombre de labels et N nombre d'instance x_i .

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. Une autre méthode de *LP* est celle de *HOMER* [Tsoumakas et al., 2008], qui effectue d'abord une hiérarchie des étiquettes multiples, puis construit un classifieur pour les ensembles d'étiquettes dans chaque nœud de la hiérarchie.

La base de ces méthodes est de combiner des ensembles entiers d'étiquettes en étiquettes atomiques (uniques) pour former un problème d'étiquettes uniques (c'est-à-dire un problème de classification de classes uniques). Pour le problème d'étiquettes uniques, l'ensemble d'étiquettes individuelles possibles représente tous les sous-ensembles d'étiquettes distincts à partir de la représentation multi-label d'origine. De cette façon, les méthodes basées sur *LP* prennent directement en compte les corrélations d'étiquettes. Cependant, l'espace des sous-ensembles d'étiquettes possibles peut être très important.

3.4. Méthodes par paires

Une troisième approche de transformation du problème pour résoudre le problème d'apprentissage multi-label est la classification par paires (*pair-wise methods*) avec des

classifieurs binaires [Hüllermeier et al., 2008]. L'idée de base ici est d'utiliser $Q \times (Q-1)/2$ classifieurs couvrant toutes les paires d'étiquettes. Chaque classifieur est entraîné en utilisant les échantillons de la première étiquette comme exemples positifs et les échantillons de la deuxième étiquette comme exemples négatifs.

Pour combiner ces classifieurs, la méthode de classification par paire adopte naturellement l'algorithme de vote majoritaire. Compte tenu d'un exemple de test, chaque classifieur prédit (c'est-à-dire, vote pour) l'une des deux étiquettes. Après l'évaluation de tous les $Q \times (Q-1)/2$ classifieurs, les étiquettes sont classées en fonction de la somme de leur voix. Un algorithme de classement des étiquettes est ensuite utilisé pour prédire les étiquettes pertinentes pour chaque exemple.

4. Approches d'apprentissage par adaptation

Les méthodes multi-label qui adaptent, élargissent et personnalisent un algorithme d'apprentissage machine existant pour la tâche d'apprentissage multi-label sont appelées méthodes d'adaptation d'algorithme. Nous présentons ici des méthodes multi-label proposées dans la littérature qui sont basées sur les algorithmes d'apprentissage machine suivants: k -voisins les plus proches, arbres de décision et réseaux de neurones. Les méthodes élargies sont capables de gérer directement les données multi-label.

4.1. k -voisins les plus proches

Les k -voisins les plus proches en multi-label ($ML-kNN$) [Zhang and Zhou, 2007] sont une extension de l'algorithme des k -voisins les plus proches (kNN). Plusieurs variantes pour l'apprentissage multi-label ($ML-kNN$) de l'algorithme d'apprentissage paresseux populaire k -Nearest Neighbors (kNN) ont été proposées [Wieczorkowska et al., 2006; Spyromitros et al., 2008]. La récupération des k -voisins les plus proches est la même que dans l'algorithme kNN traditionnel.

$ML-kNN$ 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).

4.2. Arbres de décision

Multi-Label C4.5 ($ML-C4.5$) [Clare and King, 2001] est une adaptation de l'algorithme C4.5 bien connu pour l'apprentissage multi-label permettant plusieurs étiquettes dans les feuilles de l'arbre. Cette méthode est devenue une référence et a été principalement utilisée comme classifieur de base dans des ensembles de méthodes d'apprentissage multi-label.

Dans [Blockeel and Raedt, 1998], le concept d'arbres de regroupement prévisionnel est proposé (*Predictive Clustering Trees : PCT*). Ils peuvent également être utilisés dans le contexte de l'apprentissage multi-label, où chaque étiquette est une composante du vecteur cible.

Les *PCTs* sont des arbres de décision considérés comme une hiérarchie de grappes: le nœud supérieur correspond à un cluster contenant toutes les données, qui est divisé de façon récursive en grappes plus petites tout en descendant dans l'arborescence. Les *PCTs* sont construits en utilisant un algorithme standard d'induction de haut en bas des arbres de décision, où la variance et la fonction prototype peuvent être instanciées en fonction de la tâche à accomplir [Kocev, 2011]. Les *PCTs* ont donné de très bons résultats de classification combinés avec des forêts aléatoires [Kocev et al., 2007].

4.3. Réseaux de neurones

Les réseaux de neurones ont également été adaptés pour la classification multi-label [Crammer and Singer, 2003]. *BP-MLL* [Zhang and Zhou, 2006] est une adaptation de l'algorithme populaire de rétro-propagation pour l'apprentissage multi-label. La principale modification de l'algorithme est l'introduction d'une nouvelle fonction d'erreur qui prend en compte plusieurs étiquettes.

Crammer et Singer ont proposé *Multi-Label Multi-Class Perceptron (MMP)*, une famille d'algorithmes en ligne pour le classement de sujets sur des documents texte où un perceptron a été utilisé pour chaque étiquette mais, contrairement à la pertinence binaire (*Binary Relevance*), la performance de l'ensemble a été considérée pour la mise à jour de chaque perceptron. Ils ont montré que *MMP* surpassait *BR* sur les tâches de classification de texte [Mencía et al., 2010].

La fonction de base radiale multi-label (*ML-RBF*) [Zhang, 2009] a été inspirée de la méthode *RBF* bien connue. La première couche a été obtenue à l'aide du *k*-means clustering sur les instances de chaque classe possible, le centre de chaque cluster étant le vecteur prototype d'une fonction de base. Les poids de la seconde couche ont été appris en minimisant la fonction somme des carrés. Les résultats expérimentaux ont montré que *ML-RBF* a surpassé des méthodes comme *Rank-SVM* et *BP-MLL* dans une large gamme de métriques et datasets. Cependant, alors que le temps de prédiction était similaire à *BP-MLL*, le temps d'apprentissage était beaucoup moins élevé dans *ML-RBF*.

Il convient de citer des travaux adaptant d'autres modèles de réseaux de neurones artificiels. Le réseau neuronal probabiliste (*PNN*) [Specht, 1990] a été adapté au contexte d'apprentissage multi-label dans le domaine de la catégorisation du texte [Ciarelli et al., 2009].

4.4. Machines à vastes marges

De nombreuses approches ont utilisé des *SVM* à étiquette unique avec une approche un-contre-tous [Boutell et al., 2004 ; Gonçalves and Quaresma, 2004 ; Li et al., 2004]. De plus,

deux mécanismes ont été présentés pour améliorer la qualité de la marge des SVM dans un environnement un-contre-tous. Le premier, la méthode de suppression de bandes (*BandSVM*), fonctionnait au niveau de l'instance. Une fois qu'une SVM un-contre-tous avait été apprise, elle a enlevé des exemples négatifs semblables qui se trouvaient à une distance seuil (bande) de l'hyperplan de décision appris.

L'approche d'adaptation d'algorithme a également été utilisée, ainsi [Elisseeff and Weston, 2001] ont proposé *Rank-SVM*, une méthode de classement, basée sur les SVM, qui est devenue une référence dans l'apprentissage multi-label. La fonction de coût qu'ils utilisent est la fraction moyenne de paires d'étiquettes incorrectement ordonnées. *Rank-SVM* définit un ensemble de Q classifieurs linéaires qui sont optimisés pour minimiser une mesure qui évalue la fraction moyenne de paires d'étiquettes qui sont ordonnées inversement pour l'instance (c'est-à-dire la perte de classement empirique (*Ranking-loss*) définie parmi les mesures d'évaluation du classement).

4.5. Autres modèles développés

4.5.1 Modèles génératifs et probabilistes

Ils sont principalement liés à la catégorisation textuelle, sous l'hypothèse qu'un document est généré par un mélange de modèles de documents uniques (un par catégorie). Dans la [McCallum, 1999], un modèle génératif probabiliste est présenté, fondé sur Bayes naïfs avec maximisation des espérances (*EM*) [Dempster et al., 1977] pour connaître les poids des mélanges et les distributions de mots dans chaque composante du mélange. Le modèle proposé tente de saisir la relation entre les classes et les occurrences de mots, mais il ne tient pas compte de la corrélation au sein des classes.

4.5.2. Classification associative

La classification associative multi-étiquettes (*MMAC*) [Thabtah et al., 2004] a été l'un des premiers algorithmes d'apprentissage multi-label fondé sur la classification associative. L'antécédent d'une règle était un ensemble de paires attribut-valeur sous forme conjonctive (c'est-à-dire un élément), et la conséquence était une liste d'étiquettes de classe classées par ordre.

Le *MMAC* a d'abord appliqué l'exploration de règles d'association sur les données d'apprentissage pour découvrir et générer un ensemble initial de règles de classification dans lequel chaque règle était associée à l'étiquette de classe la plus évidente. Ensuite, le processus a été répété et de nouveaux ensembles de règles ont été générés à partir des autres instances non classifiées, jusqu'à ce qu'aucun élément fréquent ne puisse être découvert. Les jeux de règles dérivés à chaque itération ont été fusionnés pour former un classifieur multi-label.

4.5.3. Approches bio-inspirées

Plusieurs approches bio-inspirées ont été décrites et qui ont construit un classifieur d'apprentissage multi-label. Un exemple est le Multi-Label Ant-Miner (*MuLAM*) [Chan and

Freitas, 2006] qui se basait sur l'algorithme Ant-Miner de l'Optimisation des colonies de fourmis (ACO) [Parpinelli et al., 2002]. Contrairement à l'Ant-Miner original, une matrice de phéromones a été créée pour chaque classe, chaque fourmi découvrait un ensemble de règles (au plus une règle pour chaque étiquette) et plus d'une classe dans la règle conséquente a été autorisée. Les résultats expérimentaux n'ont pas montré de différences significatives avec la proposition d'une seule étiquette.

GEP-MLC [Avila et al., 2011] est une autre proposition basée sur la *GEP* (Gene Expression Programming). Chaque individu codifie une fonction discriminante qui est appliquée aux caractéristiques d'entrée du motif pour produire une valeur numérique de telle sorte qu'un seuil détermine l'appartenance à la classe. Une population de fonctions discriminantes a évolué et un algorithme a été utilisé pour garantir la diversité des solutions et déterminer les fonctions qui ont finalement constitué le classifieur. Les expériences rapportées dans leurs travaux ont montré des résultats compétitifs dans des datasets sur différents domaines.

5. Approches d'apprentissage ensemble

Les méthodes d'ensemble dont les classifieurs de base sont des apprenants multi-label sont considérées par [Madjarov et al., 2012] comme un groupe spécial de méthodes parce qu'elles sont développées en plus de la transformation de problème et des approches d'adaptation d'algorithme. Les ensembles de transformation du problème les plus connus sont le système RAKEL de [Tsoumakas and Vlahavas, 2007], les ensembles d'ensembles taillés (*Ensembles of Pruned Sets : EPS*) [Read et al., 2008] et ensembles de chaînes de classification (*Ensembles of Classifiers Chains : ECC*) [Read et al., 2009].

Les méthodes binaires sont parfois appelées méthodes d'ensemble parce qu'elles utilisent plusieurs modèles binaires. Comme aucun de ces modèles n'est multi-label, le terme ensemble est préférable dans le sens d'un ensemble de méthodes d'apprentissage multi-label [Read et al., 2011].

Outre les méthodes citées ci-dessus, la forêt aléatoire des arbres de clustering prédictif (*RF-PCT*) [Kocev et al., 2007] est un ensemble qui utilise le *PCT* (voir paragraphe 4.2) comme classifieur de base. Dans [Madjarov et al., 2012], une évaluation expérimentale intensive impliquant une grande variété d'algorithmes, de métriques et de tests statistiques a été réalisée dans laquelle *RF-PCT* a obtenu des résultats de prédiction très compétitifs.

5.1. RAKEL

Le RANdom k-labELsets (RAKEL) [Tsoumakas and Vlahavas, 2007] est une méthode d'ensemble pour la classification multi-label. Il dessine m sous-ensembles aléatoires d'étiquettes de taille k à partir de toutes les étiquettes L et forme un classifieur d'étalonnage d'étiquettes à l'aide de chaque jeu d'étiquettes. Un processus de vote simple détermine le jeu final d'étiquettes pour un exemple donné.

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 . Cette méthode s'est révélée particulièrement compétitive en termes d'efficacité.

5.2. ECC

Les ECC sont des ensembles de chaînes de classification [Read et al., 2009] qui représentent une technique de classification multi-label d'ensemble à base de classifieurs. [Read et al., 2009] entraîne m CC classifieurs $C_1, C_2, \dots, C_m$. Chaque C_k est entraîné avec un ordre aléatoire de chaîne (de L) et un sous-ensemble aléatoire de X . Par conséquent, chaque modèle C_k est susceptible d'être unique et capable de donner des prédictions multi-label différentes. Ces prédictions sont additionnées par étiquette afin que chaque étiquette reçoive un certain nombre de votes. Un seuil est utilisé pour sélectionner les étiquettes les plus populaires qui forment le jeu prédit final d'étiquettes multiples. La prévision finale est obtenue en additionnant les prédictions par étiquette, puis en appliquant un seuil pour sélectionner les étiquettes pertinentes.

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.

La figure 2.1 montre la catégorisation de ces méthodes en groupes en utilisant le schéma suivant : une méthode power-set à étiquette, deux méthodes de conversion binaire et deux méthodes de transformation par paire, deux méthodes d'adaptation d'algorithme et quatre méthodes d'ensemble. De plus, l'une des méthodes d'ensemble est basée sur le pouvoir de l'étiquette (Power-set), alors que les autres méthodes sont basées sur l'adaptation d'algorithmes. La table 2.1 résume les différentes méthodes d'apprentissage multi-label représentatives de chaque famille selon le type de classifieur de base.

Méthode	Idée principale
Binary Relevance (BR)	apprend Q classifieurs binaires
Chains Classification (CC)	apprend Q classifieurs binaires dans une chaîne
Label Powerset (LP)	apprend un seul classifieur multi-classes
Calibrated Label Ranking (CLR)	apprend $Q(Q+1)/2$ classifieurs binaires
Hierarchy Of Multi-label classifiERs (HOMER)	apprend une hiérarchie de classifieurs multi-classes
ML- k NN	combine k NN avec une inférence bayésienne
Instance-Based Learning by Logistic Regression (IBLR_ML)	combine k NN avec une régression logistique
Random k -labELsets (RAkEL)	apprend N classifieurs multi-classes
Ensembles of Binary Relevances (EBR)	apprend N classifieurs BR
Ensembles of Classifiers Chains (ECC)	apprend N classifieurs CC
Random Forest of Predictive Clustering Trees (RF-PCT)	apprend N arbres de décisions multi-label

Table 2.1 – Résumé des méthodes selon le type d'apprentissage multi-label associé.

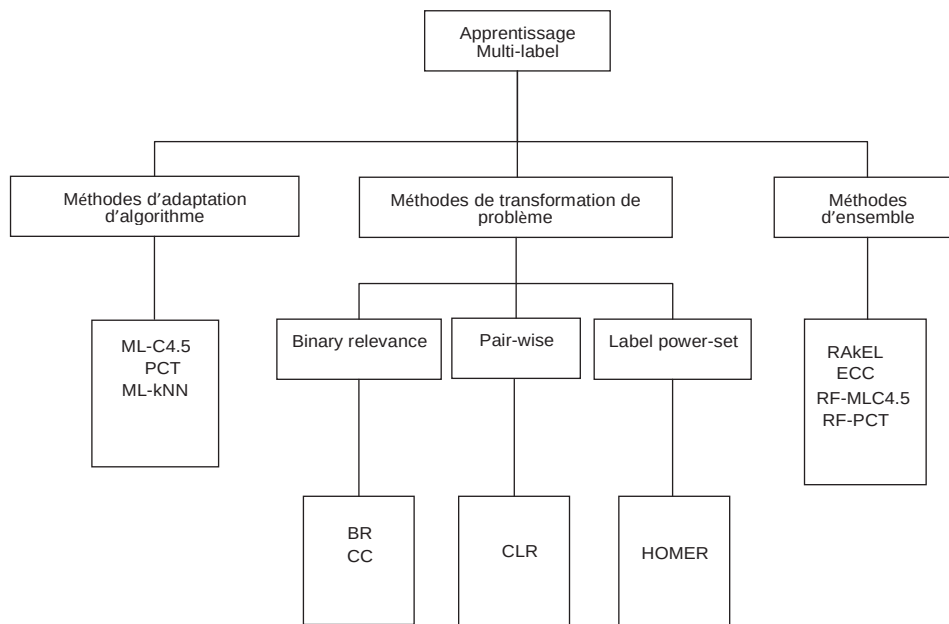


Figure 2.1 – Les principales méthodes d'apprentissage multi-label utilisées.

La figure 2.1 montre la catégorisation de ces méthodes en groupes en utilisant le schéma suivant : une méthode power-set à étiquette, deux méthodes de conversion binaire et deux méthodes de transformation par paire, deux méthodes d'adaptation d'algorithme et quatre méthodes d'ensemble. De plus, l'une des méthodes d'ensemble est basée sur le pouvoir de l'étiquette (Power-set), alors que les autres méthodes sont basées sur l'adaptation d'algorithmes.

La figure 2.2 montre la catégorisation selon les approches d'apprentissage multi-label utilisées en fonction du type d'algorithme d'apprentissage machine de base qu'elles utilisent : trois types d'algorithmes de base: SVM, arbres de décision et k -voisins les plus proches.

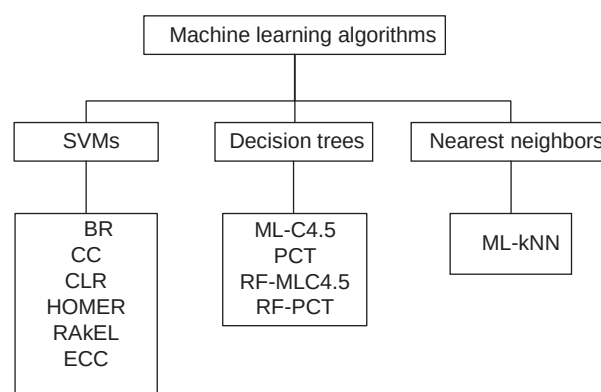


Figure 2.2 – Les principales méthodes d'apprentissage multi-label divisées en groupes selon l'algorithme d'apprentissage machine de base qu'ils utilisent.

6. Mesures d'évaluation

L'évaluation de la performance pour les systèmes d'apprentissage multi-label diffère de celle des systèmes classiques d'apprentissage simple mono-label. Dans toute expérience multi-label, il est essentiel d'inclure des mesures multiples et contrastées en raison des degrés supplémentaires de liberté que le paramètre multi-label. La figure. 2.3 représente une catégorisation générale regroupant les mesures d'évaluation pouvant être utilisées, notamment dans les expériences effectuées par [Tsoumakas et al., 2010].

Les mesures d'évaluation de la performance prédictive sont divisées en deux groupes: basées sur les bipartitions et basées selon le classement. Les mesures d'évaluation basées sur les bipartitions sont calculées sur la base de la comparaison des étiquettes pertinentes prédites avec les étiquettes pertinentes effectives de la réalité du terrain. Ce groupe de mesures d'évaluation est par ailleurs divisé en exemples et en étiquettes. Les mesures d'évaluation basées sur des exemples sont basées sur les différences moyennes entre les ensembles d'étiquettes réels et prédits sur tous les exemples de données d'évaluation.

Les mesures d'évaluation basées sur les labels, d'autre part, évaluent séparément la performance prédictive de chaque label et ensuite la performance moyenne sur tous les labels. Dans certaines expériences, on peut utiliser six mesures d'évaluation basées sur l'exemple (*Hamming loss*, *accuracy (exactitude)*, *precision*, *recall (rappel)*, *F₁score* et *subset accuracy (exactitude du sous-ensemble)*) et six mesures d'évaluation basées sur l'étiquette (*micro-precision*, *micro-recall*, *micro-F₁*, *macro-precision*, *macro-recall* et *macro-F₁*).

Notez que ces mesures d'évaluation nécessitent des prédictions indiquant qu'un label donné est présent ou non (prédictions binaires 1/0). Cependant, la plupart des modèles prédictifs prédisent une valeur numérique pour chaque étiquette et l'étiquette est prédite comme présente si cette valeur numérique dépasse un certain seuil prédéfini τ . La performance du modèle prédictif dépend donc directement de la sélection d'une valeur appropriée de τ . Pour cela, on applique généralement une méthode de calibrage de seuil en choisissant le seuil qui minimise la différence de cardinalité d'étiquette entre les données d'entraînement et les prédictions pour les données d'essai [Read et al., 2009].

Les mesures d'évaluation basées sur le classement comparent le classement prédit des étiquettes avec le classement réel de la vérité du terrain. Globalement, quatre mesures basées sur le classement peuvent être utilisées: *one-error*, *coverage (couverture)*, *ranking loss (perte de classement)* et *average precision (précision moyenne)*. Une description détaillée et globale des mesures d'évaluation est donnée en fonction de la catégorisation indiquée en figure 2.3.

Dans ce qui suit, on présente les définitions des mesures qui sont généralement utilisées pour évaluer la performance prédictive des méthodes dédiées à l'apprentissage multi-label. Dans nos expériences, on utilisera certains de ces critères pour pouvoir comparer nos expériences sur certaines bases multi-label. Dans les définitions ci-dessous, Y_i désigne l'ensemble des étiquettes vraies de l'exemple x_i et $h(x_i)$ désigne l'ensemble des étiquettes prédites pour les mêmes exemples. Toutes les définitions se réfèrent au réglage multi-label.

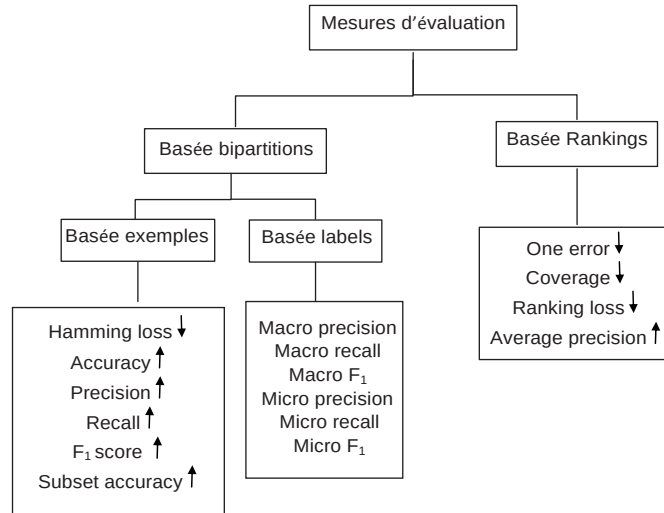


Figure 2.3 – Catégorisation globale des mesures d'évaluation pouvant être utilisées pour évaluer la performance prédictive des méthodes d'apprentissage multi-label.

↑ : métrique à maximiser, ↓ : métrique à minimiser.

6.1. Mesures basées sur les exemples

Hamming loss (la perte de Hamming) évalue combien de fois une paire d'exemple-label est mal classée, c'est-à-dire que soit l'étiquette n'appartenant pas à l'exemple est prédite ou qu'une étiquette appartenant à l'exemple n'est pas prédite. Plus la valeur de $hamming_loss(h)$ est faible, meilleure est la performance. La performance est parfaite lorsque $hamming_loss(h)=0$. Cette métrique est définie comme :

$$hamming_loss(h) = \frac{1}{N} \sum_{i=1}^N \frac{1}{Q} |h(x_i) \Delta Y_i| \quad (2.1)$$

où Δ représente la différence symétrique entre deux ensembles et correspond à l'opérateur logique booléen XOR, N est le nombre d'exemples et Q le nombre total d'étiquettes possibles des classes.

Accuracy est l'exactitude pour un exemple donné x_i , est définie par les coefficients de similarité de Jaccard entre les jeux d'étiquettes $h(x_i)$ et Y_i . L'*accuracy* est micro-moyennée sur tous les exemples :

$$accuracy = \frac{1}{N} \sum_{i=1}^N \frac{|h(x_i) \cap Y_i|}{|h(x_i) \cup Y_i|} \quad (2.2)$$

La *précision* est définie comme :

$$precision(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(x_i) \cap Y_i|}{|Y_i|} \quad (2.3)$$

Le rappel *recall* est défini comme :

$$recall(h) = \frac{1}{N} \sum_{i=1}^N \frac{|h(\mathbf{x}_i) \cap Y_i|}{|h(\mathbf{x}_i)|} \quad (2.4)$$

Le score F_1 est la moyenne harmonique entre la précision et le rappel et est défini comme :

$$F_1 = \frac{1}{N} \sum_{i=1}^N \frac{2 \times |h(\mathbf{x}_i) \cap Y_i|}{|h(\mathbf{x}_i)| + |Y_i|} \quad (2.5)$$

F_1 est une métrique basée sur un exemple et sa valeur est une moyenne sur tous les exemples de l'ensemble de données. F_1 atteint sa meilleure valeur à 1 et la plus mauvaise note à 0.

Subset Accuracy (l'exactitude du sous-ensemble) ou l'exactitude de la classification est définie comme suit:

$$subset_accuracy(h) = \frac{1}{N} \sum_{i=1}^N I(h(\mathbf{x}_i) = Y_i) \quad (2.6)$$

où $I(\text{vrai}) = 1$ et $I(\text{faux}) = 0$. Il s'agit d'une mesure d'évaluation très stricte car elle nécessite que l'ensemble prédit d'étiquettes soit une correspondance exacte du véritable ensemble d'étiquettes.

6.2. Mesures basées sur les étiquettes

La *macro-précision* (précision moyennée sur toutes les étiquettes) est définie comme :

$$macro_precision = \frac{1}{Q} \sum_{j=1}^Q \frac{tp_j}{tp_j + fp_j} \quad (2.7)$$

où tp_j et fp_j sont le nombre de vrais positifs et de faux positifs pour l'étiquette y_j considérée comme une classe binaire.

Le *macro-rappel* (rappel moyenné sur toutes les étiquettes) est défini comme :

$$macro_recall = \frac{1}{Q} \sum_{j=1}^Q \frac{tp_j}{tp_j + fn_j} \quad (2.8)$$

où tp_j est défini comme pour la macro-précision et fn_j est le nombre de faux négatifs pour l'étiquette y_j considérée comme une classe binaire.

Macro- F_1 est la moyenne harmonique entre la précision et le rappel, où la moyenne est calculée par étiquette, puis moyennée sur toutes les étiquettes. Si p_j et r_j sont la précision et le rappel pour tout $y_j \in h(\mathbf{x}_i)$ de $y_j \in Y_i$, la *macro- F_1* est :

$$macro_F_1 = \frac{1}{Q} \sum_{j=1}^Q \frac{2 \times p_j \times r_j}{p_j + r_j} \quad (2.9)$$

La *micro-precision* (moyenne de précision sur toutes les paires (exemple, étiquette)) est définie comme :

$$micro_precision = \frac{\sum_{j=1}^Q tp_j}{\sum_{j=1}^Q tp_j + \sum_{j=1}^Q fp_j} \quad (2.10)$$

où tp_j et fp_j sont définis comme pour *macro-précision*.

Le *micro-rappel* (moyenne des rappels sur toutes les paires (exemple, étiquette)) est défini comme :

$$micro_recall = \frac{\sum_{j=1}^Q tp_j}{\sum_{j=1}^Q tp_j + \sum_{j=1}^Q fn_j} \quad (2.11)$$

où tp_j et fn_j sont définis comme pour *macro-recall*.

$Micro-F_1$ est la moyenne harmonique entre micro-précision et micro-rappel. $Micro-F_1$ est défini comme :

$$micro_F_1 = \frac{2 \times micro_precision \times micro_recall}{micro_precision + micro_recall} \quad (2.12)$$

6.3. Mesures basées sur le classement

One_error évalue combien de fois l'étiquette de premier rang n'est pas dans l'ensemble des étiquettes pertinentes de l'exemple. La métrique $one_error(f)$ prend des valeurs entre 0 et 1. Plus la valeur de $one_error(f)$ est faible, meilleure est la performance. Cette mesure d'évaluation est définie comme :

$$one_error(f) = \frac{1}{N} \sum_{i=1}^N \left\| \left[\arg \max_{y \in Y} f(\mathbf{x}_i, Y) \right] \notin Y_i \right\| \quad (2.13)$$

où $y \in L = \{y_1, y_2, \dots, y_Q\}$ et $[[\omega]]$ est égale à 1 si ω se maintient et 0 autrement pour tout prédicat ω . Notez que, pour les problèmes de classification mono-label, *one_error* est identique à une erreur de classification ordinaire.

Coverage (la couverture) évalue jusqu'à quel point, en moyenne, nous devons descendre dans la liste des étiquettes classées afin de couvrir toutes les étiquettes pertinentes de l'exemple. Plus la valeur de $coverage(f)$ est faible, meilleure est la performance:

$$coverage(f) = \frac{1}{N} \sum_{i=1}^N \max_{y \in Y_i} rank_f(\mathbf{x}_i, y) - 1 \quad (2.14)$$

où $rank_f(\mathbf{x}_i, y)$ mappe (couvre) les sorties de $f(\mathbf{x}_i, y)$ pour tout $y \in L = \{y_1, y_2, \dots, y_Q\}$ de sorte que $f(\mathbf{x}_i, y_m) > f(\mathbf{x}_i, y_n)$ implique $rank_f(\mathbf{x}_i, y_m) < rank_f(\mathbf{x}_i, y_n)$. La plus petite valeur possible pour la $coverage(f)$ est l_c , c'est-à-dire la cardinalité de l'étiquette au niveau de la base de données considérée.

Ranking loss (la perte de classement) évalue la fraction moyenne de paires d'étiquettes qui sont ordonnées inversement pour l'exemple particulier, et est donné par :

$$ranking\ loss(f) = \frac{1}{N} \sum_{i=1}^N \frac{|D_i|}{|Y_i| |\bar{Y}_i|} \quad (2.15)$$

où $D_i = \{(y_m, y_n) \mid f(\mathbf{x}_i, y_m) \leq f(\mathbf{x}_i, y_n), (y_m, y_n) \in Y_i \times \bar{Y}_i\}$, tandis que $\bar{Y}_i$ désigne l'ensemble complémentaire de Y dans L . Plus petite est la valeur de $ranking_loss(f)$, meilleure est la performance, donc la performance est parfaite quand $ranking_loss(f) = 0$.

La précision moyenne (*Average precision*) est la fraction moyenne des étiquettes classées au-dessus d'une étiquette actuelle $y \in Y_i$ et qui sont effectivement en Y_i . La performance est parfaite lorsque $avg_precision(f) = 1$; plus grande est la valeur de $avg_precision(f)$, meilleure est la performance. Cette métrique est définie comme :

$$avg_precision(f) = \frac{1}{N} \sum_{i=1}^N \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|L_i|}{rank_f(\mathbf{x}_i, y)} \quad (2.16)$$

où $L_i = \{y' \mid rank_f(\mathbf{x}_i, y') \leq rank_f(\mathbf{x}_i, y), y' \in Y_i\}$ et $rank_f(\mathbf{x}_i, y)$ est défini comme dans *coverage* ci-dessus.

7. Bases de données

Ces toutes dernières années, plusieurs ateliers ont été organisés et des numéros spéciaux ont été édités sur le thème de l'apprentissage multi-label. La question de l'apprentissage à partir de données multi-label a récemment attiré l'attention de nombreux chercheurs, motivés par un nombre croissant de nouvelles applications. Parmi les principales applications de l'apprentissage multi-label :

- *La catégorisation de texte* : elle consiste à attribuer un ensemble de catégories prédéfinies aux documents. Comme un document peut appartenir simultanément à plus d'une catégorie, il peut être abordé en apprentissage multi-label. Il a été appliqué à de nombreux types de documents tels que des textes juridiques [Mencia and Fürnkranz, 2008 et 2010], documents web [Rubin et al., 2012 ; Ueda and Saito, 2002], nouvelles [Shapire and Singer, 2000], articles de recherche [Nguyen et al., 2005], textes cliniques narratifs [Spat et al., 2008], brevets [Cong, et al., 2008], ou rapports aéronautiques [Oza et al., 2009]. D'autres applications connexes sont l'indexation de documents [Lauser and Hotho, 2003], Tag suggestion d'étiquettes [Katakis et al., 2008], filtrage e-mail [Yearwood et al., 2010], codage en soins médicaux [Yan et al., 2010], catégorisation des requêtes [Tang et al., 2009] ou la

classification de nouvelles phrases en émotions à multiples catégories [Bhowmick et al., 2010].

- *Multimédia* : Les techniques de l'apprentissage multi-label ont été appliquées à de nombreux types de ressources telles que les images, vidéos et sons. Parmi les exemples d'applications: annotation automatique des images [Nasierding and Kouzani, 2010], Face [Kumar et al., 2009], annotation vidéo [Wang et al., 2010], détection des émotions dans la musique [Trohidis, 2008], extraction des métadonnées musicales [Pachet and Roy, 2009] et classification des émotions de la parole [Sobol-Shikler and Robinson, 2010].

- *La biologie* : Il convient de noter la prédiction des fonctions génétiques [Barutcuoglu and Schapire, 2006; Zhang and Zhou, 2006] et la prédiction des fonctions protéiques [Chan and Freitas, 2006] ainsi que d'autres applications telle la prédiction des structures 3D des protéines [Duwairi and Kassawneh, 2008].

- *Analyse des données chimiques* : L'apprentissage multi-label a également été appliqué pour prédire les effets indésirables des médicaments et pour identifier les médicaments qui ont deux ou plusieurs actions biologiques différentes (découverte de médicaments) [Mammadov et al., 2007], et pour détecter les contaminants dans les lubrifiants de machines en utilisant l'analyse des images spectrales [Ukwatta and Samarabandu, 2009].

- *L'exploitation des réseaux sociaux* : c'est devenu un nouveau domaine d'intérêt. L'apprentissage comportemental collectif consiste à inférer le comportement ou les préférences des individus [Tang and Liu, 2009]. La publicité en réseau social ou l'annotation automatique des nœuds d'un graphe multi-relationnel partiellement marqué sont d'autres champs d'application [Peters et al., 2010].

- *E-Learning* : L'apprentissage multi-label a également été appliqué pour classer les styles d'apprentissage sur la base des profils des apprenants et pour marquer les objets d'apprentissage [López et al., 2012].

- *Autres applications* : D'autres domaines d'application intéressants sont le marketing direct, où les acheteurs potentiels de certains produits sont identifiés et le diagnostic médical (de nombreux symptômes peuvent être associés à plus d'un syndrome) [Shao et al., 2010]. Dans la référence [Abbas et al., 2013], l'apprentissage multi-label a été appliqué pour classer les images de dermoscopie de lésions cutanées qui pourraient contenir plusieurs lésions de modèle.

Enfin, 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. Les préférences des utilisateurs se déduisent alors de leurs comportements sans explicitation préalable de leur part, en s'appuyant sur des traces

numériques d'usages recueillies automatiquement sur les différents supports d'activités. Un exemple d'application paradigmatique est le système de recommandation d'Amazon [Linden et al., 2003]. Notons, en ce sens, la thèse récente de Nair Benrekia portant sur la classification interactive multi-label pour l'aide à l'organisation personnalisée des données [Nair Benrekia, 2015].

Nous décrivons brièvement les bases de données multi-label disponibles ayant servi à une évaluation expérimentale des problèmes sélectionnés dans diverses études et évaluations de méthodes d'apprentissage multi-label [Madjarov et al., 2012] et qu'on a utilisées par la suite dans un contexte semi-supervisé. Les bibliothèques d'apprentissage multi-label suivantes sont largement utilisées par la communauté : MULAN, MeKA et CLUS.

(<http://mulan.sourceforge.net/> ; <http://meka.sourceforge.net/> ; <http://clus.sourceforge.net/> ; ou dtai.cs.kuleuven.be/clus).

Pour une vérification approfondie des performances des algorithmes, de nombreux types de jeux de données ont été testés dans nos expériences, qui sont de taille petite ou grande, de dimensions faible ou élevée, et avec un nombre de labels petit ou grande. Ces datasets comprennent 6 cas de classification multi-labels, qui couvrent quatre domaines: texte, scène, musique et biologie (http://ordinateur.njnu.edu.cn/Lab/LABIC/LABIC_Software.html).

Pour caractériser les propriétés d'un data set multi-labels, plusieurs indicateurs multi-labels utiles peuvent être utilisés. Le moyen le plus naturel de mesurer le degré de multi-labilité est la cardinalité du label, l_c , c'est-à-dire le nombre moyen de labels par exemple. En conséquence, la densité de labels normalise la cardinalité du label par le nombre de labels possibles dans l'espace des labels, l_d . La table 2.7 décrit ces 6 datasets de référence, dans lesquels l_c et l_d désignent respectivement la cardinalité et la densité du label; ces data sets couvrent différentes gammes de cas dont les caractéristiques sont diversifiées par rapport à des propriétés multi-labels différentes.

Dans le processus de sélection des problèmes, nous avons opté pour inclure des ensembles de données de référence à échelle différente et à partir de différents domaines d'application. La table 2.7 présente les statistiques de base des ensembles de données de référence. On peut noter que ces ensembles de données varient en taille: de 391 à 21 519 exemples d'apprentissage, de 191 à 7 077 exemples de tests, de 72 à 1 449 attributs, de 6 à 45 étiquettes, de 1,07 à 4,24 le nombre moyen d'étiquettes par exemple (c'est-à-dire la cardinalité de l'étiquette et la densité d'étiquette de 0,028 à 0,312 [Tsoumakas et al., 2010]).

De la littérature, ces ensembles de données sont pré-divisés en parties d'apprentissage et d'essai: ainsi, dans les expériences, nous les utilisons dans leur format d'origine. La partie d'apprentissage comprend généralement environ 2/3 de l'ensemble des données, tandis que la partie du test le 1/3 restant de l'ensemble de données.

Les ensembles de données proviennent de trois domaines: la biologie, le multimédia et la catégorisation du texte.

Du domaine biologique, nous avons les bases de données :

- ✓ *Yeast* sur la levure de [Elisseeff and Weston, 2005]. Il s'agit d'un ensemble de données largement utilisé, où les gènes sont des occurrences de la base de données et chaque gène peut être associé à 14 fonctions biologiques (étiquettes).
- ✓ *Genbase* [Diplaris et al., 2005] est un jeu de données microbiologiques où les gènes sont décrits par 1186 attributs binaires tout en étant associées avec 27 fonctions biologiques.

Les bases de données qui appartiennent au domaine multimédia sont: *emotions* et *scene* :

- ✓ *Emotions* [Trohidis et al., 2008] est un dataset dans lequel chaque instance est un morceau de musique. Chaque morceau de musique peut être étiqueté avec six émotions: triste-solitaire, fâché-agressif, étonné-surpris, détendu-calme, tranquille-serein, et heureux-content.
- ✓ *Scene* [Boutell et al., 2004] est un jeu de données de classification de scènes largement utilisé. Chaque scène peut être annotée dans les six contextes suivants: plage, coucher de soleil, champ, feuillage d'automne (chute de feuillage), montagne et urbain.

Le domaine de la catégorisation du texte est représenté par deux ensembles de données: *medical* et *tmc2007*.

- ✓ *Medical* [Read et al., 2009] est un ensemble de données utilisé dans le Medical Natural Language Processing Challenge en 2007. Chaque instance est un document qui contient un bref résumé en texte libre d'un historique des symptômes du patient. L'objectif est d'annoter chaque document avec les maladies probables de la classification internationale des maladies (CIM-9-CM) [CDC/National Center for Health Statistics, 2011]. (<http://www.computationalmedicine.org/challenge/>).
- ✓ *Tmc2007* [Srivastava and Zane-Ulman, 2005] contient des exemples de rapports sur la sécurité aérienne qui décrivent les problèmes survenus au cours de certains vols. Les étiquettes représentent les problèmes décrits dans ces rapports. Nous utilisons une version réduite de cet ensemble de données avec les 500 premiers attributs sélectionnés, comme dans [Tsoumakas and Vlahavas, 2007].

<i>Dataset</i>	<i>Domaine</i>	<i>#tr.e.</i>	<i>#t.e.</i>	<i>D</i>	<i>Q</i>	<i>l_C</i>	<i>l_D</i>
<i>Small data sets</i>							
emotions [Trohidis et al., 2008]	Musique	391	202	72	6	1.868	0.485
genbase [Diplaris et al., 2005]	Biology	463	191	1186	27	1.252	0.046
medical [Read et al., 2009]	Text	645	333	1449	45	1.245	0.028
<i>Medium data sets</i>							
yeast [Elisseeff and Weston, 2005]	Biology	1500	917	103	14	4.237	0.303
scene [Boutell et al., 2004]	Image	1211	1196	294	6	1.074	0.179
<i>Large data set</i>							
tmc2007 [Srivastava and Zane-Ulman, 2005]	Text	21 519	7077	500	22	2.158	0.098

Table 2.2 – Description des benchmark de référence en termes du domaine d'application (*domaine*), le nombre d'exemples d'entraînement (*#train.e.*), de test (*#test.e.*), le nombre de caractéristiques (features) ou d'attributs (*D*), le nombre total d'étiquettes (*Q*), la cardinalité de l'étiquette (*l_C*) et sa densité (*l_D*). Les problèmes peuvent être aussi

ordonnés par leur complexité globale calculée approximativement comme $\#train.e. \times D \times Q$.

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. Parmi ces critères, la *Cardinalité des labels* (I_c) 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.

8. Conclusions

Dans ce chapitre, nous avons présenté les méthodes pour l'apprentissage multi-label. Le sujet de l'apprentissage multi-label a récemment reçu un effort de recherche significatif. Il a également attiré beaucoup d'attention de la part des chercheurs, sous la forme de numéros spéciaux et d'ateliers lors de grandes conférences. Cela a abouti à une variété de méthodes pour aborder la tâche de l'apprentissage multi-label. Cependant, une comparaison expérimentale plus large de ces méthodes fait encore défaut dans la littérature, en dehors de celle de (Madjarov et al., 2012) où les auteurs ont procédé à une large évaluation des méthodes les plus populaires pour l'apprentissage multi-label en utilisant une large gamme de mesures d'évaluation sur une variété de jeux de données.

Les 12 méthodes multi-label qui ont été proposées dans la littérature, sont divisées en trois groupes principaux: l'adaptation d'algorithmes (trois méthodes), la transformation des problèmes (cinq méthodes) et les ensembles (quatre méthodes). Les méthodes utilisent trois types d'algorithmes d'apprentissage machine de base: SVM (sept méthodes), arbres de décision (quatre méthodes) et k -voisins les plus proches (une méthode).

En considérant les algorithmes de base d'apprentissage machine qui sous-tendent les différentes approches, les méthodes basées sur SVM sont meilleures sur des bases de données avec un grand nombre de caractéristiques (features) et un plus petit nombre d'exemples, puisqu'ils peuvent exploiter les informations de toutes les caractéristiques, alors que les arbres de décisions exploitent seulement un sous-ensemble des caractéristiques.

De la première famille d'approches (apprentissage par adaptation), nous avons sélectionné le classifieur le plus connu ML- k NN (Zhang and Zhou, 2007), de la deuxième famille d'approches (apprentissage par transformation), nous avons sélectionné le classifieur BR (Schapire and Singer, 2000). Une étude comparative préconise les classifieurs de base SVMs comme les plus efficaces (Read, 2010), d'où notre intérêt pour l'apprentissage machine en multi-label d'opter pour BR-SVM.

Dans la partie expérimentale, nous concentrerons par la suite la discussion sur les différents types d'algorithmes de base pour l'apprentissage machine en mode semi-supervisé. Tout d'abord, il ressort qu'on peut noter que la variante multi-étiquette des k -voisins les plus proches (ML- k NN) est généralement faible par rapport à la plupart des mesures d'évaluation.

Ensuite, les méthodes basées sur *SVM* fonctionnent mieux pour les bases de données plus petites, tandis que les méthodes basées sur les arbres de décision pour les bases de données plus importantes. C'est parce que le noyau gaussien peut gérer très bien le plus petit nombre d'exemples: lorsque le nombre d'exemples augmente, la performance du noyau approche de la performance d'un noyau linéaire. En outre, les méthodes basées sur *SVM* sont meilleures pour les domaines avec un plus grand nombre de features. Par exemple, dans la classification de texte, un exemple est un document typiquement représenté comme un sac-de-mots, où chaque caractéristique peut jouer un rôle crucial dans la prévision correcte. Les *SVM* exploitent les informations de toutes les caractéristiques, tandis que par exemple, les arbres de décision utilisent seulement un (petit) sous-ensemble de fonctionnalités et peuvent manquer certaines informations cruciales.

La variété des mesures d'évaluation est nécessaire pour donner une vue d'ensemble de la performance des algorithmes sous différents angles. Les 16 mesures d'évaluation différentes qui sont généralement utilisées dans le contexte de l'apprentissage multi-label, sont réparties en trois groupes: six mesures basées sur les exemples, six mesures basées sur l'étiquetage et quatre mesures basées sur le classement. Les mesures basées sur l'étiquette ont montré un comportement similaire à celui des mesures basées sur des exemples (Madjarov et al., 2012). En outre, ces mesures mettent l'accent sur les avantages des méthodes basées *SVM* sur les petits ensembles de données et les méthodes basées sur les arbres de décision sur les ensembles de données plus vastes.

Les mesures basées sur des exemples sont le plus largement utilisées pour la classification multi-label. Pour évaluer la capacité des classifieurs choisis, nous avons sélectionné 5 mesures basées sur les exemples (*Hamming-loss*, *accuracy*, *precision*, *recall* et *F-mesure*) et une mesure basée sur les étiquettes (*micro-F₁*).

En outre, les bases de données qu'on a sélectionnées proviennent de trois domaines: deux en biologie (*yeast*, *genbase*), deux en multimedia (*emotions*, *scene*), deux en texte (*medical* et *tmc2007*).

Ce chapitre avait pour but de synthétiser les méthodes d'apprentissage multi-label avec un aperçu sur leur prédiction, actuellement disponibles dans la littérature, ceci en contexte supervisé. Dans le chapitre suivant, on s'intéressera à l'aspect semi-supervisé, d'où la complexité encore croissante de la problématique, s'agissant de bases de données multi-label.

Chapitre 3

L'apprentissage semi-supervisé et l'apprentissage actif

Sommaire

1	Introduction	29
2	Classification semi-supervisée	32
2.1	Auto-apprentissage.....	32
2.2	Modèles de reproduction.....	34
2.3	Co-apprentissage.....	36
2.4	Autres méthodes d'apprentissage semi-supervisé.....	37
3	Clustering semi-supervisé	38
3.1	Le k -means clustering.....	38
3.1.1	Méthode des k -means.....	39
3.1.2	k -means clustering avec contraintes.....	41
3.2	Les cartes topologiques SOM.....	42
3.2.1	Apprentissage compétitif SOM.....	45
3.2.2	La méthode SOM_Y (SOM dédiée label).....	46
3.2.3	La carte topologique des données mixtes (SOM-mixte).....	48
4	Les classifieurs SVM	50
4.1	Les SVM inductives.....	50
4.2	Les SVM transductives (T-SVM).....	56
5	Apprentissage actif	63
5.1	Concepts et définitions.....	65
5.2	Principaux scénarios.....	67
6	Conclusion	68

Chapitre 3 : L'apprentissage semi-supervisé et l'apprentissage actif

Résumé :

Dans de nombreuses applications, les données sont non étiquetées ou l'étiquetage est coûteux ou peu pratique. Ce fait est encore plus difficile dans l'apprentissage multi-label. Ainsi, ce chapitre se consacre aux efforts qui ont également été axés sur des études semi-supervisées (utilisant de grandes quantités de données non labellisées pour augmenter les données labellisées limitées) et sur l'apprentissage actif (l'algorithme demande itérativement des exemples d'étiquetage soigneusement choisis dans le but de minimiser l'effort d'étiquetage).

Il ressort clairement que l'apprentissage semi-supervisé guidé par un apprentissage actif aboutit à de meilleurs résultats et qu'à l'avenir ces deux modes d'apprentissage seront fortement liés.

1. Introduction

L'apprentissage semi-supervisé (SSL) consiste à apprendre à la fois à partir d'échantillons pré-classifiés et non classés. Ainsi, les méthodes SSL sont situées entre des techniques classiques d'apprentissage supervisé, dans lesquelles tous les échantillons d'entrée sont pré-classifiés, et des techniques d'apprentissage non supervisées, où aucune classe n'est attribuée du tout. Pour un aperçu complet des méthodes SSL, voir [Zhu, 2007].

Formellement, SSL vise la construction d'une fonction classificatrice à partir d'un ensemble fini d'échantillons d'entrée partiellement classifiés, ensuite à attribuer des étiquettes de classe aux échantillons d'entrée. SSL vise à généraliser le processus de telle sorte que des sous-ensembles d'échantillons d'entrée cohérents avec des étiquettes de classe identiques, appelés clusters émergent.

D'autre part, il faut mentionner que les méthodes SSL sont établies selon deux objectifs : La *classification semi-supervisée* qui signifie que la fonction classificatrice résultante est limitée aux labels de classes données dans l'ensemble des instances d'entrée pré-classifiées, et le cas d'un *clustering semi-supervisé*, où la fonction du classifieur résultante est libre d'ajouter de nouveaux labels de classe ou d'en supprimer, si elle est conforme à la structure spatiale des instances d'entrée.

Le SSL se situe donc entre l'apprentissage non supervisé et celui supervisé, c'est-à-dire apprendre à partir des données étiquetées et celles non étiquetées. En fait, la plupart des stratégies d'apprentissage semi-supervisées sont basées sur l'extension de l'apprentissage choisi, soit non supervisé ou supervisé, pour inclure des informations supplémentaires typiques de l'autre paradigme d'apprentissage.

Plus précisément, le SSL englobe plusieurs contextes différents, notamment:

- *Classification semi-supervisée*. On appelle aussi classification avec des données libellées (marquées, labellisées) et non libellées (ou des données partiellement libellées), ceci est une extension du problème de classification supervisé. Les données d'apprentissage sont constituées de deux instances libellées $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ et u instances non libellées $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.

On suppose généralement qu'il y a beaucoup plus de données non libellées que les données libellées, c'est-à-dire u très grand devant l . L'objectif de la classification semi-supervisée est d'entraîner un classifieur f à partir des données libellées et non libellées, de sorte qu'il vaut mieux que le classifieur supervisé qui est entraîné sur les seules données libellées.

- *Clustering avec contraintes*. Il s'agit d'une extension de clustering non supervisé. Les données d'apprentissage se composent d'instances non libellées $\{x_i\}_{i=1}^l$, ainsi que d'une "information supervisée" sur les clusters.

Par exemple, cette information peut être ainsi appelée *must-link constraints* (contraintes de liaison obligatoire), qui stipule que deux instances proches x_i, x_j doivent être dans le même cluster (deux points proches ont probablement le même label de classe); et *cannot-link constraints* (ne peut pas lier les contraintes), que x_i, x_j éloignées, ne peuvent pas être dans le même cluster. Hypothèse de base pour la plupart des algorithmes d'apprentissage semi-supervisés : deux points qui sont connectés par un chemin traversant des régions de forte densité doivent avoir le même label. Autrement dit les frontières de décision doivent appartenir à des régions de faible densité.

On peut aussi limiter la taille des clusters. L'objectif du clustering restreint est d'obtenir un meilleur clustering que celui à partir des seules données non libellées.

Donc, l'apprentissage semi-supervisé vise à améliorer les performances en combinant les données avec labels (peu) et sans labels (beaucoup), selon (voir Fig.3.1):

- Classification semi-supervisée (discrimination) : entraîner sur des données avec labels et exploiter les données (beaucoup) sans labels pour améliorer les performances.
- Clustering semi-supervisé : clustering des données sans labels en s'aidant des données avec labels ou paires de contraintes.

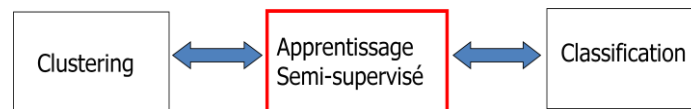


Figure 3.1– Principaux contextes en apprentissage semi-supervisé.

Il existe d'autres paramètres d'apprentissage semi-supervisés, y compris la régression avec des données étiquetées et non étiquetées, la réduction de dimensionnalité avec des instances libellées dont la représentation de caractéristiques réduites est donnée, et ainsi de suite. On se concentrera principalement sur la classification semi-supervisée.

L'étude de l'apprentissage semi-supervisé est motivée par deux facteurs: sa valeur pratique dans la construction de meilleurs algorithmes informatiques, et sa valeur théorique dans la compréhension de l'apprentissage dans les machines.

L'apprentissage semi-supervisé a une valeur pratique considérable. Dans de nombreuses tâches, il y a une pénurie de données libellées. Les labels y peuvent être difficiles à obtenir car ils nécessitent des annotateurs humains, des dispositifs spéciaux ou des expériences coûteuses et lentes.

Par exemple, en reconnaissance de la parole avec la transcription phonétique des mots en labels/phonèmes, dans l'analyse du langage naturel, avec correspondance entre une phrase et son label sous forme d'arbre d'analyse syntaxique correspondant, dans le filtrage de spam

pour étiqueter un grand nombre de courriels, dans la vidéo surveillance avec l'étiquette comme l'identité de l'objet dans la vidéo (l'étiquetage manuel des objets dans un grand nombre de trames vidéo de surveillance est fastidieux et prend du temps), dans la prédiction de la structure des protéines 3D pour une séquence d'ADN (cela peut prendre des mois de travail de laboratoire coûteux par des cristallographes experts pour identifier la structure 3D d'une seule protéine).

Alors que les données libellées (x,y) sont difficiles à obtenir dans ces domaines, les données non libellées x sont disponibles en grande quantité et faciles à collecter: les énoncés de la parole peuvent être enregistrés à partir d'émissions radio; Les phrases de texte peuvent être analysées à partir du World Wide Web, les courriels sont stockés sur le serveur de messagerie, les caméras de surveillance fonctionnent 24 heures par jour, et des séquences d'ADN de protéines sont facilement disponibles à partir de bases de données de gènes.

Cependant, les méthodes traditionnelles d'apprentissage supervisé ne peuvent pas utiliser de données non étiquetées dans les classifieurs d'apprentissage. L'apprentissage semi-supervisé est attrayant car il peut utiliser à la fois des données étiquetées et non étiquetées pour obtenir de meilleures performances que l'apprentissage supervisé. D'un point de vue différent, l'apprentissage semi-supervisé peut atteindre le même niveau de performance que l'apprentissage supervisé, mais avec moins d'exemples libellés. Cela réduit l'effort d'annotation, ce qui conduit à un coût réduit.

L'apprentissage semi-supervisé fournit également un modèle computationnel de la façon dont les humains apprennent à partir de données étiquetées et non étiquetées pour faciliter l'apprentissage conceptuel. L'étude de l'apprentissage semi-supervisé est donc une occasion de relier l'apprentissage machine et l'apprentissage humain.

Il existe en fait deux réglages d'apprentissage semi-supervisés légèrement différents, à savoir l'apprentissage semi-supervisé inductif et celui transductif. Dans la classification supervisée, l'échantillon d'apprentissage est entièrement libellé, de sorte que l'on est toujours intéressé par la performance sur les futures données de test. Dans la classification semi-supervisée, cependant, l'échantillon d'apprentissage contient des données non libellées. Par conséquent, il y a deux objectifs distincts. L'un consiste à prédire les labels sur les données de test futures. L'autre objectif est de prédire les labels sur les instances non libellées dans l'échantillon d'apprentissage. Le premier type est un apprentissage inductif semi-supervisé, et le second est un apprentissage transductif, d'où pour:

- *l'apprentissage inductif semi-supervisé.* Étant donné un échantillon d'apprentissage $\{(x_i, y_i)\}_{i=1}^l, \{x_j\}_{j=l+1}^{l+u}$, l'apprentissage inductif semi-supervisé apprend une fonction $f: X \rightarrow Y$ de sorte que f devrait être un bon prédicteur sur les données futures, au-delà de $\{x_j\}_{j=l+1}^{l+u}$. La fonction f produit non seulement des labels pour données non labellisées, mais aussi produit un classifieur.

Comme dans l'apprentissage supervisé, on peut estimer la performance sur les données futures en utilisant un échantillon séparé de test $\{(x_k, y_k)\}_{k=1}^m$, qui n'est pas disponible pendant l'apprentissage.

- *l'apprentissage transductif*. Étant donné un échantillon d'apprentissage $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, l'apprentissage transductif apprend une fonction $f: X^{l+u} \rightarrow Y^{l+u}$ de sorte que f est censée être un bon prédicteur sur les données non libellées $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$. La fonction f est définie uniquement sur l'échantillon d'apprentissage donné et n'est pas tenue de faire des prédictions à l'extérieur. C'est donc une fonction plus simple qui fournit le label uniquement pour les données disponibles non labellisées (la sortie de la méthode n'est pas un classifieur).

Il existe donc essentiellement deux façons différentes de définir les ensembles : La première façon est de considérer un ensemble de test totalement séparés (induction), cette méthode a été utilisée dans [Esuli and Sebastiani, 2009; Brinker, 2006]. L'autre façon est d'utiliser les exemples restants dans le groupe ou pool non libellé comme tests (transduction), cette approche a été utilisée dans [Singh et al., 2010; Yang et al., 2009].

2. Classification semi-supervisée

Les méthodes incluent l'auto-apprentissage (*self-training*), les modèles génératifs probabilistes, le co-apprentissage (*co-training*), les modèles basés sur les graphes, les machines à vastes marges (SVM) semi-supervisées, les méthodes de quantification durant un processus de classification d'apprentissage non supervisé, de type topologique telles les cartes SOM (*Self-Organized Map*) auto-organisatrices [Kohonen, 2001], ou de type vectoriel comme les k -means semi-supervisées, etc. Dans ce chapitre, nous allons examiner ces modèles et en particulier ceux parmi eux qui constituent des méthodes d'algorithmes d'apprentissage de base en contexte multi-label, comme on l'a vu au chapitre précédent (voir Fig.2.2).

De façon empirique, ces modèles d'apprentissage semi-supervisés produisent de meilleurs classifieurs que l'apprentissage supervisé sur certains ensembles de données. Cependant, il convient de souligner la sensibilité aux hypothèses du modèle et que choisir aveuglément une méthode d'apprentissage semi-supervisée pour une tâche spécifique n'améliorera pas nécessairement le rendement par rapport à l'apprentissage supervisé. En fait, les données non étiquetées peuvent conduire à une performance plus mauvaise avec les hypothèses de lien erroné. D'où, le point principal est que la performance d'apprentissage semi-supervisé dépend de la justesse des hypothèses faites par le modèle en question.

L'hypothèse du modèle joue un rôle important dans l'apprentissage semi-supervisé. Il compense le manque de données libellées, et peut déterminer la qualité du prédicteur. Cependant, faire de bonnes hypothèses (ou détecter de fausses hypothèses) reste une question ouverte dans l'apprentissage semi-supervisé. Par conséquent, on présentera principalement la méthodologie et les modèles distincts d'apprentissage semi-supervisé en commençant par le plus simple: auto-apprentissage.

2.1 Auto-apprentissage

L'auto-apprentissage est une technique couramment utilisée pour l'apprentissage semi-supervisé. En auto-apprentissage, un classifieur est d'abord entraîné avec la petite quantité de

données libellées. Le classifieur est ensuite utilisé pour classer les données non libellées. Généralement, les points non libellés les plus confiants, ainsi que leurs étiquettes prédites, sont ajoutés à l'ensemble d'entraînement. Le classifieur est ré-entraîné et la procédure répétée. Ainsi, le classifieur utilise ses propres prédictions pour apprendre lui-même. On peut imaginer qu'une erreur de classification peut se renforcer, autrement dit, on apprend mieux à travers nos échecs. Certains algorithmes tentent d'éviter cela en «désapprouvant» les points non libellés si la confiance de prédiction passe en dessous d'un seuil.

L'auto-apprentissage a été appliqué à plusieurs tâches de traitement du langage naturel, pour la désambiguïsation de sens des mots et la classification des dialogues comme «émotionnels» ou «non émotionnels» avec une procédure impliquant deux classificateurs [Yarowsky, 1995; Riloff et al., 2003]. L'auto-apprentissage a également été appliqué à l'analyse et à la traduction automatique. [Rosenberg et al., 2005] l'appliquent à des systèmes de détection d'objets à partir d'images et montrent que la technique semi-supervisée se compare avantageusement à celles de l'état de l'art comme détecteur de pointe.

L'auto-apprentissage est caractérisé par le fait que le processus d'apprentissage utilise ses propres prédictions pour apprendre, il peut être inductif ou transductif, selon la nature du prédicteur f . L'idée principale est d'apprendre d'abord f sur des données étiquetées. La fonction f est alors utilisée pour prédire les labels pour les données non libellées. Un sous-ensemble S des données non libellées, conjointement avec leurs labels prédits, sont ensuite sélectionnés pour augmenter les données libellées. Typiquement, S comprend les quelques exemples non libellés avec les prédictions les plus confiantes de f . La fonction f est réapprise sur l'ensemble devenu maintenant plus grand des données, et la procédure se répète. Il est également possible que S soit l'ensemble de données non libellées. Dans ce cas, L et U demeurent les échantillons entiers d'entraînement, mais les labels assignés sur des instances non libellées peuvent varier d'une itération à une itération.

L'hypothèse de l'auto-apprentissage est que ses propres prédictions, du moins celles de confiance élevée, ont tendance à être correctes. C'est probablement le cas lorsque les classes forment des clusters bien séparés. Les principaux avantages de l'auto-apprentissage sont sa simplicité et le fait qu'il s'agit d'une méthode ouverte en laissant le choix de l'apprenant pour f . Par exemple, l'apprenant peut être un simple algorithme k NN, ou un classifieur très compliqué. Ci-joint, un algorithme d'auto-apprentissage pour k NN avec $k=1$ voisin le plus proche.

Algorithme 1: Auto apprentissage k NN ($k=1$)

Entrée: données libellées $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, données non libellées $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, fonction de distance $d()$.

1. Initialement, soit $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$ et $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$
 2. Répéter jusqu'à ce que U soit vide:
 3. Selectionner $\mathbf{x} = \arg \min_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$.
 4. Assigner $f(\mathbf{x})$ au label du plus proche instance $\mathbf{x}$ de L .
 5. Retirer $\mathbf{x}$ de U ; ajouter $(\mathbf{x}, f(\mathbf{x}))$ à L .
-

Cet algorithme s'articule autour d'un classifieur à 1 voisin le plus proche. Dans chaque itération, il sélectionne l'instance non étiquetée qui est la plus proche de toute instance "libellée" (c'est-à-dire toute instance actuellement en L , dont certaines ont été étiquetées lors des itérations précédentes). L'algorithme se rapproche de la confiance par la distance aux données actuellement étiquetées. L'instance sélectionnée est alors affectée à l'étiquette de son voisin le plus proche et insérée dans L comme s'il s'agissait de données réellement étiquetées. Le processus se répète jusqu'à ce que toutes les instances aient été ajoutées à L .

2.2 Modèles de reproduction

Les modèles de reproduction ou génératifs sont peut-être la méthode d'apprentissage semi-supervisée la plus ancienne. On suppose un modèle $p(x,y) = p(y).p(x|y)$ où $p(x|y)$ est une distribution de mélange identifiable, par exemple des modèles de mélanges gaussiens. Avec une grande quantité de données non libellées, les composantes du mélange peuvent être identifiées; alors idéalement on a besoin seulement d'un exemple libellé par composante pour déterminer complètement la distribution du mélange.

[Nigam et Ghani, 2000] appliquent l'algorithme *EM* sur le mélange de multinomiales pour la tâche de classification textuelle. Ils ont montré que les classifieurs résultants ont un meilleur rendement que ceux qui ont été entraînés uniquement par L et utilisent le même algorithme sur une tâche de discrimination d'orientation de face, en étendant les modèles de mélanges génératifs en incluant un terme de correction de biais et un apprentissage discriminatif utilisant le principe de l'entropie maximale.

Il est important de construire soigneusement le modèle de mélanges pour refléter la réalité. Par exemple, dans la catégorisation de texte, un sujet peut contenir plusieurs sous-thèmes et être mieux modélisé par multinomial multiple au lieu d'un seul [Nigam et Ghani, 2000]. Même si l'hypothèse du modèle de mélange est correcte, en pratique les composantes du mélange sont identifiées par l'algorithme *Expectation-Maximisation (EM)* [Dempster et al., 1977]. *EM* est sujette aux maxima locaux. Si un maximum local est loin du maximum global, les données non libellées peuvent encore nuire à l'apprentissage. Les remèdes incluent le choix intelligent du point de départ par l'apprentissage actif [Nigam, 2001], on s'en inspire pour résoudre la classification multi-labels en mode semi-supervisé pour notre cas.

Parmi les modèles génératifs, les modèles de Markov cachés (*Hidden Markov Models*) sont utilisés pour modéliser des séquences d'instances. Chaque instance dans la séquence est générée à partir d'un état caché, où la distribution conditionnelle d'état peut être une gaussien ou une multinomiale, par exemple. En outre, les *HMMs* spécifient la probabilité de transition entre les états pour apprendre la séquence et l'apprentissage des *HMMs* implique l'estimation des paramètres des distributions conditionnelles et les probabilités de transition. Ce qui permet d'inférer les états cachés responsables de la génération des instances dans les séquences.

Un critère commun est l'estimation du maximum de vraisemblance *MLE (Maximum Likelihood Estimate)*. Compte tenues des données d'entraînement D , *MLE* est :

$$\hat{\theta} = \arg \max_{\theta} p(D|\theta) = \arg \max_{\theta} \log p(D|\theta) \quad (3.1)$$

C'est-à-dire que le *MLE* est le paramètre sous lequel la probabilité de données $p(D|\theta)$ est la plus grande. On travaille souvent avec le $\log p(D|\theta)$ au lieu de la vraisemblance directe $p(D|\theta)$, cela donne le même maximum puisque $\log()$ est monotone avec plus de facilité à manipuler.

Dans l'apprentissage semi-supervisé, D comprend des données étiquetées et non étiquetées et la probabilité dépend à la fois des deux types de données - c'est ainsi que les données non étiquetées peuvent aider les données semi-supervisées d'où l'apprentissage en modèles mixtes. Dans ce cas, il n'est plus possible de résoudre le *MLE* analytiquement. Cependant, on peut trouver un maximum local de l'estimation de paramètre en utilisant une procédure itérative bien connue, celle de l'algorithme *EM* (*Expectation-Maximisation*) d'optimisation qui tente de trouver un θ localement optimal. Pour les modèles de mélanges gaussiens (*MGM*), *Multinomial Mixture Models*, *HMM*, etc., l'algorithme *EM* a été la technique d'optimisation standard de facto pour trouver un *MLE* lorsque les données présentes ne sont pas étiquetées. L'algorithme *EM* pour les *HMMs* n'est autre que l'algorithme de Baum-Welch très connu en reconnaissance phonétique de la parole [Rabiner, 1989].

L'algorithme *EM* dans le cas général est donné en [Zhu et al., 2003] avec comme données observées $D = \{(x_1, y_1), \dots, (x_l, y_l), x_{l+1}, \dots, x_{l+u}\}$, et comme données cachées $H = \{y_{l+1}, \dots, y_{l+u}\}$. Soit θ le paramètre du modèle. L'algorithme *EM* est une procédure itérative pour trouver un θ qui maximise localement $p(D|\theta)$.

Algorithme 2 : L'algorithme général d'EM (Expectation Maximization)

Entrée: Données observées D , Données cachées H , paramètre initial $\theta^{(0)}$

1. Initialiser $t = 0$
 2. Répétez les étapes suivantes jusqu'à convergence de $p(D|\theta^{(t)})$:
 3. E-étape: calculer $q^{(t)}(H) \equiv p(H|D, \theta^{(t)})$
 4. M-étape: trouver $\theta^{(t+1)}$ qui maximise $\sum_H q^{(t)}(H) \log p(D, H|\theta^{(t+1)})$
 5. $t = t + 1$
- Sortie: $\theta^{(t)}$
-

Nous commentons quelques aspects importants de cet algorithme *EM*:

- $q^{(t)}(H)$ est la distribution d'étiquette cachée qui peut être considérée comme attribuant des étiquettes aux données non libellées selon le modèle actuel $\theta^{(t)}$.
- Il peut être prouvé que *EM* améliore le log likelihood (vraisemblance) $\log p(D|\theta)$ à chaque itération. cependant, *EM* ne converge vers un optimum local. C'est-à-dire, le θ qu'il trouve est seulement garanti pour être le meilleur dans un voisinage de l'espace de paramètres; θ peut ne pas être un optimum global (le *MLE* souhaité).
- L'optimum local auquel *EM* converge dépend du paramètre initial $\theta^{(0)}$. Un choix commun pour le $\theta^{(0)}$ est le *MLE* pris sur une petite quantité de données d'apprentissage libellées.

L'algorithme *EM* général ci-dessus doit être adapté pour des modèles génératifs spécifiques à l'application donnée ou le dataset concerné.

2.3 Co-apprentissage

Le co-apprentissage [Blum et Mitchell, 2001] suppose que (i) les caractéristiques peuvent être divisées en deux ensembles; (ii) chaque ensemble de sous-caractéristiques est suffisant pour entraîner un bon classifieur; (iii) les deux ensembles sont conditionnellement indépendants compte tenu de la classe. Initialement, deux classifieurs distincts sont entraînés avec les données libellées, respectivement sur les deux sous-ensembles de caractéristiques. Chaque classifieur classe alors les données non libellées et fait apprendre à l'autre classificateur, les quelques exemples non libellés (avec les labels prédits) dont il se sent le plus confiant. Chaque classifieur est recyclé avec les exemples d'entraînement supplémentaires donnés par l'autre classifieur, et le processus se répète.

Les deux classifieurs apprennent l'un de l'autre mutuellement, la formalisation de ce processus sous forme algorithmique est donnée comme :

Algorithme 3: L'algorithme de Co-Apprentissage

Entrée: données libellées $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, données non libellées $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, la vitesse d'apprentissage k .

Toute instance a deux vues $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}]$.

1. Initialement, soit l'échantillon d'apprentissage $L_1 = L_2 = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$.
2. Répéter jusqu'à épuisement des données non libellées :
3. Apprendre une vue-1 du classifieur $f^{(1)}$ de L_1 , et une vue-2 du classifieur $f^{(2)}$ de L_2 .
4. Classifier les données non libellées restantes avec $f^{(1)}$ et $f^{(2)}$ séparément.
5. Ajoutez les k premières prédictions $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ les plus confiantes de $f^{(1)}$ à L_2 .
6. Ajoutez les k premières prédictions $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ les plus confiantes de $f^{(2)}$ à L_1 .
7. Supprimez celles-ci des données non libellées.

Sortie : L_1 et L_2

Note $f^{(1)}$ est un classifieur de vue-1: bien qu'on lui donne le vecteur de caractéristiques complet $\mathbf{x}$, il ne prête attention qu'à la première vue $\mathbf{x}^{(1)}$ et ignore la deuxième vue $\mathbf{x}^{(2)}$. $f^{(2)}$ est l'inverse. Ils fournissent chacun leurs prévisions de données non libellées préalablement, les plus confiantes en tant que données d'apprentissage pour l'autre vue. Dans ce processus, les données non libellées seront finalement épuisées.

Le co-apprentissage est une méthode d'encapsulation. La seule exigence est que les classifieurs peuvent attribuer un score de confiance à leurs prédictions et qui est ensuite utilisé pour sélectionner les instances non libellées à transformer en données d'apprentissage supplémentaires pour l'autre vue.

En co-apprentissage, les données non libellées aident à réduire la taille de l'espace de conversion. En d'autres mots, les deux classifieurs (ou hypothèses) doivent s'accorder sur le plus grand nombre de données non libellées ainsi que celles libellées. On a besoin de

l'hypothèse que les sous-caractéristiques sont suffisamment bonnes, de sorte qu'on peut faire confiance aux labels déduits de chaque apprenant sur U et que ces classifieurs soient conditionnellement indépendants.

[Nigam et Ghani, 2000] ont effectué des expériences empiriques approfondies pour comparer le co-apprentissage avec les modèles de mélange générateurs et EM , le résultat montre un co-apprentissage qui fonctionne bien si l'hypothèse d'indépendance conditionnelle est effectivement valable. En outre, il est préférable d'étiqueter de manière probabiliste l'ensemble U , au lieu de quelques points de données les plus confiants (paradigme de $co-EM$).

2.4 Autres méthodes d'apprentissage semi-supervisé

- *Méthodes basées sur les graphes* : Les méthodes semi-supervisées basées sur un graphe définissent un graphe où les nœuds sont des exemples libellés et non libellés du dataset, et les arêtes (peuvent être pondérées) reflètent la similitude des exemples. Ces méthodes prennent habituellement le lissage des labels autour du graphe. Les méthodes basées graphes sont non paramétriques, discriminatives et généralement transductives, autrement dit, qu'elles ne peuvent pas facilement s'étendre à de nouveaux points de test en dehors de $L \cup U$, réservés pour l'ensemble d'apprentissage [Zhu, 2005].

- *Méthodes Cluster-then-Label* :

L'algorithme EM peut aussi être utilisé pour identifier les composants de mélanges à partir de données non libellées, sachant que les algorithmes de clustering non supervisés peuvent également identifier les clusters à partir de données non libellées. Cela suggère un algorithme de cluster-then-label naturel pour la classification semi-supervisée.

On peut aussi mentionner également qu'au lieu d'utiliser un modèle de mélanges génératif probabiliste, certaines approches utilisent divers algorithmes de regroupement pour regrouper des datasets entiers, puis étiqueter chaque cluster avec des données libellées. Bien qu'ils puissent bien fonctionner si les algorithmes de clustering particuliers correspondent à la véritable distribution des données, ces approches sont difficiles à analyser par la suite [Zhu, 2007]. L'algorithme de regroupement puis labellisation (cluster-then-label) est donné comme suit :

Algorithme 4: Algorithme de regroupement puis labellisation

Entrée: données libellées $(x_1, y_1), \dots, (x_l, y_l)$, données non libellées $x_{l+1}, \dots, x_{l+u}$,
un algorithme de clustering A et un algorithme d'apprentissage supervisé L .

1. Regrouper $x_1, \dots, x_{l+u}$ utilisant A .
 2. Pour chaque cluster résultant, soit S les instances libellées dans ce cluster.
 3. Si S est non-vide, apprendre un prédicteur supervisé de S : $f_S = L(S)$.
 f_S à toutes les instances non libellées de ce cluster.
 4. If S est vide, utiliser le prédicteur f appris à partir de toutes les données libellées.
- Sortie: labels des données non libellées $y_{l+1}, \dots, y_{l+u}$.
-

À l'étape 1, l'algorithme de regroupement A est non supervisé. À l'étape 2, un prédicteur supervisé est appris en utilisant les instances libellées qui appartiennent à chaque cluster et le prédicteur est utilisé pour étiqueter les instances non labellisées dans ce cluster. On peut utiliser tout algorithme de regroupement A et tout classifieur apprenant supervisé L .

Il est intéressant de noter que cluster-then-label n'implique pas nécessairement un modèle de mélanges probabiliste, on peut utiliser, par exemple, un clustering agglomératif hiérarchique pour A et un vote majoritaire simple dans chaque cluster pour L [Zhu, 2007].

3. Clustering semi-supervisé

Etant donné l'ensemble D représentant de l'espace des observations ; les observations sont supposées numériques et en dimension multiple, on suppose que chaque observation $\mathbf{z}_i$ est de dimension n et que $D \subset \mathbb{R}^n$. Chaque vecteur de D correspond à un codage particulier d'un individu issu d'une population donnée. On suppose que l'on dispose d'un ensemble de N observations représentées par le sous ensemble $A = \{\mathbf{z}_i ; i=1, \dots, N\}$ de D , représentatif de la population et qui constituera l'ensemble d'apprentissage permettant d'estimer les paramètres du modèle. On cherche par la suite, un ensemble $W = \{\mathbf{w}_c, c=1, \dots, p\}$ de p vecteurs référents de dimension n de D , qui puisse résumer toute l'information contenue dans A .

La fonction d'affectation $\varphi : D \rightarrow \{1, 2, \dots, p\}$, permet de réaliser une partition de D en p sous ensembles $P = \{P_1, \dots, P_c, \dots, P_p\}$ où $P_c = \{\mathbf{z} \in D / \varphi(\mathbf{z}) = c\}$.

La détermination des référents W et la fonction d'affectation φ réalisent d'une part, une quantification vectorielle, puisque chaque vecteur $\mathbf{z}$ dans D peut-être affecté au référent (ou prototype) $\mathbf{w}_{\varphi(\mathbf{z})}$, et d'autre part, la partition P permet de faire un regroupement (clustering) en p classes où chaque cluster P_c est décrit par un prototype représenté par le référent $\mathbf{w}_c$. La détermination de l'ensemble des paramètres W et de la fonction d'affectation φ se fait par minimisation d'une fonction de coût qui est déterminée par l'ensemble d'apprentissage A .

3.1 Le *k-means clustering*

L'approche *k-means* de clustering effectue un processus alternatif itératif d'ajustement pour former le nombre de clusters spécifiés. La méthode des *k-means* sélectionne d'abord un ensemble de n points appelés prototypes de clusters comme première estimation des moyennes des clusters. Chaque observation est affectée au prototype le plus proche pour former un ensemble de clusters temporaires. Les prototypes sont alors remplacés par les moyennes de clusters, les points sont réattribués et le processus se poursuit jusqu'à ce qu'aucun changement ne se produise dans les clusters. L'approche *k-means* est un cas particulier d'une approche générale appelée algorithme *EM*; *E* pour Expectation (signifiant dans ce cas le cluster), et *M* pour Maximisation, ce qui signifie assigner des points aux clusters les plus proches dans ce cas.

La méthode *k-means* est destinée à être utilisée avec des données volumineuses, d'environ 200 à 100 000 observations. Avec des données plus petites en nombre, les résultats peuvent être très sensibles à l'ordre des observations des données.

3.1.1 Méthode des k -means

La méthode des k -means est la méthode de quantification vectorielle la plus connue qui détermine l'ensemble des vecteurs prototypes W et la fonction d'affectation φ en minimisant la fonction de coût :

$$I(W, \varphi) = \sum_{z_i \in A} \|z_i - w_{\varphi(z_i)}\|^2 = \sum_c I_c \quad \text{avec} \quad I_c = \sum_{\substack{z_i \in A \\ \varphi(z_i)=c}} \|z_i - w_c\|^2 \quad (3.2)$$

la fonction I_c représente l'inertie des observations d'un ensemble $P_c \cap A$ par rapport à un vecteur w_c . La quantité $I(W, \varphi)$ que l'on cherche à minimiser sur l'ensemble A , représente la somme des inerties locales I_c et cette minimisation revient à former une partition des observations en sous ensembles compacts.

L'algorithme des k -means procède d'une manière itérative pour minimiser la fonction d'inertie I avec chaque itération comportant deux phases :

- 1) Phase d'affectation : pour minimiser la fonction $I(W, \varphi)$ par rapport à la fonction d'affectation φ et en supposant que les vecteurs prototypes sont fixés à la valeur courante. Cette minimisation s'obtient en affectant chaque observation z au prototype w_c à l'aide de cette fonction φ :

$$\varphi(z) = \arg \min_r \|z - w_r\|^2 \quad (3.3)$$

La nouvelle fonction d'affectation définit une nouvelle partition P de l'ensemble D , chaque observation z étant affectée au prototype le plus proche au sens de la distance euclidienne; cette partition est formée par les Voronoï des centroïdes w_c (voir Fig.3.2).

- 2) Phase de minimisation : cette deuxième phase de l'itération fait décroître à nouveau $I(W, \varphi)$ en fonction de l'ensemble des prototypes W , on suppose dans ce cas que φ est fixée à la valeur courante. Les référents w_c minimisant I sont calculés, à l'aide de la formule suivante :

$$w_c = \frac{\sum_{z_i \in P_c \cap A} z_i}{n_c} \quad (3.4)$$

où n_c représente le nombre d'éléments de P_c ou cardinalité, $n_c = |P_c|$ et w_c sont alors les centres de gravités des observations $P_c \cap A$.

En résumé, k -means clustering [MacQueen, 1967] est une méthode couramment utilisée pour partitionner automatiquement un ensemble de données en k clusters. Il procède en sélectionnant k centres de clusters initiaux puis en les affinant itérativement comme suit: chaque instance est affectée à son centre de cluster le plus proche, ensuite, chaque centre de cluster est mis à jour pour être la moyenne de ses instances constitutives et l'algorithme converge lorsqu'il n'y a plus de changement dans l'affectation des instances aux clusters. Les

clusters sont initialisés au préalable en utilisant des instances choisies au hasard dans l'ensemble de données. L'algorithme proprement établi des k -means s'énonce comme suit :

Algorithme 5 : Algorithme des k -means

1. Initialisation
 - $t = 0$, choisir les p prototypes initiaux (en général d'une manière aléatoire), et le nombre d'itérations N_{iter} .
 2. Etape itérative : à l'itération t on suppose connu l'ensemble des prototypes $w(t-1)$ et la fonction d'affectation $\varphi(t-1)$ calculés à l'itération $(t-1)$.
 - Phase d'affectation : mise à jour de la fonction d'affectation $\varphi(t)$ associée à $w(t-1)$, on affecte chaque observation z au prototype défini à partir de l'expression (3.3).
 - Phase de minimisation : la fonction d'affectation étant fixée, on calcule les nouveaux prototypes $w(t)$ en appliquant l'équation (3.4).
- Répéter l'étape itérative jusqu'à atteindre N_{iter} ou une stabilisation de la fonction de coût I .
-

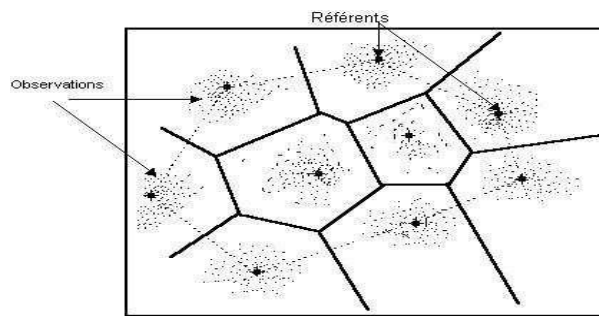


Figure 3.2 – Diagramme de Voronoï: chaque sous ensemble de la partition est associé à un des référents (prototypes).

Le clustering est traditionnellement considéré comme une méthode non supervisée d'analyse des données. Les algorithmes de clustering sont présentés avec un ensemble d'instances de données qui doivent être regroupées selon une certaine notion de similarité. L'algorithme n'a accès qu'à l'ensemble de caractéristiques décrivant chaque objet; Il ne reçoit aucune information (par exemple, des labels) quant à l'endroit où chacune des instances doit être placée dans la partition.

Les datasets peuvent être composés uniquement de caractéristiques numériques ou de caractéristiques symboliques. Pour les caractéristiques numériques, on utilise généralement la métrique de distance euclidienne et pour les caractéristiques symboliques, celle de Hamming. Concernant le choix de k , pour les datasets UCI, la valeur optimale de k est déjà connue (c'est-à-dire tous les datasets UCI: référentiel d'apprentissage machine *UC Irvine*, actuellement plus de 360 datasets disponibles pour la communauté d'apprentissage automatique <http://archive.ics.uci.edu/ml>); pour d'autres problèmes réels, il y a lieu de procéder à une recherche enveloppée pour localiser la meilleure valeur de k .

3.1.2. K-means Clustering avec contraintes

Dans le contexte des algorithmes de partitionnement, les contraintes au niveau des instances sont un moyen utile d'exprimer une connaissance a priori des instances qui doivent ou non être regroupées [Wagstaff et Rogers, 2001]. Par conséquent, deux types de contraintes par paires sont considérés:

- Les contraintes de liaison obligatoire (*must-link constraints*) spécifient que deux instances doivent être dans le même cluster.
- Les contraintes ne peuvent lier (*cannot-link constraints*) spécifient que deux instances ne doivent pas être placées dans le même cluster.

Les contraintes de liaison obligatoire définissent une relation binaire transitive sur les instances. Par conséquent, en utilisant un ensemble de contraintes (des deux types), on aura un ensemble complet des contraintes dérivées qui est ensuite présenté à l'algorithme de clusters. En général, les contraintes peuvent être dérivées de données partiellement étiquetées ou de connaissances de base sur le domaine ou l'ensemble de données.

L'algorithme *k*-means modifié (*COP-Kmeans*) prend en compte un ensemble de données (D), un ensemble de contraintes de liaison obligatoire ($Con_{=}$) et un ensemble de contraintes de non-liaison ($Con_{\neq}$) et renvoie une partition des instances de D qui satisfait toutes les contraintes spécifiées.

Algorithme 6 : Cop-kmeans (dataset D , must-link constraints $Con_{=}\subseteq D\times D$, cannot-link constraints $Con_{\neq}\subseteq D\times D$)

1. Soit $C_1 \dots C_k$ les centres initiaux des clusters.
2. Pour tout point d_i dans D , l'assigner au plus proche cluster C_j de sorte que les contraintes de violation ($(d_i, C_j, Con_{=}, Con_{\neq})$ soient fausses. Si aucun cluster existe, échec (retourner $\{\}$).
3. Pour tout cluster C_i , mettre à jour son centre en faisant la moyenne de tous les points d_j qui lui ont été assignés.
4. Répéter (2) et (3) jusqu'à convergence
5. Retourner $\{C_1 \dots C_k\}$.

Contraintes de violation (point de données d , cluster C , must-link contraintes $Con_{=}\subseteq D\times D$, cannot-link contraintes $Con_{\neq}\subseteq D\times D$)

1. Pour tout $(d, d_{=}) \in Con_{=}$: If $d_{=} \notin C$, retourner vrai.
 2. Pour tout $(d, d_{\neq}) \in Con_{\neq}$: If $d_{\neq} \in C$, retourner vrai.
 3. sinon, retourner faux.
-

La modification majeure est qu'aucune des contraintes spécifiées n'est violée lors de la mise à jour des assignations de clusters, où chaque point x_i est assigné à son plus proche cluster C_j . Cela se produira à moins qu'une contrainte ne soit violée. S'il existe un autre point $d_{=}$ qui doit être affecté au même cluster que d , mais qui est déjà dans un autre cluster, ou il

il y a un autre point $d \neq$ qui ne peut être groupé avec d mais qui est déjà dans C , d ne peut pas être placé en C . On continue sur la liste triée de clusters jusqu'à ce qu'on trouve un qui peut légalement accueillir d . Les contraintes ne sont jamais rompues; si un cluster légal ne peut pas être trouvé pour d , la partition vide ($\{\}$) est renvoyée. Une démo interactive de cet algorithme de k -means clustering avec contraintes peut être trouvée à <http://www.cs.cornell.edu/home/wkiri/cop-kmeans/>.

Les ensembles de données utilisés pour l'évaluation comprennent une «réponse correcte» ou un label pour chaque instance de données. Les labels sont utilisés dans une étape de post-traitement pour évaluer le rendement. La précision globale augmente constamment avec l'incorporation des contraintes [Wagstaff et Rogers, 2001].

3.2 Les cartes topologiques SOM

Les cartes SOM (*Self Organizing Map*) est un algorithme de réseau neuronal artificiel populaire basé sur l'apprentissage non supervisé. Le SOM est capable de projeter des données de grande dimension dans une dimension inférieure qui peut être utile pour analyser les motifs dans l'espace d'entrée [Lawrence et al., 1999 ; Haykin, 2003]. La Fig.3.3 présente l'architecture d'une carte SOM. Leur rôle principal est donc de faire une projection non linéaire des données de haute dimension sur un espace de faible dimension. Les cartes auto-organisatrices sont largement utilisées dans la classification de données.

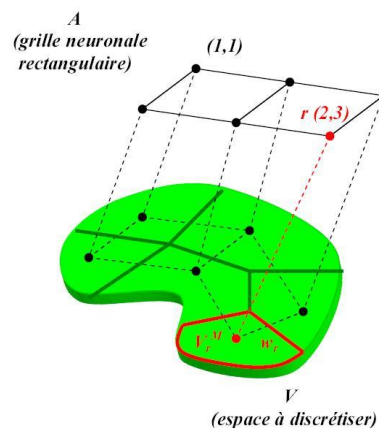


Figure 3.3 – Architecture des cartes auto-organisatrices SOM.

L'espace d'entrée V est divisé en plusieurs zones. w_r représente un vecteur référent associé à une petite zone de l'espace V_r^M et $r(2,3)$ représente son neurone associé dans la grille A . Chaque zone peut être adressée facilement par les index des neurones dans la grille.

Le réseau se compose d'une entrée et d'une couche de sortie. Le nombre de neurones dans la couche d'entrée est égal à la dimension du vecteur d'entrée de caractéristiques. La couche de sortie consiste en une grille 2-D régulière de neurones appelée carte. Les neurones de la carte peuvent être disposés sur un réseau rectangulaire ou hexagonal, chaque neurone de la carte étant relié à tous les neurones de la couche d'entrée en utilisant un vecteur de poids. La fig.3.4 présente l'architecture d'une carte réseau neuronal SOM.

Chaque neurone a un vecteur référent qui le représente dans l'espace d'entrée V (de Voronoï). Un vecteur d'entrée v est présenté, il sélectionne le neurone vainqueur s , le plus proche dans l'espace d'entrée. Le vecteur référent du vainqueur w_s est rapproché de v . Les

vecteurs référents des autres neurones sont aussi déplacés vers v , mais avec une amplitude moins importante.

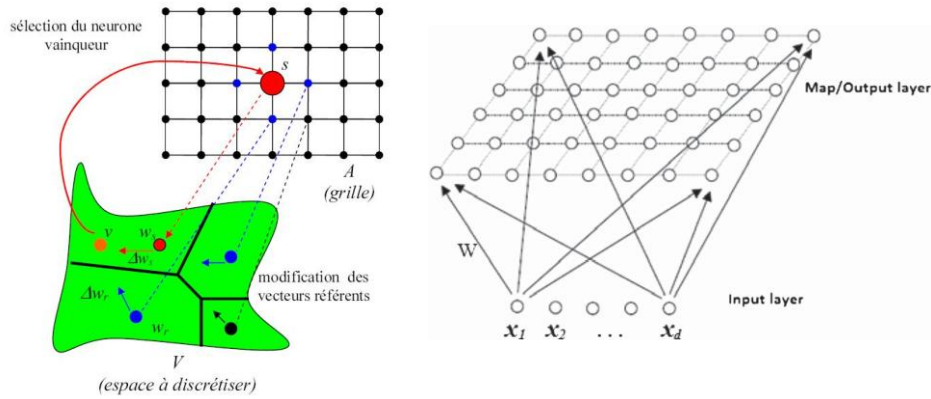


Figure. 3.4 – Architecture détaillée d'un réseau neuronal SOM.

- *Formulation mathématique* : La cartographie de l'espace d'entrée est réalisée en adaptant les vecteurs référents w_r . L'adaptation est faite par un algorithme d'apprentissage dont la puissance réside dans la compétition entre neurones et dans l'importance donnée à la notion de voisinage. Une séquence aléatoire de vecteurs d'entrée est présentée pendant l'apprentissage. Avec chaque vecteur v , un nouveau cycle d'adaptation est démarré. Pour chaque vecteur v dans la séquence, on détermine le neurone vainqueur, c'est-à-dire le neurone dont le vecteur référent approche v le mieux possible par la fonction d'affectation φ :

$$s = \varphi_w(v) = \arg \min_{r \in A} \|v - w_r\| \quad (3.5)$$

Le neurone vainqueur s et ses voisins (définis par une fonction d'appartenance au voisinage) déplacent leurs vecteurs référents vers le vecteur d'entrée.

$$w_r(t+1) = w_r(t) + \Delta w_r(t) \quad (3.6)$$

avec
$$\Delta w_r(t) = \eta(t) \times h_{sr}(t) \times (v - w_r(t)) \quad (3.7)$$

où t désigne le temps, $\eta(t)$ ($0 < \eta(t) < 1$) est le pas d'apprentissage ou paramètre de la vitesse d'apprentissage qui représente l'amplitude du déplacement global de la carte et la fonction $h_{sr}(t)$ définit l'appartenance au voisinage, c'est un multiplicateur scalaire donnant le noyau voisin autour du neurone gagnant s du cluster r .

La forme de la carte ou topologie définit les voisinages des neurones et donc les liaisons entre neurones. La fonction de voisinage $h_{sr}(t)$ décrit comment les neurones dans la proximité du vainqueur s sont entraînés dans le mouvement de correction. On utilise en général une fonction continue de forme gaussienne:

$$h_{sr}(t) = \exp(-(\|\vec{r} - \vec{s}\|/2\sigma(t))) \quad (3.8)$$

où σ est le coefficient de voisinage. Son rôle est de déterminer un rayon de voisinage autour du neurone vainqueur. La décroissance de la taille du voisinage s'obtient par

diminution de l'écart type σ : σ grand, beaucoup de neurones se rapprochent du noyau s , σ petit, l'adaptation reste très localisée.

La fonction de voisinage h force les neurones qui se trouvent dans le voisinage de s à rapprocher leurs vecteurs référents du vecteur d'entrée v . Moins un neurone est proche du vainqueur dans la grille, moins son déplacement est important.

La correction de vecteurs référents est pondérée par les distances dans la grille. Cela fait apparaître, dans l'espace d'entrée, les relations d'ordre dans la grille. Pendant l'apprentissage la carte décrite par les vecteurs référents du réseau évolue d'un état aléatoire vers un état de stabilité dans lequel elle décrit la topologie de l'espace d'entrée tout en respectant les relations d'ordre dans la grille.

- *Propriétés :*

- 1) Similitude des densités dans l'espace d'entrée : la carte reflète la distribution des points dans l'espace d'entrée. Les zones dans lesquelles les vecteurs d'entraînement v sont tirés avec une grande probabilité d'occurrence sont cartographiées avec une meilleure résolution que les zones dans lesquelles les vecteurs d'entraînement v sont tirés avec une petite probabilité d'occurrence.
- 2) Préservation des relations topologiques : des neurones voisins dans la grille occupent des positions voisines dans l'espace d'entrée (préservation des voisinages de la grille) ; et des points proches dans l'espace d'entrée se projettent sur des neurones voisins dans la grille (préservation de la topologie de l'espace d'entrée). Les neurones ont tendance à discrétiser l'espace de façon ordonnée.

- *Avantages et inconvénients des cartes auto adaptatives :* Les ancêtres des cartes auto-organisatrices, les algorithmes comme k -means, réalisent la discrétisation de l'espace d'entrée en ne modifiant à chaque cycle d'adaptation qu'un seul vecteur référent. Leur processus d'apprentissage est donc très long.

L'algorithme de Kohonen dont le concept a été développé en 1984 [Kohonen, 2001] profite des relations de voisinage dans la grille pour réaliser une discrétisation dans un temps très court. On suppose que l'espace n'est pas constitué de zones isolées, mais de sous-ensembles compacts. Donc en déplaçant un vecteur référent vers une zone, on peut se dire qu'il y a probablement d'autres zones dans la même direction qui doivent être représentées par des vecteurs référents. Cela justifie le fait de déplacer les neurones proches du vainqueur dans la grille dans cette même direction, avec une amplitude de déplacement moins importante. L'algorithme présente des opérations simples; il est donc très léger en termes de coût de calcul.

Le voisinage dans les cartes auto adaptatives est malheureusement fixe, et une liaison entre neurones ne peut être cassée même pour mieux représenter des données discontinues. Les Growing Neural Gas sont la solution à ce problème: des neurones et les liaisons entre neurones peuvent y être supprimés ou rajoutés quand le besoin s'en fait sentir [Zhang and Zhou, 2006].

3.2.1 Apprentissage compétitif SOM

L'algorithme SOM est basé sur le concept d'apprentissage compétitif. Lorsqu'un échantillon d'apprentissage (qui ne comprend pas d'informations sur l'étiquette de classe) est alimenté en entrée au réseau, une distance métrique est calculée pour tous les vecteurs de poids. Le neurone de la carte avec le vecteur de poids le plus semblable au motif d'entrée est appelé la meilleure unité d'adaptation (*BMU: Best Matching Unit*). Les poids du *BMU* et de ses neurones voisins sont ensuite ajustés en fonction du modèle d'entrée. L'ampleur du changement diminue avec le temps et avec la distance du *BMU*. Après l'apprentissage, la carte produite par l'algorithme SOM préserve la propriété topologique des motifs d'entrée, c'est-à-dire que les vecteurs de poids qui sont voisins dans l'espace d'entrée sont mappés sur des neurones voisins de la carte.

Plus précisément, soit $X \in R^d$ et $W \in R^d$ l'ensemble des vecteurs d'entrée et de poids dans un espace d-dimensionnel, respectivement. Soit à chaque neurone k de la carte, un vecteur poids est associé $w_k \in W$. Les valeurs initiales pour les vecteurs de poids peuvent être définies au hasard. L'algorithme SOM suit un apprentissage séquentiel (incrémental) ou en différé (en batch) pour mettre à jour les vecteurs poids [Kohonen, 2001].

- *Apprentissage séquentiel*: Dans cette méthode, les vecteurs de poids sont mis à jour immédiatement après la présentation d'un modèle d'entrée. Ainsi, l'algorithme SOM séquentiel peut être formalisé comme suit:

Algorithme 7 : Apprentissage SOM séquentiel

- 1: Sélectionner aléatoirement un échantillon x_i de l'ensemble d'apprentissage X .
- 2: Trouver le *BMU* correspondant, noté comme c_i , comme suit:

$$c_i = \arg \min_{K=1,2,\dots,|W|} \{\|x_i - w_k(t)\|^2\} \quad (3.9)$$

- 3: Mettre à jour le vecteur de poids du neurone k ($k = 1, 2, \dots, |W|$) comme suit:

$$w_k(t+1) = w_k(t) + \eta(t)h_{c_i k}(t)[x_i - w_k(t)] \quad (3.10)$$

où t désigne le temps, $0 < \eta(t) < 1$ est le paramètre de la vitesse d'apprentissage, et le multiplicateur scalaire $h_{c_i k}(t)$ est le noyau voisin autour du neurone gagnant c_i .

- 4: Répéter à partir de l'étape 1 pour toutes les instances d'apprentissage x_i ($i = 1, 2, \dots, |X|$), complétant une époque.
 - 5: Diminuer la valeur du noyau du voisinage et le taux d'apprentissage.
 - 6: Répéter à partir de l'étape 1 jusqu'à ce que le critère de convergence soit atteint.
-

- *Apprentissage par lot (batch)*: Dans cette méthode, la totalité de l'ensemble d'apprentissage est analysée immédiatement, et seulement après cette analyse la carte est mise à jour compte tenu des effets de tous les échantillons. Les nouveaux vecteurs de poids sont calculés comme suit:

$$w_k(t+1) = \frac{\sum_{i=1}^{|x|} h_{c_{ik}}(t)x_i}{\sum_{i=1}^{|x|} h_{c_{ik}}(t)} \quad (3.11)$$

L'algorithme SOM en batch peut être formalisé comme suit:

Algorithme 8 : Apprentissage SOM en batch

- 1: Trouver le *BMU* pour un vecteur d'entrée x_i en utilisant (3.9).
 - 2: Accumuler le numérateur et le dénominateur de (3.11) pour tous les neurones.
 - 3: Répéter à partir de l'étape 1 pour toutes les instances d'apprentissage x_i ($i = 1, 2, \dots, |X|$), complétant une époque.
 - 4: Mise à jour des poids des neurones avec (3.11).
 - 5: Diminuer la valeur du noyau du voisinage.
 - 6: Répéter à partir de l'étape 1 jusqu'à ce que le critère de convergence soit atteint.
-

Le facteur du taux d'apprentissage n'est pas présent dans la méthode par batch qui est plus rapide que l'apprentissage séquentiel. Le processus d'apprentissage des deux algorithmes SOM peut être décomposé en deux phases: une phase d'ordonnancement, suivie d'une phase de convergence. Dans la phase d'ordonnancement, le paramètre du taux d'apprentissage et la fonction de voisinage commencent par de grandes valeurs et puis se rétrécissent lentement avec le temps (époque). Dans la phase de convergence, le paramètre du taux d'apprentissage conserve de petites valeurs et la fonction de voisinage ne contient que les voisins les plus proches du *BMU*.

En général, l'apprentissage semi-supervisé vise à découvrir des structures spatiales dans des espaces d'entrée de grande dimension, lorsque l'on dispose d'informations insuffisantes sur les clusters [Lebbah et al., 2000]. Ceci est basé sur l'hypothèse que les entités proches devraient appartenir à la même classe, tandis que les entités lointaines pourraient appartenir à des classes différentes.

3.2.2 La méthode SOM_Y (SOM dédiée label)

Elle utilise des cartes auto-organisées (SOM) comme algorithme de clustering sous-jacent. Après le processus de clustering, les poids des neurones sont étiquetés et ensuite utilisés comme un ensemble de données d'échantillons à apprendre afin d'effectuer la prédiction finale. Les résultats ont montré que l'échantillon obtenu à partir d'un clustering SOM semi-supervisé peut présenter une performance relativement similaire à l'ensemble de données et peut être utilisé comme un dataset libellé convenable pour une classification ultérieure de clusters non libellés.

Ce travail tente de tirer profit de l'apprentissage de clusters des cartes SOM pour pouvoir regrouper les données multidimensionnelles sur un espace dimensionnel plus petit. Ce type de modèle de réseau neuronal artificiel peut fournir un espace cartographique bidimensionnel à partir d'un espace d'entrée multidimensionnel ainsi que la cartographie de préservation de la topologie. Chaque neurone à l'intérieur du SOM produit des zones de clusters basées sur l'approximation de la densité de données connectées aux vecteurs d'entrée sur des itérations

dans l'apprentissage non supervisé. Selon [Abbas, 2008], *SOM* montre plus de précision dans le classement de la plupart des objets à leurs clusters que d'autres algorithmes de clustering tels que les *k*-means et la maximisation-attente (*EM*).

En outre, les connaissances acquises lors d'une approche semi-supervisée ont permis une meilleure classification des modules non étiquetés par rapport au regroupement non supervisé et à la classification supervisée, d'où l'intérêt grandissant accordé par de nombreux chercheurs à l'apprentissage semi-supervisé [Abaei et al., 2015].

Abaei et al. ont proposé un modèle de prédiction en utilisant la carte auto-organisée (*SOM*) à seuil pour construire un meilleur modèle de prédiction lorsque les données sont non libellées. Dans leur étude, la carte *SOM* est utilisée au lieu de *k*-means en raison de plusieurs inconvénients tels que le problème des minima locaux et la sensibilité aux données bruitées (voir table 3.1). L'idée générale de la méthode d'étiquetage *SOM* non supervisé proposé en [Azcarraga et al., 2005] consiste en 5 étapes principales (voir Fig.3.5) :

- 1) Répartir en clusters tous les nœuds qui ont des vecteurs de poids de référence similaires.
- 2) Pour chaque cluster de nœuds, éliminez les nœuds à valeurs aberrantes (outliers) qui sont très différents des centroïdes de leur cluster.
- 3) Pour chaque cluster de nœuds (moins les outliers), classer l'ensemble des instances d'apprentissage (non libellées) en tant que *in-patterns* ou *out-patterns*, selon que leur nœud le plus proche dans la carte est dans le cluster ou en dehors.
- 4) Sur la base de l'ensemble des *in-patterns* et *out-patterns* d'un cluster donné, identifier les dimensions saillantes.
- 5) Sur la base des dimensions saillantes, attribuer un libellé descriptif à chaque cluster de nœuds qui est significatif dans le contexte du domaine d'application.

À l'étape 1 : regroupement des nœuds en *k* clusters de vecteurs poids de référence similaires.

À l'étape 2, suppression de certains nœuds de leurs clusters s'ils diffèrent trop des centroïdes. Pour ce faire, on calcule le centroïde de chaque cluster :

$$C_{kj} = \frac{\sum_{n_i \in C_k} w_{ij}}{|C_k|}, \quad j = 1, \dots, D \quad (3.12)$$

Où *D* est la dimensionnalité des données et w_{ij} est la $j^{\text{ième}}$ composante du vecteur de poids de référence du nœud n_i , l'un des nœuds en C_k . $|C_k|$ retourne la cardinalité du cluster.

La distance d_i de chaque nœud n_i dans C_k à son centroïde est:

$$d_i = \sqrt{\sum_{j=1}^D (w_{ij} - C_{kj})^2} \quad (3.13)$$

Un nœud est retenu si sa distance est suffisamment proche (moins que la déviation standard) de la distance moyenne de tous les nœuds dans le même cluster d'où le choix approprié des dimensions saillantes.

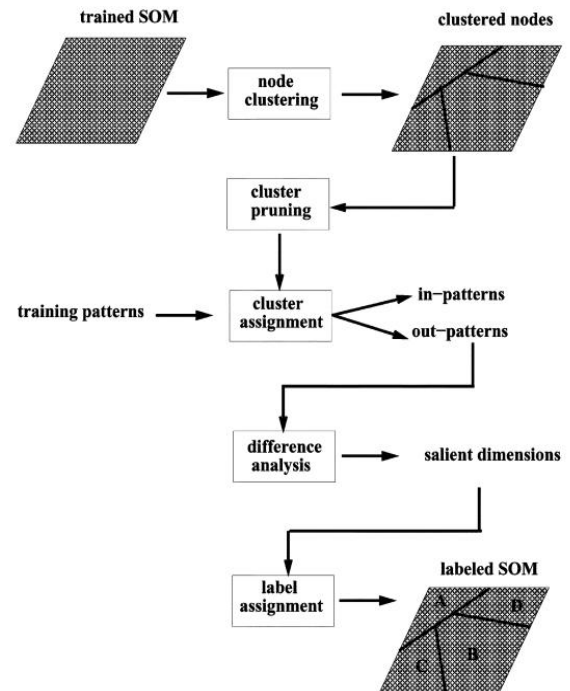


Figure 3.5—Méthode de SOM_Y (label clustering)

Par la suite, au chapitre 5 relatif aux résultats expérimentaux, en plus des méthodes *kNN*, *SOM*, *SOM-mixte*, on utilisera *SOM_Y* qui permet d'associer à chaque prototype des labels, en prenant les exemples les plus proches au prototype et procéder par la suite à un vote majoritaire, nous avons appliqué aussi différents masque ou lissage sur la partie des labels pour voir leurs comportements.

Comparaison entre les deux clustering <i>SOM</i> vs <i>k-means</i> .	
<i>SOM</i>	<i>k-means</i>
Atteindre l'optimum global dans l'espace de recherche	Algorithme gourmand donc plus enclin à l'optima local
Plus approprié pour les data sets bruités	Sensible aux données bruitées
Structure complexe, mais les résultats sont visuels et faciles à analyser	Facile et simple à mettre en œuvre et à analyser
Sensible à la topologie de la couche de sortie	Sensible à la division d'initiative
La taille de la carte <i>SOM</i> peut être déterminée.	Des paramètres tels que la valeur de <i>k</i> doivent être identifiés dès le début

Table 3.1 – Comparaison entre *SOM* clustering vs *k-means* clustering [Abaei et al. 2015].

3.2.3 La carte topologique des données mixtes (*SOM-Mixte*)

En [Rogovschi et al., 2011], les auteurs N. Rogovschi, M. Lebbah et N. Grozavu de LIPN-Paris 13 et de LIPADE-Paris Descartes, ont proposé une approche de classification topologique et de pondération des variables mixtes (qualitatives et quantitatives codées en binaire) durant un processus d'apprentissage non supervisé. Comme dans le cas des cartes auto-organisatrices classiques, le graphe est une grille régulière C à une ou deux dimensions, permettant de définir une distance $\delta(i, j)$, entre deux cellules i et j de C . Le modèle proposé dans [Rogovschi et al., 2011], *lw-MTM* (*Local Weighted Mixed Topological Map*) est basé sur le formalisme de quantification des cartes topologiques.

Soit A l'ensemble de données $\mathbf{x}$ d'apprentissage où chaque observation $\mathbf{x} = (x^1, x^2, \dots, x^k, \dots, x^d)$ est composée de deux parties: une partie continue $\mathbf{x}^{r[.]} = (x^{r[1]}, x^{r[2]}, \dots, x^{r[n]})$ ($\mathbf{x}^{r[.]} \in \mathbb{R}^n$) et une autre partie catégorielle $\mathbf{x}^{c[.]} = (x^{c[1]}, x^{c[2]}, \dots, x^{c[l]}, \dots, x^{c[k]})$ où la $l^{\text{ième}}$ composante $x^{c[l]}$ a M_l modalités. Chaque variable catégorielle peut être codée avec une variable binaire, comme un vecteur $\mathbf{x}^{b[.]} = (x^{b[1]}, \dots, x^{b[M]})$ où $x^{b[l]} \in \{0, 1\}$.

- La minimisation de la fonction de coût :

$$I(\phi, W, Y) = \sum_{x \in A} \sum_{j \in C} K(\delta(\phi(x), j)) y_j^r \|x - w_j\|^2 \quad (3.14)$$

où r est un paramètre d'ajustement qui est nécessaire pour l'estimation de l'ensemble des pondérations Y . On note par ϕ la fonction d'affectation qui attribut chaque observation $\mathbf{x}$ à

une cellule de C . K^T est une fonction de voisinage qui dépend du paramètre T (appelé température) : $K(\delta) = K^T(\delta/T)$, où K est une fonction noyau particulière qui est positive et symétrique ($\lim_{|x| \rightarrow \infty} K(x) = 0$). Ainsi K définit pour chaque cellule j une région de voisinage sur la carte C . Le paramètre T permet de contrôler la taille du voisinage d'influence d'une cellule sur la carte, celle-ci décroît avec le paramètre T (voir Fig.3.6).

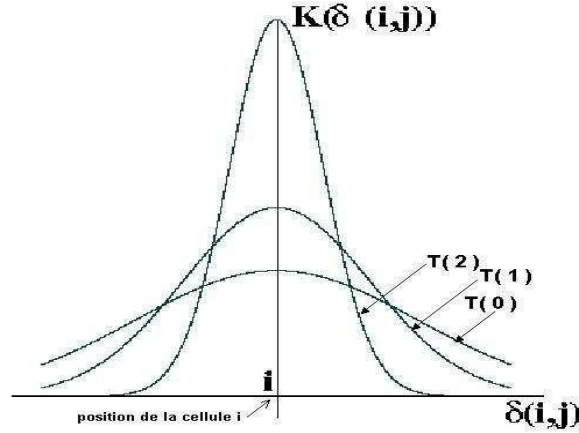


Figure 3.6 – Fonction de voisinage de type gaussien, l'influence entre deux neurones dépend de la distance entre ces neurones ainsi que de la température.

Le vecteur $y_j = (y^{r[.]}, y^{c[.]})$ est le vecteur de pondération, où $y^{r[.]}$ est la pondération de la partie continue des observations et $y^{c[.]}$ est le vecteur de pondération des variables catégorielles. On notera par Y l'ensemble des vecteurs de pondération. Pour le codage binaire la distance euclidienne est remplacée par la distance de Hamming H , ainsi on peut réécrire la fonction de coût de la manière suivante :

$$I(\phi, W, Y) = \sum_{x \in A} \sum_{j \in C} K(\delta(\phi(x), j)) (y_i^{r[.]})^r D_{euc}(x^{r[.]}, w_j^{r[.]}) + \sum_{x \in A} \sum_{j \in C} K(\delta(\phi(x), j)) (y_j^{c[.]})^r H(x^{b[.]}, w_j^{b[.]}) \quad (3.15)$$

La *lw-MTM* (*Local Weighted Mixed Topological Map*) surpasse la carte issue de l'algorithme *MTM* (*Mixed Topological Map*) classique ne tenant pas compte de la pondération [Rogovschi et al., 2008].

Il s'avère que lorsque la taille du dataset est faible, la taille des clusters représentatifs qui est nécessaire pour l'apprentissage des classifieurs n'est pas assez grande, d'où l'intérêt d'utiliser, dans le cadre de l'approche semi-supervisée, les grands datasets (big data) pour récupérer un nombre relativement assez grand de clusters représentatifs des caractéristiques générales des données d'entrée.

4. Les classifieurs SVM

Les SVM ont été introduites par Vapnik en 1995 et reposent sur l'existence d'un classifieur linéaire dans un espace approprié. Généralement le problème de classification fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyaux (kernels) qui permettent une séparation optimale des données [Vapnik, 2001].

Support Vector Machine ou Séparateur à Vaste Marge (SVM) est une approche supervisée de l'apprentissage machine pour résoudre les problèmes de reconnaissance des modèles de classification binaires. SVM adopte la marge maximale pour trouver la surface de décision qui sépare les exemples d'entraînement libellés positifs et négatifs d'une classe [Burges, 1998]. Pour une instance donnée, la SVM régulier donne des distances entre les instances de données de 0 à 1. La valeur 0 indique que cette instance de données se situe sur l'hyperplan et la valeur 1 signifie qu'elle est un vecteur de support.

Les SVM sont le résultat de l'application du principe de la Minimisation du Risque Structurel (SRM) proposé par Vapnik à la théorie bien étudiée des hyperplans (séparateurs linéaires). L'intérêt suscité par les SVM est essentiellement dû à deux facteurs. Le premier facteur est le fait que les SVM obtiennent des performances qui sont généralement parmi les meilleures en apprentissage. Ces résultats proviennent du bon niveau de généralisation induit par la SVM.

À la vue de ces performances, de nombreuses variantes du SVM ont été développées pour traiter différents types de problèmes. Le second facteur, expliquant le succès des SVM, est l'utilisation des noyaux pour transformer le SVM en un algorithme non-linéaire pouvant être appliqué sur des données variées. Les noyaux et les SVM ont alors permis d'utiliser l'apprentissage numérique dans des problèmes traitant des données complexes telles que les données textuelles.

4.1 Les SVM inductives

Bien que la SVM ait été utilisée avec succès dans divers domaines, cependant, dans de nombreuses applications réelles, il n'y a pas assez de données libellées pour apprendre un bon modèle de classification. Comparer à la SVM standard qui utilise uniquement des données d'entraînement libellées, de nombreuses SVM semi-supervisées (ou S3VM) utilisent des données non libellées avec des données libellées pour les classifieurs d'apprentissage avec une généralisation et une performance améliorées.

La S3VM a été bien accueillie pour deux raisons. Tout d'abord, l'étiquetage d'un grand nombre d'exemples prend beaucoup de temps et exige beaucoup de main-d'œuvre. Cette tâche doit également être effectuée par des experts qualifiés et coûte donc chère. Deuxièmement, certaines études montrent que l'utilisation de données non libellées pour l'apprentissage peut améliorer la précision des classifieurs, la machine à vecteur de support transducteur (TSVM) [Joachim, 1998] est une méthode efficace pour améliorer la précision de généralisation de SVM en trouvant un étiquetage pour les données non libellées, avec une limite linéaire

commune à la marge maximale à la fois sur les données initiales libellées et les données libellées non libellées initialement [Tian et al., 2012].

La caractéristique remarquable de *TSVM*, étant transductive, vise de tels problèmes d'apprentissage qui ne s'intéressent vraiment qu'aux datasets spécifiques des données de test et/ou d'apprentissage [Joachim, 1998], alors que le travail traditionnel sur l'apprentissage inductif estime un classifieur sur la base de certaines données d'apprentissage qui se généralisent à tous les exemples d'entrée.

L'idée principale de l'apprentissage transductif est de construire des modèles pour la meilleure performance de prédiction sur un dataset de test particulier au lieu de développer des modèles généralisés à être appliqués à tout dataset de test. En d'autres termes, en incluant explicitement le dataset de travail constitué d'exemples non libellés dans la formulation des problèmes, une meilleure généralisation peut être obtenue sur des problèmes avec des instances de données libellées insuffisantes [Joachim, 1998].

L'un des problèmes les plus courants est que la machine peut libeller incorrectement le dataset d'apprentissage, ce qui entraînera une erreur de classification. La solution à ce problème réside dans l'apprentissage actif, d'où l'intérêt de ce dernier volet dans un contexte semi-supervisé (voir paragraphe 5 suivant).

La *SVM* est un classifieur binaire supervisé dont l'objectif est de diviser l'espace de l'entité d'entrée d -dimensionnelle en deux sous-espaces (un pour chaque classe) en utilisant un hyperplan de séparation. Une caractéristique importante de *SVM* est liée à la possibilité de projeter les données d'origine dans un espace de caractéristique de dimension supérieure via une fonction de noyau $K(.,.)$ qui modélise implicitement le problème de classification dans un espace dimensionnel supérieur où la séparation linéaire entre classes peut être approchée [Vapnik, 2001; Burges, 1998].

Supposons qu'un ensemble d'apprentissage se compose de N échantillons libellés $(x_i, y_i)_{i=1}^N$, où $x_i \in R^d$ désigne les échantillons d'apprentissage et $y_i \in \{+1, -1\}$ désigne les labels associés (de quelles classes de modèle ω_1 et ω_2). L'apprentissage *SVM* standard aussi appelé *SVM* inductif (*ISVM*), essaie de séparer les données dans l'espace d'entrée avec les données d'entraînement disponibles en définissant un hyperplan: $f(x) : wx + b = 0$, de sorte que la distance entre les vecteurs les plus proches de l'hyperplan soit maximale. La marge géométrique maximale générée par l'hyperplan est : $\Phi(w) = 2 / \|w\|$.

Dans le cas des données d'apprentissage linéairement non séparables, l'objectif de la fonction de l'apprentissage *ISVM* est de trouver un hyperplan en résolvant un problème d'optimisation quadratique. Cette phase d'apprentissage du classifieur peut être formulée comme un problème d'optimisation qui, en utilisant la théorie de l'optimisation de Lagrange, conduit à la représentation duale suivante:

$$\begin{aligned} \max_{\alpha} \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(x_i, x_j) \right\} \\ \sum_{i=1}^N y_i \alpha_i = 0 \quad 0 \leq \alpha_i \leq C \quad i = 1, 2, \dots, N \end{aligned} \quad (3.16)$$

où α_i désigne les multiplicateurs lagrangiens et C est un paramètre de régularisation qui permet de contrôler la pénalité attribuée aux erreurs. La fonction de décision $f(x)$ est définie comme :

$$f(x) = \sum_{x_i \in SV} \alpha_i y_i K(x_i, x) + b \quad (3.17)$$

où SV représente l'ensemble des vecteurs de support. L'instance d'apprentissage x_i est un vecteur de support si l' α_i correspondant a une valeur non nulle. Pour un échantillon de test donné x , le signe de la fonction discriminante $f(x)$ définie en (3.17) est utilisé pour prédire son label de classe.

Les SVM ont été développés initialement pour traiter des problèmes binaires. Toutefois, les bases de données réelles soulèvent des problèmes avec plusieurs classes. Pour traiter ces problèmes multi-classes, les deux stratégies les plus utilisées sont le (Un-contre-un) et le (Un-contre-tous). Cependant contrairement au mode (Un-contre-un), le mode (Un-contre-tous ou OAA) permet intuitivement de classifier des données multi-labels (appartenant à plus d'une classe à la fois (voir Fig. 3.7) ou encore appelée un-contre-le-reste : dans cette stratégie, le problème multi-classes, avec k classes, est décomposé en k sous-problèmes binaires.

Afin d'aborder les problèmes multi-classes ici, nous adoptons la stratégie un contre tous (OAA), qui implique une architecture parallèle composée de n SVM binaires, une pour chaque classe d'information. Chaque SVM résout un problème de deux classes défini par une classe d'information contre toutes les autres [Melgani and Bruzzone, 2004]. Ainsi, un sous-problème pour une classe donnée, peut être ramené à un problème binaire avec les données appartenant à cette classe comme instances positives et les autres données comme instances négatives. Pour chaque problème, un hyperplan séparateur, (w_i, b_i) , est "appris" (voir équation 3.16) [Vapnik, 2001].

Les SVM reposent principalement sur la fonction de décision $(\langle w, x_i \rangle + b)$ qui indique l'appartenance d'une donnée à une classe. Dans le cas non séparable, on fait appel au kernel (ou fonction noyau). Les SVM utilisent la notion de la marge et la procédure de recherche de l'hyperplan séparateur pour classer les données. Cependant, il existe des cas où, il est impossible de trouver une séparation linéaire pour ces données. Les SVM sont incapables de résoudre un tel problème, car il est impossible de satisfaire toutes les contraintes : $y_i(\langle w, x_i \rangle + b) \geq 1$. Pour remédier à ce problème, l'idée de base des SVM est de reconsidérer le problème dans un espace de dimension supérieure ou éventuellement infinie. Ceci peut être réalisé en procédant à une transformation des données de sorte qu'elles soient linéairement séparables.



Figure 3.7-Transformation d'un problème non linéairement séparable en un problème linéairement séparable.

L'espace des entrées (input space) est l'espace où se trouvent les données avant la transformation. L'espace de redescription ou des caractéristiques (feature space) est l'espace des données qui ont subi la transformation. Le problème consiste alors à trouver l'hyperplan séparateur dans l'espace de redescription qui va séparer au mieux les données transformées. Formellement, nous appliquons aux vecteurs d'entrées x , une transformation non-linéaire Φ :

$$\begin{aligned} \Phi: \quad \mathbb{R}^n &\rightarrow \mathbb{R}^r \\ x &\rightarrow \Phi(x) \end{aligned} \quad \text{où : } r > n \quad (3.18)$$

En utilisant cette transformation, nous aboutissons au problème d'optimisation sous la forme primale suivante :

$$\begin{aligned} \min_{w, b, \xi, \rho} \quad & \frac{1}{2} \|w\|^2 \\ \text{s. c.} \quad & y_i (\langle w, \Phi(x_i) \rangle + b) \geq 1 \\ & i \in \{1, \dots, N\} \end{aligned} \quad (3.19)$$

et sa forme duale sera :

$$\begin{aligned} \min_{\alpha} \quad & - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle \\ \text{s. c.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \quad \sum_{i=1}^N \alpha_i \geq 0 \quad \forall i \in \{1, \dots, N\}, \alpha_i \geq 0 \end{aligned} \quad (3.20)$$

Cependant, l'utilisation de cette transformation implique le calcul du produit scalaire entre les vecteurs d'entrées dans l'espace de redescription. Ceci engendre des calculs extrêmement longs à effectuer vu que l'espace de redescription est plus grand que l'espace d'entrée. Pour pallier ce problème, les chercheurs ont proposé d'introduire la notion de fonction noyau qui est définie comme suit :

$$\begin{aligned} K: \quad \mathbb{R}^n \times \mathbb{R}^n &\rightarrow \mathbb{R} \\ (x_i, x_j) &\rightarrow \langle \Phi(x_i), \Phi(x_j) \rangle \end{aligned} \quad (3.21)$$

La fonction noyau vise deux objectifs spécifiques :

- La fonction prend en entrée deux points dans l'espace d'entrée et calcule leur produit scalaire dans l'espace de redescription. L'avantage d'une telle fonction est qu'il n'est pas nécessaire d'appliquer une transformation aux données afin de calculer leur produit scalaire dans l'espace de redescription. Ce calcul peut se faire directement à partir des données de l'espace d'entrée.
- Le deuxième objectif consiste à faire les calculs sans tenir compte de la transformation Φ car seule la fonction noyau intervient.

Le problème dual peut être réécrit comme suit :

$$\begin{aligned} \min_{\alpha} \quad & - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{s. c.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \quad \forall i \in \{1, \dots, N\}, \alpha_i \geq 0 \end{aligned} \quad (3.22)$$

et le classifieur aura la forme suivante :

$$h(x) = \text{sign} \left(\sum_{i \in S} y_i \alpha_i^* K(x_i, x_j) + b^* \right) \quad (3.23)$$

La construction de la fonction noyau impose le respect de certaines règles définies par le théorème de Mercer qui s'énonce comme suit :

Si un noyau $K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ est symétrique et semi-défini positif alors il admet un développement de la forme

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \quad (3.24)$$

La symétrie et la semi-défini-positivité sont des conditions nécessaires et suffisantes [Christiani and Shaw-Taylor, 2000] et sont définies comme suit :

Une fonction scalaire $K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ est semi-définie positive si et seulement si

$$\forall (\Psi: \mathbb{R}^n \rightarrow \mathbb{R}) \neq 0, \iint_{\mathbb{R}^n \times \mathbb{R}^n} K(x_i, x_j) \Psi(x_i) \Psi(x_j) dx_i dx_j \geq 0 \quad (3.25)$$

Une fonction scalaire $K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ est définie positive si et seulement si

$$\forall (\Psi: \mathbb{R}^n \rightarrow \mathbb{R}) \neq 0, \iint_{\mathbb{R}^n \times \mathbb{R}^n} K(x_i, x_j) \Psi(x_i) \Psi(x_j) dx_i dx_j > 0 \quad (3.26)$$

Une matrice carrée K de taille $N \times N$ est semi-définie positive (resp. définie positive) si et seulement si pour tout vecteur colonne $\Psi^T \in \mathbb{R}^N$ non nul $\Psi^T K \Psi \geq 0$ (resp. $\Psi^T K \Psi > 0$).

Autrement dit, si $K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ est défini positif alors il peut s'exprimer comme un produit scalaire dans un espace vectoriel où sont projetées les données (l'espace de redescription). Inversement, si on définit une correspondance entre des données d'entrées et un espace vectoriel alors le produit scalaire dans cet espace sera un noyau défini positif.

Cependant, cette condition est difficile à vérifier et elle ne donne aucune information sur la construction de la fonction noyau et sur la transformation Φ .

De plus, le théorème de Mercer permet aussi de construire de nouveaux noyaux sur la base de ceux déjà existants. En effet, Si K_1 et K_2 sont des noyaux alors les fonctions suivantes sont aussi des noyaux :

$$K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j) \quad (3.27)$$

$$K(x_i, x_j) = \alpha K_1(x_i, x_j) \quad \alpha \in \mathbb{R} \quad (3.28)$$

$$K(x_i, x_j) = K_1(x_i, x_j) K_2(x_i, x_j) \quad (3.29)$$

Différents types de noyaux sont proposés dans la littérature selon les problématiques étudiées. Nous nous proposons de citer les noyaux les plus utilisés :

Noyau linéaire : Il correspond au produit scalaire sans transformation et permet de traduire la forme traditionnelle des algorithmes sans utiliser le kernel trick :

$$K(x, x') = x^T \cdot x' \quad (3.30)$$

Noyau polynomial homogène : Il permet d'appliquer le principe de maximisation de la marge aux classifieurs polynômiaux, il est défini comme suit :

$$K(x, x') = (x^T \cdot x')^d \quad (3.31)$$

Noyau polynomial inhomogène : L'introduction d'une constante dans le produit scalaire permet d'inclure dans la transformation Φ tous les monômes d'ordre inférieur ou égal à d . Ce qui engendre un espace transformé de dimension supérieure à celle du noyau homogène, il sera défini comme suit :

$$K(x, x') = (x^T \cdot x' + c)^d \quad (3.32)$$

Noyau Gaussien : Les fonctions à base radiale (RBF Radial Basis Functions) sont définies par le fait qu'elles ne dépendent que de la distance entre leurs arguments : $\phi(x, y) = \phi(\|x - y\|)$. Le noyau Gaussien RBF applique ainsi une gaussienne sur la distance entre les exemples. On peut montrer que l'espace transformé dans ce cas est de dimension infinie, puisque les exemples d'une collection arbitrairement grande y sont linéairement indépendants. Le caractère radial présente la particularité de placer tous les exemples sur la sphère unité dans l'espace transformé ($\|\Phi(x)\|^2 = k(x, x) = \exp(0) = 1, \forall x$).

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (3.33)$$

Noyau sigmoïdal : La fonction de décision construite à l'aide du noyau sigmoïdal est pareille à celle d'un réseau de neurones à deux couches. Ce noyau est couramment utilisé et donne généralement de très bons résultats malgré qu'il ne respecte pas la condition de Mercer pour toutes les valeurs de P_1 et P_2 .

$$K(x, x') = \tan(P_1 \cdot x^T \cdot x' + P_2) \quad (3.34)$$

En conclusion, nous pouvons affirmer que les problèmes que nous avons soulevés lors de l'utilisation des SVM sont: le choix du paramètre de régularisation C , la fonction noyau utilisée et ses paramètres. Ce choix a un impact majeur sur les performances des SVM et doit être pris en considération lors de la représentation des données. En pratique, le choix du noyau se fait par expérimentation.

Dans les applications typiques d'exploration de données, étiqueter les grandes quantités de données est difficile, coûteux et prend beaucoup de temps si elles sont annotées manuellement. Pour éviter l'étiquetage manuel, l'apprentissage semi-supervisé utilise des données non libellées avec les données libellées dans le processus d'apprentissage.

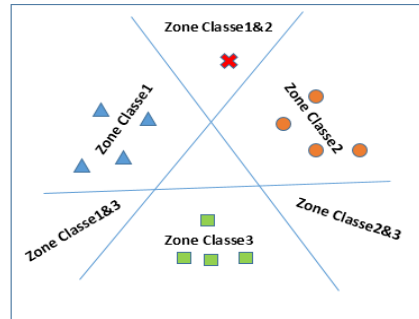


Figure 3.8 – Un SVM multi-classes en mode un contre tous permet intuitivement de faire une classification multi-labels (la donnée « x » appartient à la classe 1 et 2 à la fois)

La machine à vecteur de support transductive (TSVM) est dédiée au cas semi-supervisé et s'est révélée efficace pour améliorer la performance de classification. La SVM standard pour l'apprentissage est connue sous *SVM inductive* telle qu'on l'a présentée ci-dessus, par contre en contexte semi-supervisé, on utilise souvent la *SVM transductive*. Cette transductivité apporte sa contribution et améliore les performances du classifieur SVM.

4.2 Les SVM transductives (T-SVM)

Dans la grande majorité des problèmes de la vie réelle, les données libellées sont rarement disponibles en quantité. En effet, l'étiquetage des données nécessite une intervention humaine pouvant être coûteuse. Lorsque les données d'apprentissage ne sont pas caractéristiques de la source alors la fonction apprise risque d'être fortement biaisée. L'idée de l'approche transductive proposée par [Vapnik, 2001] est de prendre en compte les données non libellées pour induire une fonction générale. Cette fonction devant non seulement minimiser le risque sur l'ensemble d'apprentissage mais aussi sur l'ensemble des données de test.

La figure 3.9 illustre un exemple de problème avec des données libellées (instances positives et négatives) et des données non libellées (représentées par des croix noires). Lorsque seules les données libellées sont prises en compte, nous obtenons une marge contenant plusieurs points non libellés. Cependant, lorsque les points non libellés sont utilisés pour l'apprentissage, nous obtenons un SVM (représenté par des pointillés) de meilleure qualité.

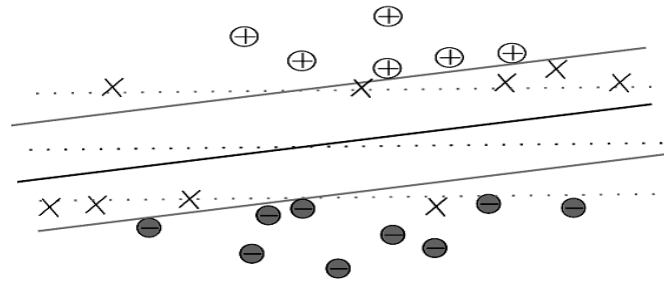


Figure 3.9 – Illustration de l'apprentissage SVM transductif

La machine à vecteur de support transductive (*TSVM*) est une version semi-supervisée de *SVM* [Vapnik, 1998]. Au cours de la phase d'entraînement, elle recherche progressivement (processus itératif) un hyperplan de séparation fiable dans l'espace du noyau en prenant en compte à la fois, les échantillons libellés et non libellés. Les données non libellées peuvent être utilisées comme source supplémentaire d'informations sur la marge pour *SVM*. Dans l'apprentissage transductif, l'objectif est de trouver un étiquetage des données non libellées, de sorte qu'une limite linéaire a la marge maximale sur les données à la fois libellées originales et les données non libellées (maintenant libellées) (définies ici comme des données transductives pour des raisons de commodité).

Cette tâche peut améliorer les performances de généralisation des *SVM*, en particulier lorsque des ensembles d'entraînement médiocre sont disponibles ou lorsque les échantillons d'entraînement disponibles sont inadéquats [Joachim, 1998].

Dans la littérature de nombreuses techniques semi-supervisées basées sur *TSVM* existent pour la classification des modèles [Joachim, 1999; Bruzzone et al., 2006]. Toutes ces méthodes tentent de trouver un hyperplan de décision passant par la région de faible densité de l'espace du noyau. Une bonne revue des approches semi-supervisées peut être trouvée dans [Zhu, 2005]. Dans [Joachim, 1999], l'auteur a résolu le problème d'optimisation quadratique pour la mise en œuvre du *TSVM* avec une application à la classification de texte. Cet algorithme est efficace lorsque le rapport entre les échantillons positifs et négatifs non libellés est connu au début de l'apprentissage transductif.

[Chapelle and Zien, 2005] ont proposé une méthode qui optimise la fonction de *SVM* transductive en utilisant la technique de descente en gradient pour déterminer la limite de décision dans les régions à faible densité de l'espace du noyau.

[Sinhhwani and Keerthi, 2006] ont proposé un algorithme rapide pour *TSVM* linéaire, adapté aux applications de texte à grande échelle.

Dans [Adankon and Cheriet, 2007], un critère supplémentaire est inclus avec la fonction objectif standard du *TSVM* et un algorithme génétique est utilisé pour optimiser cette fonction objectif. On a proposé un algorithme *TSVM* progressif qui sélectionne itérativement un échantillon positif et un échantillon négatif comme échantillons transducteurs des échantillons non libellés disponibles qui sont à l'intérieur de la marge de décision *SVM* et qui ont une distance minimale entre la marge positive et la marge négative respectivement. La méthode utilise également l'ajustement dynamique pour réduire l'étiquetage erroné d'échantillons transductifs sélectionnés.

Dans [Bruzzone et al., 2006], ils ont modifié l'algorithme présenté dans [Chen et al., 2003] pour sélectionner un lot de motifs positifs et négatifs comme échantillons transducteurs à partir des modèles non libellés disponibles à chaque itération du processus d'apprentissage transducteur.

L'approche *SVM* transductive a été proposée en impliquant des échantillons non libellés dans la phase d'entraînement [Vapnik, 2001]. La *SVM* transductive (*TSVM*) est une méthode de classification à large marge semi-supervisée basée sur l'hypothèse de séparation à faible densité. Semblable au *SVM* traditionnel, *TSVM* recherche un hyperplan avec la plus grande marge pour séparer les classes et prend simultanément en compte des exemples libellés et non libellés. Les descriptions détaillées et les preuves des concepts peuvent être trouvées dans [Joachim, 1999].

Lors de l'itération initiale, la *SVM* inductive standard est utilisée pour obtenir un hyperplan de séparation en utilisant l'ensemble d'apprentissage uniquement. Ensuite, en fonction de la distance à partir de cet hyperplan, des labels sont attribuées aux instances non libellées qui sont ainsi appelées données semi-libellées. Ensuite, selon un critère défini, des instances transductives choisies parmi les instances semi-libellées sont incluses dans l'ensemble d'apprentissage original. L'ensemble d'apprentissage résultant est utilisé aux itérations suivantes pour trouver un hyperplan discriminant plus fiable séparateur avec la marge maximale et se déduit comme suit:

Soit un groupe d'exemples libellés indépendants et identiquement distribués :

$\{(x_1, y_1), \dots, (x_l, y_l)\} \in R^n \times R$, $i = 1, \dots, l$, $y_i = \{-1, +1\}$ et u exemples non libellés $\{x_l, \dots, x_{l+u}\}$
En général, le processus d'apprentissage de *TSVM* peut être formulé comme un problème d'optimisation suivant :

$$\begin{aligned} \min (y_1, \dots, y_l, w, b, \xi_1, \dots, \xi_l, \xi_{l+1}, \dots, \xi_{l+u}) & \frac{1}{2} \|w\|^2 + C_1 \sum_{i=1}^l \xi_i + C_2 \sum_{i=l+1}^{l+u} \xi_i \\ \text{subject to} \quad & \forall_{i=1}^l: y_i [w \cdot x_i + b] \geq 1 - \xi_i; \quad \xi_i \geq 0 \\ & \forall_{i=l+1}^{l+u}: y_i [w \cdot x_i + b] \geq 1 - \xi_i; \quad \xi_i \geq 0 \end{aligned} \quad (3.35)$$

Afin de gérer l'apprentissage inséparable et les données transductives, de la même façon que les *SVM* transductives, les variables molles ξ_i et ξ_j et les valeurs de pénalité associées C_1 et C_2 à la fois des échantillons d'entraînement et de transduction sont introduites. Dans le processus d'apprentissage des *TSVM*, le but de C_1 et C_2 est de contrôler le nombre d'échantillons mal classifiés appartenant à l'ensemble d'entraînement original et à l'ensemble non libellé, respectivement. En augmentant leurs valeurs, la pénalité associée aux erreurs sur les échantillons d'entraînement et de transduction augmente. En d'autres termes, plus le paramètre de régularisation est grand, plus l'influence des échantillons associés sur la sélection de l'hyperplan discriminant est importante. q ($q \leq u$) est le nombre d'échantillons transducteurs choisis à chaque itération de l'apprentissage transductif et u est le nombre d'exemples non libellés.

Donc, C_1 et C_2 sont des paramètres spécifiés par l'utilisateur, qui sont utilisés pour pénaliser les échantillons mal classifiés. C_2 est appelé «facteur d'effet» des exemples non libellés dans le processus d'apprentissage. $C_2 \xi_j$ est appelé «terme d'effet» du $j^{\text{ème}}$ exemple non libellé dans la fonction objectif. Le processus d'apprentissage du *TSVM* est de résoudre le problème d'optimisation ci-dessus. L'algorithme d'apprentissage *TSVM* peut être décrit comme suit [Wang et al., 2015]:

- Étape 1: Spécifier les paramètres C_1 et C_2 , effectuer un apprentissage inductif sur les instances libellées et obtenir un classifieur initial. Spécifier q - un nombre estimé d'exemples libellés positifs dans les exemples non libellés.
- Étape 2: Calculer les valeurs de la fonction de décision pour tous les exemples non libellés en utilisant le classifieur initial. Les exemples ayant les q valeurs de fonction de décision les plus importantes sont libellés comme positifs et les restants étiquetés comme négatifs. Définir un facteur d'effet temporaire C_{temp} .
- Étape 3: Réapprendre la *SVM* en fonction de tous les exemples. Pour le classifieur nouvellement produit, commuter les labels d'une paire d'exemples non libellé libellé différents en utilisant une certaine règle pour abaisser autant que possible la valeur de la fonction objectif dans (3.40). Cette étape est répétée jusqu'à ce qu'aucune paire d'exemples ne rencontre l'état ou la condition de commutation.
- Étape 4: Augmenter uniformément la valeur de C_{temp} et retourner à l'Étape 3. Sur $C_{temp} \geq C_2$, l'algorithme est terminé et les labels de tous les exemples non libellés sont renvoyés.

- Problématique en *SVM* transducteurs

TSVM peut obtenir de meilleures performances que l'apprentissage inductif car il prend en compte l'information de distribution, qui est implicitement incorporée dans le grand nombre des exemples non libellés. Toutefois, elle présente également certains inconvénients, tels que la minimisation de sa fonction objectif (problème de programmation quadratique non convexe) et aussi le choix du paramètre q (problème de pré-réglage de q , voir l'étape 2 de l'algorithme de *TSVM* ci-dessus) doit être précisé à l'avance et il n'y a pas d'accord tacite sur l'utilisation de plus d'exemples non libellés peut mener à une meilleure performance d'apprentissage.

Les limites des techniques existantes semi-supervisées basées sur *TSVM* et même progressive *PTSVM* dans la littérature [Chen et al., 2003; Bruzzone et al., 2006], itérativement sélectionnent des échantillons transductifs à partir des instances semi-étiquetées qui sont à l'intérieur et plus proches des limites de la marge *SVM*. Il en résulte une probabilité élevée de sélectionner des instances erronées comme échantillons transductifs lorsque l'hyperplan de décision initial de *SVM* est très faible, c'est-à-dire dans les cas où il passe par une région incorrecte de l'espace du noyau.

Au fait, cela peut introduire des labels erronés dans les données d'apprentissage, car l'étiquetage se fait machinalement, et de telles erreurs d'étiquetage sont critiques pour la

performance de classification (problème d'exploitation des exemples non libellés, voir Fig.3.10).

La figure 3.10 (a) représente une situation dans laquelle une marge de SVM délimitée est proche de l'hyperplan de décision réel (c'est-à-dire qu'elle passe à travers la zone de faible densité de l'espace caractéristique du noyau) et l'autre est loin de celui-ci. Maintenant, si on applique les méthodes *TSVM* classiques pour sélectionner des échantillons transductifs, elles choisissent quelques instances semi-libellées comme échantillons transductifs qui changent également la limite de la marge qui passe par la région de faible densité vers une mauvaise direction.

En conséquence, un hyperplan de mauvaise décision sera généré. La raison principale est que ces algorithmes n'imposent aucune contrainte explicite au processus de sélection d'échantillon transductif de sorte que la limite de la marge qui passe par la région de densité élevée ne peut être déplacée vers la région de faible densité.

La figure 3.10 (b) illustre une autre situation extrême où les limites de la marge positive et négative passent à travers les deux classes. Dans ce cas également, les méthodes *TSVM* classiques peuvent ne pas sélectionner les échantillons transductifs appropriés. En conséquence, la performance de classification peut être dégradée.

En Fig.3.10, les instances appartenant à la classe (-1) et (+1) sont représentées sous forme de carrés et cercles blancs, respectivement. Les premiers échantillons libellés pour la classe (1) et (+1) sont affichés comme des carrés et des cercles noirs, respectivement. L'hyperplan de séparation est représenté en pointillés alors que les lignes pleines définissent la marge.

Les carrés et les cercles en pointillés mettent en évidence les échantillons transductifs sélectionnés par la 1^{ère} itération des méthodes *TSVM* conventionnelles étiquetées (-1) et (+1), respectivement :

(a) lorsqu'une marge de décision est proche de la valeur réelle de l'hyperplan actuel de décision et l'autre marge est loin du plan;

(b) lorsque les marges de décision positives et négatives traversent toutes deux l'espace des deux classes à la fois.

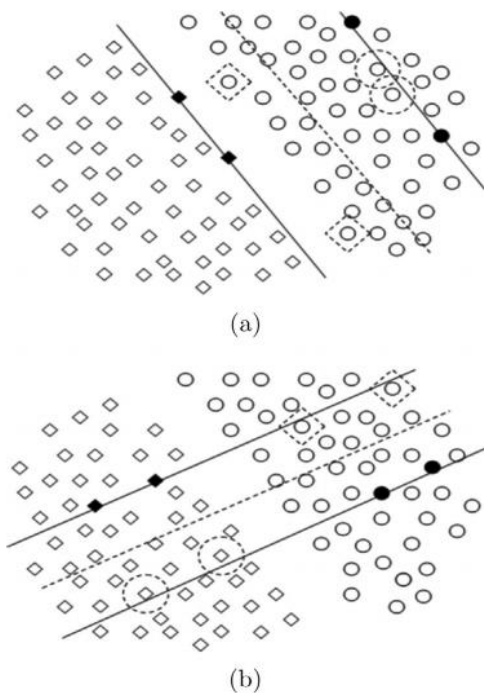


Figure 3.10 – Problématique du *T-SVM*

Pour le prééclage du problème q , en général, le choix de la valeur correcte de N est très difficile avant l'entraînement de *TSVM*, car la distribution des données est difficile à estimer (voir Fig.3.10). Si q estimé *a priori* est incorrect, il conduira à un classifieur qui ne peut pas décrire la distribution des données réelles. Ainsi, cet inconvénient limite sérieusement les applications de *TSVM*. Pour résoudre ce problème, certains auteurs dans [Chen et al., 2003;

Bruzzone et al., 2006] ont proposé une version améliorée appelée *SVM progressive (PTSVM)* qui consiste à étiqueter à la fois, un exemple négatif et un exemple positif simultanément en fonction de la valeur absolue de la fonction de décision absolue. Mais cet algorithme de *PTSVM* ne convient que pour résoudre un petit nombre d'échantillons non libellés.

Les algorithmes progressifs *TSM (PTSVM)* existant dans la littérature [Chen et al., 2003; Bruzzone et al., 2006] sélectionnent itérativement les modèles les plus certains en tant qu'échantillons transductifs à partir des motifs non libellés disponibles qui sont à l'intérieur des limites de la marge *SVM*. La certitude d'un échantillon est mesurée en considérant seulement sa distance par rapport à la limite de la marge *SVM* la plus proche. Il peut en résulter une probabilité élevée de sélectionner des modèles erronés en tant qu'échantillons transducteurs (les étiquettes réelles des instances sont différentes des étiquettes qui leur sont automatiquement attribuées), en particulier lorsque l'hyperplan de décision initial est pauvre, c'est-à-dire passe par une région incorrecte de l'espace du noyau. Ainsi, la précision de classement finale peut être dégradée.

Dans [Singla et al., 2014], une technique semi-supervisée est proposée basée sur l'apprentissage *PTSVM* qui atténue la limitation susmentionnée. La technique proposée utilise non seulement la distance de la marge de *SVM* la plus proche liée mais elle exploite également les propriétés de l'approche des k -voisins les plus proches (k -NN) et l'hypothèse de clusters pour sélectionner les échantillons les plus certains comme échantillons transducteurs à chaque itération du processus d'apprentissage.

De ce fait, on trouve de plus en plus, d'articles combinant *SVM* avec *SOM* semi-supervisée pour mieux gérer ces échantillons transductifs, en s'appuyant d'une part sur la carte *SOM* pour identifier les échantillons disponibles importants appartenant aux régions de faible densité de l'espace des caractéristiques, en supposant toutefois que *SOM* préserve la propriété topologique de l'espace des entrées [Patra et al., 2014].

Il convient de noter que, comme toute autre technique non linéaire de réduction de dimensionnalité, un réseau neuronal *SOM* ne garantit pas la préservation de la topologie quelque soit la nature du problème donné, en particulier lorsque la caractéristique dimensionnelle est très élevée [Villmann et al., 1997] et cela se répercute sur le choix des clusters les plus informatifs et représentatifs.

Dans certaines expériences [Singla et al., 2014], les auteurs ont adopté un classifieur *SVM* avec noyau *RBF (Radial Basis Functions)*. Les paramètres *SVM* avec kernel gaussien, ont été obtenus en appliquant la technique de validation croisée [Patra and Bruzzone, 2011]. La procédure de validation croisée vise à sélectionner les meilleures valeurs pour les paramètres de la *SVM* initiale.

La même fonction kernel *RBF* est également utilisée pour implémenter la technique du kernel k -NN. La valeur de k pour le noyau k -NN est également calculée automatiquement en utilisant la technique de validation croisée. Pour examiner progressivement l'influence des échantillons transductifs pour définir l'hyperplan de décision correspondant, la valeur initiale

du paramètre de régularisation pour les motifs transductifs sélectionnés C_2 devrait être faible car au stade initial de l'apprentissage, l'hyperplan de décision *SVM* est mal défini. Lors des itérations suivantes, puisque la confiance en étiquetage augmente, la valeur de C_2 augmentera également.

Pour évaluer l'efficacité des techniques issues des *TSVM* classiques déjà proposées en littérature où les exemples positifs sont choisis comme transductifs, des variantes transductives progressive (*PTSVM*) et modifiée (*MPTSVM*) ont été explorées [Bruzzone et al., 2006].

Dans *PTSVM*, à chaque itération de l'apprentissage transductif, une instance positive et une instance négative semi-libellées sont sélectionnées comme échantillons transductifs parmi ceux disponibles qui sont à l'intérieur de la limite de la marge *SVM* tout en étant les plus proches des marges positives et négatives, respectivement. Dans *MPTSVM*, à chaque itération, un groupe d'instances positives et négatives semi-libellées est sélectionné comme lot d'échantillons transductifs en utilisant le même critère que celui utilisé dans *PTSVM*. La taille du lot ou batch est déterminée automatiquement en fonction du nombre de supports vecteurs positifs et négatifs disponibles à une itération particulière du processus d'apprentissage.

Le *SVM* multiclass avec l'architecture OAA (One-Against-All) standard est généralement implémenté manuellement en utilisant la bibliothèque *LIBSVM* (pour l'interface Matlab) [Chang and Lin, 2001]. Tous les algorithmes qu'on présentera dans cette thèse ont été mis en œuvre sous Matlab.

Il est évident que lors d'apprentissage de la *TSVM*, des exemples non libellés d'apprentissage sont utilisés, mais on ne peut conclure, d'après l'équation (3.35) ci-dessus, qu'en utilisant plus d'exemples d'entraînement, on va certainement augmenter la performance d'apprentissage, d'autant plus que dans de nombreuses applications pratiques, des exemples non libellés pour les méthodes d'apprentissage semi-supervisées peuvent provenir d'environnements différents, où la distribution des exemples est complexe, inconnue et contient très probablement du bruit.

Dans la suite de ce chapitre, on étudiera une nouvelle technique semi-supervisée basée sur l'apprentissage actif qui aborde ce genre de problématique mentionnée ci-dessus lors de la sélection d'échantillons transductifs à chaque itération du processus d'apprentissage. Donc, on s'intéressera de près, à la sélection des modèles transductifs.

Selon les caractéristiques de l'apprentissage actif, on tire partie des connaissances existantes et on initie la sélection des instances les plus probables pour résoudre le problème du choix des instances transductives ou semi-libellées. Cet apprentissage actif réduit efficacement le nombre d'exemples requis pour l'évaluation, ce qui peut être utilisé pour la *TSVM* pour améliorer la performance de la sélection des exemples non libellés. Il en résulte la sélection des exemples les plus favorables au modèle de classification *TSVM*, ce qui améliore ainsi sa performance.

En apprentissage actif, on doit s'attendre grâce à une meilleure sélection des instances semi-libellées et transductives plus informatives à obtenir une précision beaucoup plus grande avec moins de paires d'apprentissage libellées.

5. Apprentissage actif

L'apprentissage supervisé ne requiert que des données étiquetées pour l'apprentissage. Les résultats de classification dépendent de la quantité et de la qualité de ces échantillons labellisés. Cependant, la production d'échantillons étiquetés corrects est souvent difficile, coûteuse et longue, car cela nécessite l'effort d'annotateurs humains expérimentés. D'autre part, les données non étiquetées sont relativement faciles à collecter, bien qu'elles n'aient aucune utilité dans l'apprentissage supervisé.

Deux approches d'apprentissage par machine populaires pour traiter ce problème sont l'apprentissage actif et l'apprentissage semi-supervisé. L'apprentissage actif développe l'ensemble d'entraînement original selon un processus interactif impliquant un superviseur qui peut assigner l'étiquette correcte aux instances de données inconnues [Patra and Bruzzone, 2012]. Le but de l'apprenant actif est de sélectionner le moins d'échantillons possibles mais toutefois les plus informatifs afin d'apprendre avec précision de ces données additionnellement libellées. En revanche, l'apprentissage semi-supervisé exploite les données non libellées, avec les données libellées, pour construire de meilleurs classifieurs [Nigam, 2001]. En conséquence, sous des hypothèses appropriées, il faut moins d'effort humain pour l'étiquetage en apprentissage actif, ainsi il devient d'un grand intérêt à la fois en théorie et en pratique.

L'apprentissage actif et l'apprentissage semi-supervisé font face au même problème, c'est-à-dire la rareté des données ainsi que la difficulté de les obtenir. Il est tout à fait naturel de combiner les deux types d'apprentissage en mode semi-supervisé pour aborder la problématique des deux côtés.

[McCallum, 1999] utilise le procédé de l'expectation-maximisation *EM* avec des données non libellées dans l'algorithme d'apprentissage actif. [Muslea et al., 2002] proposent le *CO-EMT* qui combine l'apprentissage multi-vision (co-apprentissage) avec l'apprentissage actif. [Zhou et al., 2012] appliquent l'apprentissage semi-supervisé ainsi que l'apprentissage actif à la récupération d'images basée sur le contenu.

L'apprentissage actif est un problème bien étudié dans la littérature d'apprentissage automatique, plusieurs techniques ont été développées au cours des dernières années et un examen minutieux de celles-ci peut être trouvé dans [Settles, 2012].

L'idée clé de l'apprentissage actif est de minimiser le coût de l'étiquetage en permettant à l'apprenant de demander les étiquettes des instances de données les plus informatives non étiquetées. Ces requêtes sont posées à un oracle, par ex. un annotateur humain, qui comprend la nature du problème. De cette façon, un apprenant actif peut considérablement réduire le nombre de données libellées requises pour construire le classifieur (voir Fig.3.11).

L'apprentissage actif a été développé de manière substantielle pour soutenir l'apprentissage à un seul label, où chaque objet (instance) du dataset est associé à un seul label de classe. Cependant, ce n'est pas le cas dans l'apprentissage multi-labels, où chaque objet est associé à un sous-ensemble de labels. En raison du grand nombre de problèmes réels qui entrent dans cette catégorie et des défis intéressants qu'elle pose, l'apprentissage multi-labels a suscité un grand intérêt au cours de la dernière décennie [Yang et al., 2009].

On verra au chapitre 4, l'apprentissage multi-labels en contexte semi-supervisé avec les différentes approches utilisées sur différents datasets tirés de la bibliothèque Mulan [Tsumakas et al., 2011], avec l'apport de l'apprentissage actif, y compris les différentes stratégies adoptées en classification multi-labels.

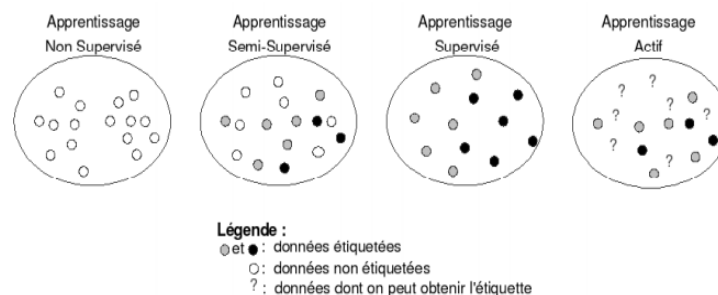


Figure 3.11 – Quelles données pour quel type d'apprentissage [Bondu and Lemaire, 2008].

Enfin, l'apprentissage actif permet au modèle de construire son ensemble d'apprentissage au cours de son entraînement, en interaction avec un expert (humain). L'apprentissage débute avec peu de données étiquetées. Ensuite, le modèle sélectionne les exemples (non étiquetés) qu'il juge les plus "instructifs" et interroge l'expert à propos de leurs étiquettes.

La particularité de l'apprentissage actif réside dans l'interaction du modèle avec son environnement. Contrairement à la stratégie "passive" où les exemples sont choisis avant l'apprentissage, de manière aléatoire, les stratégies "actives" permettent d'accélérer l'apprentissage en considérant d'abord les exemples les plus informatifs.

Les algorithmes d'apprentissage actif sont des procédés d'échantillonnage itératifs, où un modèle de classification est régulièrement adapté en l'alimentant avec de nouvelles instances libellées correspondantes à celles qui sont les plus bénéfiques pour l'amélioration de la performance du modèle. Ces instances sont habituellement trouvées dans les zones d'incertitude du modèle (à faible densité) et leur inclusion dans le jeu d'entraînement oblige le modèle à résoudre les régions de faible confiance.

Un autre avantage des méthodes d'apprentissage actif est qu'elles peuvent souvent aider à éliminer les cas bruités des données, ce qui peut être bénéfique du point de vue de la précision. En fait, certaines études [Schohn and Cohn, 2000] ont montré qu'une méthode d'apprentissage actif soigneusement conçue peut parfois fournir une meilleure précision que ce qui est disponible à partir des données de base. C'est-à-dire qu'on a plus besoin d'un utilisateur au cours du processus d'apprentissage pour corriger les classifieurs qu'au début du

processus, c'est-à-dire à l'entrée avec les instances libellées. Le but est d'obtenir une exactitude à moindre coût justifiée, tout est alors dans la sélection des instances à libeller.

De toute évidence, étant donnée la valeur différentielle de différents enregistrements, une question importante qui se pose dans l'apprentissage actif est la suivante: Comment sélectionner les instances des données sous-jacentes à libeller afin d'obtenir l'apprentissage le plus efficace pour un niveau d'effort donné?

Différents critères de performance peuvent être utilisés pour quantifier et ajuster les compromis entre l'exactitude et le coût, mais l'objectif plus large de tous les critères est de maximiser l'étiquetage en dépensant le minimum d'efforts pour sélectionner des exemples qui maximisent la précision autant que possible. Une excellente étude sur l'apprentissage actif peut être trouvée dans [Settles, 2012].

5.1 Concepts et définitions

Le problème de l'échantillonnage sélectif a été posé formellement par [Muslea, 2002] (voir l'algorithme suivant). Celui-ci met en jeu une heuristique sous forme d'une fonction d'utilité, $Hutile(u, M)$, qui estime l'intérêt d'une instance u pour l'apprentissage du modèle M . Grâce à cette fonction, le modèle présente à l'utilisateur les instances pour lesquelles il espère la plus grande amélioration de ses performances. L'algorithme est générique dans la mesure où seule la fonction $Hutile(u, M)$ doit être modifiée pour exprimer une stratégie d'apprentissage actif particulière.

Algorithme 9 : Algorithme général de l'apprentissage actif en batch

Entrée: données libellées $\{(x_i, y_i)\}_{i=1}^l$, données non libellées $\{x_j\}_{j=l+1}^{l+u}$, nombre de points de données q à ajouter à chaque itération (en définissant le lot S des points sélectionnés).

1. Initialement, soit l'ensemble d'apprentissage initial $L = \{(x_i, y_i)\}_{i=1}^l$ et le groupe des candidats

$$U = \{x_j\}_{j=l+1}^{l+u}$$

2. **Répéter**

3. Apprendre un model avec l'ensemble d'apprentissage L .

4. **Pour** tout candidat in U **faire**

5. Evaluer une heuristique définie par l'utilisateur

6. **fin Pour**

7. Classer les candidats dans U selon le score de l'heuristique

8. Sélectionner les q points de données les plus intéressants, $\{x_k\}_{k=1}^q$

9. L'utilisateur assigne des labels aux point de données sélectionnés, $S = \{x_k, y_k\}_{k=1}^q$

10. Ajouter le lot S au nouvel ensemble d'apprentissage $L^{new} = L^{old} \cup S$

11. Supprimez le lot S de l'ancien groupe des candidats $U^{new} = U^{old} \setminus S$

12. **Jusqu'à** ce que le critère d'arrêt est atteint.

Pour une itération donnée, l'algorithme sélectionne à partir du groupe (pool) U , les q candidats qui maximiseront en même temps le gain de performance et réduiront l'incertitude du modèle lorsqu'il sera ajouté à l'ensemble d'entraînement courant L . Une fois le groupe des

instances $\{\mathbf{x}_m\}_{m=1}^q \subset U$ a été sélectionné, il est libellé par l'utilisateur, c'est-à-dire que les étiquettes $\{y_m\}_{m=1}^q$ sont découvertes. Enfin, l'ensemble $S = \{\mathbf{x}_k, y_k\}_{k=1}^q$ est ajouté à l'ensemble d'entraînement courant ($L^{new} = L^{old} \cup S$) et supprimé du groupe des instances non libellées au départ ($U^{new} = U^{old} \setminus S$), c'est-à-dire qu'on garde le complément de U par rapport à S . Le processus est réitéré jusqu'à ce qu'un critère d'arrêt soit atteint.

Dans l'apprentissage actif, les requêtes ou les questions les plus informatives, étant donnés les objectifs de la tâche de classification, sont sélectionnées par l'algorithme d'apprentissage à la différence de l'apprentissage passif où les instances d'apprentissage sont sélectionnées au hasard. L'apprentissage actif peut être effectué selon l'établissement de plusieurs paramètres distincts et selon des stratégies dédiées qui seront traitées dans le chapitre 4, selon le contexte mono ou multi-label.

L'idée de base dans l'apprentissage actif est d'estimer la valeur de l'étiquetage des instances non libellées. Dans l'algorithme suivant est donné une version plus détaillée de l'apprentissage actif général de base, où les classifieurs h_t sont issus des instances libellées pour guider l'étiquetage des instances non libellées, beaucoup plus nombreuses, moyennant la maximisation de la fonction requête d'utilité $f_q()$ pour aboutir à q_i .

Algorithme 10: Algorithme séquentiel détaillé général de l'apprentissage actif

1. Entrée: L'ensemble d'apprentissage initial $L_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$; l'ensemble des instances non libellées $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$; $f_q()$ fonction d'utilité de requête.
 2. Sortie: h_t , classifieur appris
 3. Initialiser L_1 , h de U
 4. $i = 1$
 5. **Tant que** les critères d'arrêt ne sont pas atteints **faire**
 6. h_t = apprendre (L_i), générer le classifieur h_t en utilisant l'ensemble actuel libellé L_i
 7. Utiliser h_t pour classifier les instances dans l'ensemble actuel des données non libellées
 8. $q_i = \arg \max_{\mathbf{x}_j} f_q(\mathbf{x}_j)$, $\mathbf{x}_j \in U_i$, sélectionner $q_i = \mathbf{x}_j \in U_i$ qui maximisent l'utilité de la requête
 9. Demander à l'oracle le label y_j pour $\mathbf{x}_j$
 10. $L_{i+1} = L_i \cup \langle \mathbf{x}_j, y_j \rangle$
 11. $U_{i+1} = U_i \setminus \mathbf{x}_j$
 12. $i++$
 13. **fin Tant que**
 14. **retourner** h_t
-

$\langle \mathbf{x}_j, y_j \rangle$ où $\mathbf{x}_j$ est la $j^{\text{ième}}$ instance (ensemble de caractéristiques descriptives) et y_j est sa vraie classe (label).

Donc, il ressort qu'un processus d'apprentissage actif nécessite une interaction entre l'utilisateur et le modèle M : le premier fournit l'information libellée et la connaissance des classes souhaitées, tandis que le second fournit à la fois sa propre interprétation de la

distribution des classes et les instances S les plus pertinentes nécessaires pour résoudre les écarts rencontrés.

Ce point relatif au modèle de sélection, est crucial pour le succès d'un algorithme d'apprentissage actif: la machine a besoin d'une stratégie pour classer les instances dans le groupe U . Ces stratégies, ou heuristiques, différencient les algorithmes proposés dans la littérature et peuvent être généralement divisées en heuristiques fondées sur des comités, heuristiques à base de probabilité postérieure et heuristiques basées sur les larges marges (SVM) [Tuia et al., 2013].

5.2 Principaux scénarios

Contrairement au modèle passif d'apprentissage supervisé où les valeurs des variables cibles sont obtenues en tenant compte seulement globalement et surtout moyennement de l'algorithme d'apprentissage, et à l'image de l'apprentissage actif sur la classification des textes qui a été bien étudié; sur la base de la stratégie adoptée de sélection de l'échantillon où l'apprenant interroge interactivement la supervision des instances de son propre choix, il existe essentiellement trois principaux scénarios d'apprentissage actif [Settles, 2010; Aggarwal et al., 2014]:

1) synthèse d'une requête d'adhésion (membership) : L'apprenant actif sélectionne les données à libeller qui sont en plus grand désaccord entre plusieurs membres du comité (classifieurs) de l'espace des différentes versions proposées. Le travail de requête par comité [Seung et al., 1992] a été le premier algorithme de ce genre et dans [Tong, 2001], l'idée est étendue à l'apprentissage actif des SVM .

2) à base de flux (streaming) : Les données sont considérées séquentiellement, en décidant si un objet non libellé doit l'être ou non [Cohn et al., 1995]. Dans de tels cas d'échantillonnage d'incertitude, l'apprenant actif étiquette itérativement les données non libellées sur lesquelles l'hypothèse actuelle est la plus incertaine [Luo et al., 2005].

3) à base de groupe (pool) : Tout le groupe des données non libellées est évalué avant de sélectionner un ou plusieurs objets à libeller. Ce type de scénario se base sur la réduction des erreurs attendues [Tong, 2001]. La stratégie vise à étiqueter les données pour minimiser l'erreur attendue sur les données non libellées. Habituellement, il nécessite un effort coûteux de calcul sur l'estimation de l'erreur attendue, puisque chacune des données non libellées associée à chaque étiquetage possible doit être évaluée.

Dans nos expériences, on s'intéressera à ce dernier scénario, comme il convient pour les datasets choisis reflétant un grand nombre de problèmes réels, tels que la classification des textes, la classification et la récupération des images, la classification vidéo, la reconnaissance de la parole et le diagnostic du cancer [Cortes and Vapnik, 1995; Joachim, 1998].

Cependant, la plupart des objectifs de recherche précédents concernaient des problèmes de classification uni-label. La stratégie de sélection d'échantillon évalue chaque donnée non

libellée en supposant qu'elle ne comporte qu'un seul label. Par exemple, la stratégie d'échantillonnage de l'incertitude se focalise sur la mesure de la confiance de la classe la plus probable, et la stratégie de réduction des erreurs permet d'estimer l'erreur tout en considérant les cas d'un seul label. Ainsi, ces stratégies ne peuvent pas être directement appliquées dans la classification en contexte multi-label. Comme on le verra, au chapitre 4 suivant, peu de recherches en apprentissage actif multi-label ont été explorées [Brinker, 2006], ce qui nous incite à davantage de motivations.

Pour notre travail de recherche, on a utilisé l'apprentissage actif multi-label. Il se décompose en un problème à plusieurs classifications binaires multi-label en utilisant l'approche un-contre-tous. La stratégie de sélection minimise la plus petite marge *SVM* parmi tous les problèmes de classification binaire. L'approche ne tient pas compte de l'information multi-label et traite toutes les classes de façon égale.

Dans [Li et al., 2004], une méthode *SVM* d'apprentissage actif a été proposée pour une classification multi-label d'images. Il sélectionne les données non libellées qui ont la valeur de perte moyenne maximale sur les classes prédites. Le problème de classification multi-label est également considéré comme des tâches de classement binaires. Un seuil de valeur de perte est estimé pour chaque classifieur binaire, puis utilisé pour décider des classes prédites pour les données non libellées.

Cependant, cette méthode n'est pas appropriée pour la tâche de classification de texte. Parce qu'il va introduire beaucoup plus de coût si un document est lu plusieurs fois. Évidemment, le coût de lire un document et de juger son label est beaucoup plus élevé que celui d'une image. [Esuli and Sebastiani, 2009] ont proposé plusieurs stratégies d'apprentissage pour la classification de texte multi-label. Chaque stratégie de sélection consiste en une règle pour combiner la sortie des classifieurs binaires individuels, selon trois dimensions orthogonales: preuve, classe et poids. Aussi, ils ne tiennent pas compte du résultat de prédiction d'étiquette pour chaque instance dans la stratégie de sélection.

Dans nos travaux, nous avons adopté l'approche d'apprentissage actif basée sur les groupes qui est habituellement la plus utilisée dans la littérature en supposant qu'on a un groupe de données partiellement libellées. Aussi, on considère comme classifieur de base multi-label, celui du *SVM* de part son succès dans différents domaines déjà cités et son utilisation de plus en plus fréquente en mode multi-label. On tentera, par la suite, d'appliquer à la carte *SOM* l'apprentissage actif pour nos six datasets choisis de différents domaines d'application.

6. Conclusion

L'apprentissage semi-supervisé combine des données libellées et non libellées pendant l'apprentissage pour améliorer les performances, selon deux aspects :

- le regroupement semi-supervisé: utilisant une petite quantité de données libellées pour aider, guider et biaiser (polariser) le regroupement des données non libellées.
- la classification semi-supervisée: l'apprentissage sur des données libellées exploite les données supplémentaires non libellées, ce qui se traduit souvent par un classifieur plus précis.

Parmi les algorithmes de clustering semi-supervisé basé sur la recherche d'un bon partitionnement, des modifications pour tenir compte des contraintes (must-link, cannot-link) sur les données libellées lors du clustering ont été considérées. Côté expérimental et en contexte multi-labels, on a préféré utiliser la carte topologique *SOM* et ses variantes (*SOM_Y* et *SOM mixte*) pour une bonne cartographie des données à l'entrée et aussi dans certains cas de les utiliser non comme c'est le cas fréquent en littérature servant d'entrée aux classifieurs *SVM* mais tout simplement comme concurrents aux *SVMs* pour l'apprentissage actif.

Dans ce travail, on propose d'utiliser cette carte topologique auto-organisatrice *SOM_mixte*, basée sur la pondération des instances d'entrée pour analyser les données mixtes (attributs des instances et des labels). Ces pondérations des attributs indiquent à un utilisateur l'importance relative de chacun des différents attributs pour la discrimination des classes. C'est un modèle de quantification qui fournit un ensemble conséquent de prototypes qui possèdent la propriété d'être facilement interprétables (les prototypes et les données appartiennent au même espace).

Parmi les algorithmes semi-supervisés et qu'on a utilisés par la suite, la Transductive *SVM* (*TSVM* avec terme additionnel sur les instances non libellées) et ses variantes (progressive et celle modifiée) permettent d'améliorer les scores et rendent les classifieurs plus performants en différents domaines applicatifs.

Cependant, il existe certaines déficiences dans le *TSVM*, telles que le prééglage du nombre d'échantillons de classe positifs, l'échange fréquent d'étiquette de classe et son exigence pour une plus grande quantité de données non libellées. Pour faire face à ces défauts, d'autres algorithmes ont été proposés semi-supervisés basés sur l'apprentissage actif combiné à la *TSVM*. L'algorithme progressif transductif *PTSVM* applique l'apprentissage actif pour sélectionner les instances les plus informatives pour améliorer la performance de classification.

L'utilisation de l'apprentissage actif en semi-supervisé est fréquent de nos jours car on l'utilise pour sélectionner les exemples les plus instructifs et informatifs. On verra au chapitre suivant son apport en contexte multi-label semi-supervisé, généralement on devrait s'attendre sur des datasets tirés du référentiel de l'apprentissage machine UCI [Frank and Asuncion, 2010] et les données textuelles, à obtenir une précision beaucoup plus grande avec moins d'instances d'apprentissage libellées.

L'apprentissage actif permet d'améliorer encore les performances des classifieurs *SVM* en tenant compte des instances non libellées positives et négatives se trouvant à l'intérieur de la marge et permettent d'adopter plusieurs stratégies d'optimisation en ce sens. Comme travail future, au chapitre 4, on se basera sur des stratégies de calcul d'incertitude des échantillons non libellés à libeller en apprentissage actif à base de *SVM* pour produire de meilleurs scores et aussi à base de *SOM-mixte*, combinant instances et labels à la fois, et qui constituera notre principale contribution en ce sens.

Chapitre 4

L'apprentissage semi-supervisé actif multi-label

Sommaire

1	Introduction	70
2	Travaux connexes relatifs à la classification multi-label	72
3	Apprentissage semi-supervisé multi-label	74
	3.1 Classification k NN multi-label (ML- k NN).....	75
	3.2 Classification SOM multi-label.....	76
	3.3 Classification SVM multi-label.....	77
	3.3.1 Les SVM probabilistes.....	79
	3.3.2 L'espace de versions.....	80
4	Apprentissage actif multi-label	82
	4.1 Travaux connexes sur l'apprentissage actif multi-label.....	82
	4.2 Stratégies de sélection d'échantillons avec SVM multi-label.....	85
	4.2.1 Estimation de la réduction de perte.....	86
	4.2.2 Prédiction des étiquettes.....	87
5	Classifieur d'apprentissage actif topologique proposé	91
	5.1 Formulations mathématiques.....	94
	5.2 Stratégies de sélection d'échantillons avec <i>Act-SOM</i> multi-label.....	96
6	Conclusion	98

Chapitre 4 : L'apprentissage semi-supervisé actif multi-label

Résumé :

L'apprentissage actif et l'apprentissage semi-supervisé font face au même problème, c'est-à-dire la rareté des données libellées ainsi que la difficulté de les obtenir. Il est tout à fait naturel de combiner les deux types d'apprentissage en mode semi-supervisé pour aborder la problématique des deux côtés.

Dans de nombreuses applications, les données sont non étiquetées ou l'étiquetage est coûteux ou peu pratique. Ce fait est encore plus difficile dans l'apprentissage multi-label. La clé de l'apprentissage actif est donc la stratégie de sélection d'échantillons dont le but est de choisir l'élément le plus informatif pour obtenir la meilleure performance de classification. L'apprenant actif sera pour nous entièrement automatique à travers les classifieurs de base SVM, ainsi une comparaison sera faite par la suite avec les TSVM transductives. Ainsi, l'importance des SVM déjà soulignée en mode multi-label au chapitre 2 se retrouve encore renforcée aussi en mode semi-supervisé où elle constitue le classifieur de base approprié.

L'apprentissage actif a été initialement développé pour soutenir l'apprentissage semi-supervisé en mono-label, il est devenu la technique actuelle qui offre le meilleur compromis entre la robustesse aux échantillons biaisés (médiocres) de l'apprentissage initial, la complexité, l'exactitude de la classification et le choix du nombre de nouveaux échantillons à libeller nécessaires pour atteindre la convergence.

Rien d'étonnant puisque dès 2002, certains auteurs et pas des moindres dont Muslea, ont donné comme titre sous forme d'équation, à leur article de base : Actif + apprentissage semi-supervisé = apprentissage multi-vision robuste [Muslea et al., 2002].

Malgré l'importance du problème, la recherche actuelle sur l'apprentissage actif pour la classification multi-label reste à l'état préliminaire.

1. Introduction

Dans de nombreuses applications, de plus en plus les efforts sont axés sur des études semi-supervisées (utilisant de grandes quantités de données non libellées pour augmenter les données libellées limitées) et sur l'apprentissage actif (l'algorithme demande itérativement des exemples d'étiquetage soigneusement choisis dans le but de minimiser l'effort d'étiquetage).

Les méthodes semi-supervisées profitent de l'information fournie par les cas non libellés qui surpassent l'apprentissage supervisé lorsque le nombre de données d'apprentissage est relativement faible et que le nombre de classes est important.

La classification multi-label, où chaque instance est attribuée à plusieurs catégories, est un problème dans l'analyse des données. Cependant, les annotations des instances multi-label sont généralement plus gourmandes en temps ou chères à obtenir que les annotations des cas mono-label. Bien que l'apprentissage actif a été largement étudié sur la réduction des efforts d'étiquetage pour les problèmes liés à un seul label, celui multi-label reste dans un état

préliminaire. Dans ce chapitre, nous proposons d'abord deux stratégies d'apprentissage actif multi-label, une stratégie d'incertitude de prédiction de la marge maximal et une stratégie d'incohérence de cardinalité d'étiquette, puis les intégrer dans un cadre adaptatif d'apprentissage. Nos résultats empiriques sur des datasets multi-label démontrent l'efficacité des stratégies de sélection active d'instance proposées et de l'approche d'apprentissage actif intégré.

Les problèmes classiques de classification multi-classes supposent que chaque instance est associée à un seul libellé d'un ensemble de catégories Y , où $|Y| > 2$. La classification multi-label généralise la classification multi-classes en permettant à chaque instance d'être associée à de multiples étiquettes de Y . Dans beaucoup de problèmes d'analyse de données du monde réel, les objets de données peuvent être assignés à plusieurs catégories et donc produire une classification multi-label.

De nombreuses approches ont été développées dans la littérature pour les problèmes de classification multi-label. Une solution standard et simple pour la classification multi-label est néanmoins pour généraliser le schéma «un-contre-tous» de classification multi-classes. C'est-à-dire que l'on décompose le problème en un ensemble de problèmes de classification binaire, un pour chaque classe, et on résout le problème de classification multi-label en effectuant des classifications binaires standard [Boutell et al., 2004; Joachims, 1998; Lewis et al., 2005]. Quelle que soit la méthode utilisée, l'apprentissage multi-label nécessite en général une quantité suffisante de données libellées pour récupérer des modèles de classification de haute qualité.

Cependant, le processus d'étiquetage lié aux problèmes multi-label est beaucoup plus coûteux et prend beaucoup plus de temps que celui en mono-label. Dans le cas d'une seule étiquette, un annotateur humain doit seulement identifier une seule catégorie pour compléter une étiquette d'instance, alors que dans le cas multi-label, l'annotateur doit considérer chaque étiquette possible pour chaque instance, même si les étiquettes positives sont éparpillées.

L'apprentissage actif, qui tend à un étiquetage sélectif de l'instance et la réduction de l'effort d'étiquetage pour l'entraînement de bons modèles de prédiction, est particulièrement important pour la classification multi-label.

La majorité des études d'apprentissage actif dans la littérature se concentre sur les problèmes de classification mono-label, en particulier les problèmes de classification binaire [Settles, 2012]. Les stratégies d'apprentissage actif mises au point pour les classifications mono-label ne sont cependant pas directement applicables au cas multi-label.

Ce chapitre est organisé comme suit. Le paragraphe 2 présente les travaux connexes relatifs à la classification multi-label. Le paragraphe 3 synthétise les différentes approches vues au chapitre 3 et qu'on a adoptées pour le problème d'apprentissage multi-label en mode supervisé. Le paragraphe 4 traite du problème d'apprentissage actif multi-label où l'apprenant actif est basé sur *SVM*, y compris l'optimisation de la perte et la stratégie de sélection des échantillons. Au chapitre 5, on montre les résultats expérimentaux des algorithmes sur plusieurs datasets du monde réel par rapport à d'autres méthodes connues dans la littérature.

2. Travaux connexes relatifs à la classification multi-label

La plupart des approches sur l'apprentissage multi-label semi-supervisé travaillent dans le cadre transductif. N'empêche qu'une classification multi-label (*MLC*) dans le cadre inductif avec des données non libellées (*iMLCU*) [Wu et al., 2013] a été formulée comme un problème d'optimisation de l'apprentissage de q modèles linéaires, qui s'adapte à des données libellées en exploitant les corrélations de labels pairwise et utilise des données non libellées via des régularisations appropriées pour l'optimisation résultante sur les classifieurs binaires *SVM* [Chapelle et al., 2008].

La clé de l'apprentissage actif est la stratégie de sélection d'échantillons dont le but est de choisir l'instance la plus informative pour obtenir la meilleure performance de classification. *BinMin* [Brinker, 2006] sélectionne des exemples non libellés en fonction de l'étiquette la plus incertaine et l'OVA (one-against-all) a été utilisé pour l'apprentissage multi-label avec *SVM* comme classifieur de base. Cette méthode ne tire pas d'avantage de l'information multi-label.

Dans le domaine de la classification des images, il a été proposé la stratégie moyenne de perte maximale *MML* (Mean Max Loss) [Li et al., 2004] qui sélectionne l'occurrence non libellée qui a la valeur de perte moyenne maximale par rapport aux classes prédites. Une *SVM* a été entraînée pour chaque label et une méthode de seuil limitatif a permis de déterminer les labels pertinents. La valeur de perte globale a été moyennée sur les labels. Étant donné que cette stratégie sélectionne seulement par moyennage, elle ne tire pas d'avantage des corrélations d'étiquettes pour réduire le coût d'étiquetage humain. En outre, lorsqu'un échantillon est sélectionné, tous ses labels devraient être étiquetés. Par la suite, la méthode a été améliorée avec un apprentissage actif bidimensionnel tenant compte des relations entre les échantillons et entre les labels.

Des paires d'échantillons-labels ont été choisies afin de minimiser la limite d'erreur bayésienne multi-label. Cela a permis l'annotation d'un sous-ensemble de labels et l'inférence du reste des labels a été effectuée à partir de corrélations de labels avec *EM* (Expectation-Maximisation). En tirant parti de l'apprentissage actif et de l'apprentissage multi-vue, l'effort d'annotation a été réduit par rapport à la sélection aléatoire et à *MML* (Mean Max Loss). Comme l'apprentissage multi-vue et l'apprentissage actif peuvent être efficacement intégrés, cela peut être une ligne à explorer dans le paramétrage de l'apprentissage multi-label.

C'est ce qu'on compte aborder dans ce chapitre, en appliquant un apprentissage actif en conjonction avec un apprentissage semi-supervisé. En d'autres termes, si nous devons étiqueter quelques exemples d'apprentissage semi-supervisé, il peut être intéressant de laisser l'algorithme d'apprentissage nous dire quelles instances devraient être libellées, plutôt que de les sélectionner au hasard. Nous limiterons la portée de la sélection des requêtes à l'ensemble de données non étiquetées, une pratique connue sous le nom d'apprentissage actif à base de groupe (pool) ou d'échantillonnage sélectif, déjà vue au paragraphe 5 du chapitre 3.

Une approche pour la classification des textes, appelée *MMC* (Maximum loss reduction with Maximum Confidence), est proposée dans [Yang et al., 2009]. Une *SVM* a été entraînée

pour chaque label et la perte globale du classifieur a été mesurée en recueillant la perte de tous les classifieurs binaires. Au lieu d'estimer les labels pour chaque instance, le nombre de labels a été estimé en appliquant une régression logistique. Les caractéristiques d'apprentissage du modèle de régression logistique sont les probabilités obtenues par les classifieurs binaires et le nombre de labels est la cible théorique à prédire. *MMC* a surclassé la sélection aléatoire, *MML* (Mean Max Loss) et *BinMin* dans le domaine de la classification des textes tout en réduisant significativement le coût d'étiquetage.

Enfin, dans [Esuli and Sebastiani, 2009], plusieurs stratégies ont été proposées pour réaliser un étiquetage global dans la classification des textes, dans lequel un classement unique des instances non libellés combine les sorties de q classifieurs binaires individuels en appliquant la *Rank-SVM* (voir paragraphe 4.5 du chapitre 2) qui est devenue une méthode de classement de référence en apprentissage multi-label.

L'étiquetage des données textuelles prend beaucoup de temps mais est essentiel pour la classification automatique des textes. En particulier, créer manuellement plusieurs labels pour chaque document peut devenir impraticable lorsqu'une très grande quantité de données est nécessaire pour l'entraînement de classifieurs de texte multi-label. Pour minimiser l'étiquetage humain, on peut appliquer une approche proposée dans [Yan et al., 2010] d'apprentissage actif multi-label qui peut réduire les données libellées requises sans sacrifier la précision de classification.

Traditionnellement les algorithmes d'apprentissage actifs ne peuvent traiter que les problèmes à un seul label, c'est-à-dire que chaque donnée est limitée à avoir une seule étiquette. L'approche adoptée prend en compte les informations multi-label, et tend à sélectionner les données non libellées qui peuvent conduire à la plus grande réduction de la perte attendue du classifieur. Spécifiquement, la perte du modèle est approximée par la taille de l'espace de versions, et le taux de réduction de la taille de l'espace de versions est optimisé avec les machines à vaste marge (*SVM*).

Une méthode efficace est appliquée de prédiction des labels, conçue pour prédire les labels possibles pour chaque instance de données libellées, et la perte attendue est approximée en additionnant les pertes sur tous les labels selon le résultat le plus confiant de la prédiction des labels.

Les expériences sur plusieurs ensembles de données du monde réel (tous sont accessibles au public et pas seulement textuelles) démontrent que l'approche peut obtenir un résultat de classification prometteur avec beaucoup moins de données libellées que les méthodes de référence issues de la littérature.

Dans la littérature, les algorithmes d'apprentissage supervisé sont utilisés majoritairement dans la classification des textes. Elle nécessite une quantité suffisante de labels pour l'apprentissage d'un modèle de haute qualité. Cependant, l'étiquetage est généralement un processus long et coûteux accompli par des experts du domaine. L'apprentissage actif est une approche pour réduire ce coût d'étiquetage. L'apprenant actif sélectionne itérativement un

échantillon de données à étiqueter sur la base de certaines stratégies de sélection suggérant les données qui méritent le plus d'être étiquetées.

Ainsi, il peut atteindre des performances comparables à celles des apprenants supervisés cependant en utilisant beaucoup moins de données libellées. L'apprentissage actif est particulièrement important pour la tâche de classification de texte multi-label. La raison en est que, dans le cas d'un seul label, un juge humain peut cesser de labelliser une instance une fois sa catégorie est identifiée. Mais dans le cas multi-label, le juge a besoin de décider pour chaque instance une fois toutes les catégories possibles visitées. Ainsi, l'affectation de labels pour des données multi-label est beaucoup plus complexe que pour les données à un seul label.

Malgré la valeur et la signification de ce problème, la recherche est très limitée sur l'apprentissage actif multi-label. La plupart des recherches sur l'apprentissage actif se focalisent sur le problème de classification [Lewis et al., 2004; Luo et al., 2005; Tong and Koller, 2002; Yan et al., 2003]. La stratégie de sélection de l'échantillon suit strictement l'hypothèse que chaque instance a un seul libellé et ne peut donc pas être directement appliquée dans l'apprentissage actif multi-label.

La raison peut être expliquée dans la tâche de classification multi-label où la technique populaire de un-contre-tous [Cortes and Vapnik, 1995] est utilisée et les probabilités de classification sur toutes les classes possibles sont données.

Cependant, si nous supposons que pour chaque instance, on tire seulement un label et en prenant la stratégie la plus incertaine, une instance serait considérée comme plus difficile à classer si le score de probabilité sur son label prédit peut être inférieur à celui d'une autre instance renfermant le même label. C'est pourquoi il est très important de tenir compte de l'information multi-label dans la stratégie de sélection de l'échantillon.

Dans ce travail, nous proposons l'application d'une approche d'apprentissage actif multi-label pour la classification dans différents domaines (biologie, médecine, texte, multimédia). La stratégie de sélection de l'échantillon vise à étiqueter les données qui peuvent aider à maximiser le taux de réduction de la perte escomptée du modèle. Pour mesurer la réduction des pertes, nous utilisons les *SVM* en termes d'espace de versions [Tong and Koller, 2002] en raison de leur efficacité dans l'apprentissage actif sur la classification des textes.

3. Apprentissage semi-supervisé multi-label

Deux classifieurs principaux sont généralement utilisés comme classifieurs de base pour la conduite des résultats expérimentaux et éventuellement saisir l'impact des différentes mesures d'évaluation sur les datasets, les *SVM* et les *kNN*, côté instances (exemples) pour la classification ou côté labels (étiquettes) pour le classement (ranking).

Les *SVM* ont été traditionnellement appliquées pour résoudre des problèmes de classification binaire. Comme d'autres techniques, elles ont évolué avec le temps, pour faire face à d'autres types de tâches telles que la classification multi-classe et multi-label. Dans

notre travail, plusieurs des méthodes de classification multi-label basées sur les SVM et ses variantes seront adoptées. Au fait, les SVM, en mode semi-supervisé multi-label seront prises comme nos classifieurs de base pour aussi appliquer les BR-SVM (Binary Relevance-SVM) et surtout pour l'apprentissage actif par pool ou groupe.

Puisque les SVMs ont tendance à avoir un bon comportement lors du traitement des données (images, textuelles), on a adopté donc ce type de modèle comme classifieur sous-jacent. L'équivalent de la méthode côté classement (arrangement), la *Rank-SVM* [Elisseef and Winston, 2001] est une bonne option car les classifieurs binaires de base ne tiennent pas compte des corrélations entre les labels, puisque l'indépendance totale entre eux est supposée par la plupart des méthodes basées binaires. Pour alléger ce problème, la *Rank-SVM* est une approche directe basée sur les principes SVM, et reposant sur une nouvelle métrique à minimiser, à savoir l'approximation linéaire de la perte de Hamming (voir section 4.4 du chapitre 2). Une fois que le modèle basé SVM a été entraîné pour produire le classement des labels, une fonction spécifique permet de régler le seuil de coupe à partir duquel est extrait le jeu d'étiquettes comme un sous-ensemble du classement complet.

Puisque la *Rank-SVM* tient compte des corrélations d'étiquettes, elle devrait donc théoriquement être plus performante que les modèles binaires purs, les tests expérimentaux confirment son comportement et globalement sa supériorité [Elisseef and Winston, 2001].

3.1 Classification kNN multi-label (ML-kNN)

L'algorithme ML-kNN est une des approches les plus simples de la classification, basée sur celui du kNN (voir la section 4.1 du chapitre 2). Une fois qu'un nouvel échantillon d'une donnée est considéré, un classifieur kNN recherche ses k -voisins les plus proches. Pour ce faire, la distance (dans un certain d -dimensionnel espace) entre les caractéristiques du nouvel échantillon et celles des instances du dataset est calculée. Une fois que les instances les plus proches ont été rassemblées, leurs classes sont utilisées pour prédire celle du nouvel échantillon.

Puisque kNN n'engendre pas de modèle, seulement quand un nouvel échantillon arrive le classifieur fait un certain travail, il est habituellement connu comme une méthode paresseuse [Aha, 1997]. Il est également souvent référencé comme apprentissage basé-instance [Aha et al., 1991].

ML-kNN [Zhang and Zhou, 2007] est une adaptation de la méthode kNN au scénario multi-label. Contrairement à l'algorithme kNN classique, ML-kNN n'est pas si paresseux. Il commence par construire un modèle limité qui se compose de deux éléments d'information:

- Les probabilités a priori pour chaque label. Ce sont simplement le nombre de fois où le label apparaît dans le dataset multi-label, divisé par le nombre total d'instances. Un facteur de lissage est appliqué pour éviter de multiplier par zéro.
- Les probabilités conditionnelles pour chaque label, calculées comme la proportion des instances avec le label considéré dont les k -voisins les plus proches, ont également le même label.

Ces probabilités sont calculées indépendamment pour chaque label, comme une collection de problèmes binaires individuels. Par conséquent, les dépendances potentielles parmi les labels sont entièrement ignorées par cet algorithme. Après ce processus limité d'apprentissage, le classifieur est capable de prédire de nouvelles instances. Lorsqu'un nouvel échantillon arrive, il passe par les étapes suivantes:

- Premièrement, on obtient les k -voisins les plus proches de l'échantillon donné. Par défaut, la norme L^2 (distance euclidienne) est utilisée pour mesurer la similitude entre l'instance de référence et les échantillons dans le dataset multi-label.
- Ensuite, la présence de chaque label dans les voisins est utilisée comme preuve pour calculer les probabilités maximales a posteriori (*MAP*) à partir de celles conditionnelles obtenues avant.
- Enfin, le labelset (ensemble des labels) du nouvel échantillon est généré à partir des probabilités *MAP*. La probabilité elle-même est fournie comme un niveau de confiance pour chaque label, ce qui rend possible de générer également un classement des labels.

L'implémentation MATLAB de référence pour l'algorithme ML- k NN est fournie par [Zhang and Zhou, 2007] à leur propre site Web.

<http://cse.edu.cn/people/zhangml/Ressources.htm#codes>.

Afin de porter un jugement sur nos résultats mentionnés au chapitre suivant au niveau de la configuration expérimentale et résultats sur les six datasets de différents domaines d'application, on fera souvent référence à cet algorithme en comparaison avec ceux qu'on a adoptés et il constitue de ce fait un algorithme de base de référence.

3.2 Classification SOM multi-label

L'apprentissage semi-supervisé vise à découvrir des structures spatiales dans des espaces d'entrée de grande dimension lorsque les informations disponibles de base sur les clusters sont insuffisantes. Il s'avère que la méthode populaire de propagation d'étiquettes de [Zhu, 2007] peut être considérée comme une modification de la technique d'apprentissage par lots bien connue de *SOM*. Ainsi, les cartes topologiques *SOM* sont de plus en plus utilisées comme approche pour l'apprentissage semi-supervisé et offrant en outre, une simple mais puissante méthode d'estimation des paramètres essentiels pour l'algorithme de clustering résultant.

Par conséquent, *SOM* est devenue récemment un centre d'intérêt, en particulier à cause de l'apprentissage semi-supervisé qui vise à incorporer une petite quantité de données pré-classifiées dans des méthodes d'apprentissage non supervisées afin d'améliorer les performances de l'analyse des données. Les concepts de base des cartes auto-organisatrices et les algorithmes associés ont été déjà introduits dans le paragraphe 3.2 du chapitre 3.

Plusieurs travaux dans la littérature font appel aux cartes *SOM* et les utilisent comme entrée aux classifieurs de base pour résoudre le problème de la large dimensionnalité des vecteurs d'espace d'entrée [Abaei et al., 2015] pour la détection de défauts dans les logiciels; [Azcarraga et al., 2005] pour la prévision des ventes dans le domaine des marchés d'automobile; [Patra and Bruzzone, 2012-2014] pour une nouvelle technique d'apprentissage actif basée sur *SOM* suivi de *SVM*; et aussi d'autres auteurs dans le cadre de la classification

de textes en les utilisant à différents niveaux de co-occurrences de mots dans les documents ou catégories, sous forme d'architecture hiérarchique *SOM* à trois niveaux [Luo and Heywood, 2005].

On utilise généralement, la capacité de la carte *SOM* pour un système d'apprentissage non supervisé pour fournir des approximations d'une entrée à dimension élevée dans un espace dimensionnel inférieur. La *SOM* agit comme un encodeur pour représenter un grand espace d'entrée en trouvant un plus petit ensemble de prototypes.

Ainsi, dans notre travail, on fait appel à la carte *SOM*, sous différents aspects, d'abord en tant que modèle de clustering pour les instances d'entrée, ensuite comme modèle d'étiquetage (*SOM-Y*) et surtout comme modèle d'analyse des données mixtes *SOM-mixte* où l'apprentissage est combiné à un mécanisme de pondération des instances et labels associés sous forme de poids d'influence sur la pertinence de ces deux attributs.

L'apprentissage des prototypes est réalisé d'une manière simultanée en favorisant une classification optimisée des données. L'approche mentionnée s'inspire de celle proposée dans le cadre non-supervisé mono-label par les auteurs N. Rogovschi, M. Lebbah, N. Grozavu et Y. Bennani [Rogovschi et al., 2008-2011] de LIPN-Paris 13 et de LIPADE-Paris Descartes, et qui a été validée sur des données qualitatives codées en binaire et plusieurs bases de données mixtes.

Bien qu'à l'état préliminaire, mais tenant compte de la qualité intrinsèque des données, cette approche issue des méthodes géométriques bio-inspirées dont on a pu apprécier les avantages, par visualisation des référents ainsi que les variables qui les caractérisent par cellule de la carte, durant les séjours de recherche à LIPN, semble prometteuse car on peut l'utiliser pour des données symboliques mixtes et pouvoir accélérer le processus d'apprentissage [Hajjar, 2015].

Des recherches sont en cours tenant compte de l'exploration des corrélations entre les instances à l'entrée, les labels à la sortie, ou carrément les deux à la fois, en axant sur des fonctions d'inférence et/ou des méthodes d'extraction de caractéristiques pour la réduction de l'espace d'entrée, ou celui de sortie ou sous forme de représentation multi-instances multi-label (*MIML*) avec comme défi, à l'avenir, la récupération des labels les plus pertinents dans une phase d'apprentissage [Zhou et al., 2012].

Il convient de souligner qu'on tourne de plus en plus vers des problèmes de grande envergure où le nombre de labels est extrêmement important d'où la nécessité de méthodes de cartographie telles, les cartes topologiques compétitives *SOM*, pour la réduction des espaces entrée-sortie (instances-labels) qui se traduit par la réduction de complexités informatique et spatiale [Gibaja and Ventura, 2014].

3.3 Classification SVM multi-label

Transformer un problème de classification multi-label en un ensemble de problèmes de classification binaires indépendants via le schéma (one-vs-all) est un concept simple et

efficace en calcul pour la classification multi-label. Dans ce travail, nous effectuons un apprentissage multi-label dans le cadre d'un tel mécanisme en utilisant des machines SVM standard pour les problèmes de classification binaire associée à chaque classe.

Au chapitre 2 relatif à l'apprentissage multi-label, dans le cadre des approches d'apprentissage d'adaptation, au paragraphe 4.4, les travaux connexes aux SVM multi-label ont été présentés; dans ce présent paragraphe on s'intéresse aux approches qu'on a adoptées y compris en mode semi-supervisé, à savoir la *semi-SVM* ou *BR-SVM* (binary relevance SVM).

Au chapitre 3 relatif à l'apprentissage semi-supervisé, dans des classifieurs de base les plus utilisés, au paragraphe 4.1, les travaux connexes ainsi que la formalisation des SVM inductives standards (S3VM) et celle des SVM transductives (TSVM) ont été présentées (voir respectivement les équations 3.16-17-19).

Afin de mieux saisir la prise en charge de l'aspect formalisation multi-label par les SVM, on reprend dans un souci de clarté, le principe pour saisir la prédiction de l'étiquette à travers sa valeur de confiance.

La classification multi-label est la tâche de classer les instances dans un sous-ensemble de classes prédéfinies. Étant donné un ensemble d'apprentissage multi-label libellé $L = \{(x_i, y_i)\}_{i=1}^N$, où x_i est le vecteur de caractéristiques d'entrée pour la $i^{\text{ème}}$ instance, et son vecteur d'étiquette y_i est un vecteur évalué $\{+1, -1\}$ avec une longueur k telle que $k = |Y|$. Donc, les exemples d'entraînement sont les instances libellées $x_1, \dots, x_N$ et les k classes comme 1, ..., k . Nous représentons le jeu d'étiquettes de x_i par un vecteur binaire $y_i = [y_i^1, \dots, y_i^k]$, $y_i^j \in \{+1, -1\}$, où $y_i^j = 1$ si x_i appartient à la classe j , sinon $y_i^j = -1$. On peut dénoter l'ensemble de toutes les combinaisons de classes possibles comme $Y = \{+1, -1\}^k$. Pour la classe $j^{\text{ème}}$ ($j = 1, \dots, k$), l'entraînement du SVM binaire est un problème standard d'optimisation quadratique :

$$\min_{w_j, b_j, \{\xi_i^j\}} \frac{1}{2} \|w_j\|^2 + C \sum_{i=1}^N \xi_i^j \quad (4.1)$$

$$\text{Sous réserve de : } y_i^j (w_j^T x_i + b_j) \geq 1 - \xi_i^j, \xi_i^j \geq 0, \forall_i$$

où $\{\xi_i^j\}$ sont les variables molles (ressorts) et C le paramètre d'ajustement qui maximise la marge de séparation de classe. Le classifieur multi-label peut être exprimé comme une fonction de décision $f: X \rightarrow Y$. Dans notre étude d'apprentissage actif, nous considérons SVM comme le classifieur multi-label de base, puisque SVM a démontré un succès significatif sur les tâches de classification de texte [Joachims, 2002; Yang, 2001]. Habituellement, le SVM multi-label adopte l'approche un-contre-tous, qui entraîne un classifieur binaire séparé pour chaque classe possible contre le reste des classes, et combine la sortie de tous les classifieurs binaires pour déterminer les étiquettes finales des instances données. Dans la classification binaire, SVM essaie de trouver l'hyperplan qui peut séparer les données d'entraînement par une marge maximale.

Selon l'équation (4.1), les paramètres du modèle w_j et b_j issus de ce problème d'apprentissage binaire définissent un classifieur binaire associé au $j^{\text{ième}}$ classe: $f_j(x_i) = w_j^T x_i + b_j$.

L'ensemble des classifieurs binaires de toutes les classes peuvent être utilisés indépendamment pour prédire le vecteur étiquette $\hat{y}$ pour une instance non libellée $\hat{x}$. La $j^{\text{ième}}$ composante du vecteur étiquette $\hat{y}_j$ a la valeur 1 si $f_j(\hat{x}) > 0$, et a une valeur -1 autrement. La valeur absolue $|f_j(\hat{x})|$ peut être considérée comme une valeur de confiance pour sa prédiction $\hat{y}_j$ sur l'instance $\hat{x}$.

Quelques notions sont à rappeler et dont on a besoin par la suite au niveau des programmes, avant d'entamer une étude plus approfondie sur les SVM choisies comme classifieur de base, principalement le noyau ou kernel et ses paramètres de réglage σ et C de régularisation, déjà mentionnés à la section 4.1 du chapitre 3, les SVM probabilistes avec leurs sorties probabilistes, ainsi que la transformation dans l'espace des versions à la recherche des instances les plus informatives à libeller.

3.3.1 Les SVM probabilistes

Les SVM reposent principalement sur la fonction de décision $(\langle w, x_i \rangle + b)$ qui indique l'appartenance d'une donnée à une classe. Son résultat n'est malheureusement pas borné et ne représente donc pas une valeur probabiliste.

Les SVM probabilistes sont une famille d'algorithmes qui permettent de fournir une probabilité d'appartenance d'une observation x_i à une classe. Habituellement, la classification d'un exemple inconnu est faite en fonction de la sortie maximale entre toutes les SVM. L'approche la plus intuitive pour estimer la probabilité a posteriori avec la technique OVA (Un-Contre-Tous) consiste à réinterpréter séparément les résultats de chaque classifieur SVM en terme de probabilités en utilisant la méthode proposée par [Platt, 1999], qui utilise une sigmoïde supplémentaire. L'information qu'apportent donc les SVM, représente une probabilité $P(y = 1|x)$ et elle permet d'effectuer des traitements avec une transition douce. Cette probabilité est définie de la manière suivante :

$$P(y = 1|x) = \begin{cases} 1 & \text{si } \langle w, x_i \rangle + b \geq 1 \\ 0 & \text{si } \langle w, x_i \rangle + b < 1 \\ p_x \in]0,1[& \text{sinon} \end{cases} \quad (4.2)$$

Pour obtenir une sortie probabiliste des SVM, la solution utilisée est obtenue par la bijection par sigmoïde [Platt, 1999].

$$P(y = 1|x) = \frac{1}{1 + \exp(Af(x) + B)} \quad (4.3)$$

Les constantes A et B (paramètres de la sigmoïde) sont évaluées à partir de la distribution de $f(x)$ sur l'ensemble d'apprentissage, $f(x) = \langle w, x_i \rangle + b$.

A et B sont estimés par la maximisation de la log-vraisemblance (maximum de la vraisemblance). On peut définir un ensemble d'apprentissage D' à partir de l'ensemble d'apprentissage initial D , $D' = \{(x_1, t_1), \dots, (x_N, t_N)\}$, où t_i étant la probabilité d'appartenance d'une donnée x_i à une classe :

$$\begin{cases} t_i = 1 - \epsilon & \text{pour } y_i = 1 \\ t_i = \epsilon & \text{pour } y_i = -1 \end{cases} \quad \text{où } \epsilon \in [0,1] \quad (4.4)$$

En utilisant la correction de Laplace, t_i peut être redéfini comme suit :

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{pour } y_i = 1 \\ \frac{1}{N_- + 2} & \text{pour } y_i = -1 \end{cases} \quad (4.5)$$

où N_+ le nombre d'instances positives et N_- le nombre d'instances négatives dans D' .

L'estimation des paramètres A et B revient à maximiser la probabilité de bon classement sur l'ensemble d'apprentissage D' :

$$\max_{A,B} \prod_{(x_i, t_i)} P(y = 1|x_i)^{t_i} (1 - P(y = 1|x_i))^{(1-t_i)} \quad (4.6)$$

En remplaçant la probabilité $P(y = 1|x_i)$ par p_i , et en introduisant le logarithme, le problème (4.6) s'écrit :

$$\min_{A,B} - \sum_i (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \quad (4.7)$$

où p_i représente la sortie de la sigmoïde et t_i est la probabilité ciblée.

Le problème d'optimisation (4.7) peut être résolu efficacement en utilisant des algorithmes d'optimisation [Lin et al., 2003]. L'apprentissage des SVM pourra être effectué sur l'ensemble des données d'apprentissage après l'estimation des paramètres A et B . Il est préférable d'utiliser un sous-ensemble de D pour l'estimation des paramètres A et B afin de diminuer le risque de sur-apprentissage.

3.3.2 L'espace de versions

Les SVM ont rencontré un succès significatif dans de nombreuses tâches de systèmes d'apprentissage du monde réel. Cependant, comme la plupart des algorithmes d'apprentissage automatique, elles sont généralement appliquées en utilisant un ensemble d'entraînement sélectionné au hasard et classé à l'avance. Dans de nombreux contextes, on a l'option de l'apprentissage actif basé groupe ou pool. Au lieu d'utiliser un ensemble d'apprentissage sélectionné aléatoirement, l'apprenant a accès à un pool d'instances non libellées et peut demander les labels pour un certain nombre d'entre-elles.

L'apprentissage actif en pool pour la classification a été introduit par [Lewis and Gale, 1994]. L'apprenant a accès à un pool de données non libellées et peut demander le vrai label

de la classe pour un certain nombre d'instances dans le pool. Dans de nombreux domaines, il s'agit d'une approche raisonnable car une grande quantité de données non libellées est facilement disponible. Le problème principal de l'apprentissage actif est de trouver un moyen de choisir les bonnes requêtes à partir du pool.

Afin de mieux saisir par la suite, la stratégie de sélection d'échantillon avec SVM, notamment la réduction de la perte estimée, et tout en se basant sur le travail de [Tong and Koller, 2002], on peut utiliser la marge SVM comme mesure de la taille de l'espace de versions (voir équation 4.10).

La notion d'espace de versions permet de fournir une motivation théorique pour la méthode de l'apprentissage actif permettant de procéder à un choix significatif pour réduire considérablement le besoin d'instances libellées d'entraînement à la fois au niveau du paramétrage inductif et transductif.

Étant donné un dataset d'entraînement libellé et un noyau K de Mercer, il existe un ensemble d'hyperplans qui séparent les données dans l'espace de caractéristiques induit $\mathcal{F}$ (voir Fig.4.1). Cet ensemble d'hypothèses cohérentes est appelé l'espace de versions [Mitchell, 1982]. En d'autres termes, l'hypothèse f est dans l'espace de versions si pour chaque instance d'entraînement $\mathbf{x}_i$ avec l'étiquette y_i , on a $f(\mathbf{x}_i) > 0$ si $y_i = 1$ et $f(\mathbf{x}_i) < 0$ si $y_i = -1$. Plus formellement, l'ensemble d'hypothèses possibles est donné comme suit:

$$\mathcal{H} = \left\{ f \left| f(x) = \frac{w \cdot \Phi(x)}{\|w\|} \text{ où } w \in W \right. \right\} \quad (4.8)$$

où l'espace de paramètre W est simplement égal à $\mathcal{F}$. L'espace de version, V est alors défini comme:

$$V = \{f \in \mathcal{H} \mid \forall i \in \{1, \dots, n\}, y_i f(\mathbf{x}_i) > 0\} \quad (4.9)$$

Notons que puisque $\mathcal{H}$ est un ensemble d'hyperplans, il existe une bijection entre les vecteurs unitaires $\mathbf{w}$ et les hypothèses f dans $\mathcal{H}$. Ainsi V est redéfini comme:

$$V = \{\mathbf{w} \in W \mid \|\mathbf{w}\| = 1, y_i (\mathbf{w} \cdot \Phi(\mathbf{x}_i)) > 0, i = 1, \dots, n\} \quad (4.10)$$

En d'autres termes, un espace de versions V est un dispositif utilisé en apprentissage supervisé pour induire des concepts généraux ou des règles à partir d'un ensemble mêlant des exemples vérifiant la règle qu'on cherche à établir et des contre-exemples ne la vérifiant pas.

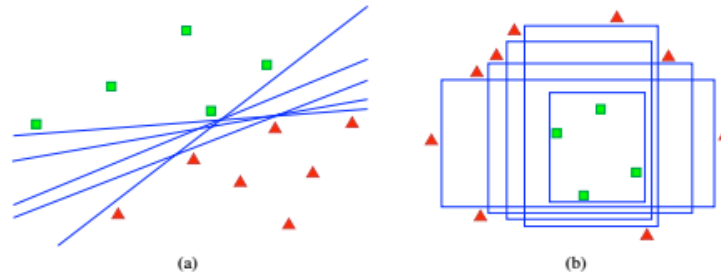


Figure 4.1 – Exemples d'espace de versions pour (a) des classifieurs linéaires et (b) des classifieurs carrés à axes parallèles. Toutes les hypothèses sont cohérentes avec les données d'apprentissage libellées dans L (comme indiqué par polygones ombrés), mais chacune représente un modèle différent dans l'espace de versions [Settles, 2012].

La prémisse fondamentale est de minimiser l'espace de versions V , qui est l'ensemble des hypothèses $\mathcal{H}$ qui sont en cohérence avec les données d'apprentissage libellées actuelles dans L . La figure 4.1 illustre le concept d'espaces de version pour (a) des fonctions linéaires et (b) des classifieurs à axes parallèles dans différentes tâches de classification binaires.

La représentation des hypothèses de rectangles à partir d'exemples positifs (les carrés, qui doivent être à l'intérieur du plus petit rectangle) et d'exemples négatifs (les triangles, qui doivent être à l'extérieur des autres rectangles). Le plus petit rectangle est l'hypothèse la plus spécifique (en spécialisant plus, on ne couvrirait plus certains exemples positifs) et les autres rectangles représentent les hypothèses les plus générales (en généralisant plus, on couvrirait des exemples négatifs), et l'espace entre les rectangles représente d'autres hypothèses de l'espace de versions.

Si nous considérons l'apprentissage automatique comme une recherche pour le «meilleur» modèle dans l'espace de versions, alors notre but dans l'apprentissage actif est de limiter la taille de cet espace autant que possible (de sorte que la recherche devient plus précise) avec le moins d'exemples libellés possibles.

L'idée d'espace des versions, comme ensemble de toutes les hypothèses cohérentes avec les données d'apprentissage, autorise à étudier des algorithmes d'apprentissage nouveaux et il est remarquable de noter que la vision de l'apprentissage comme sélection de bonnes hypothèses au sein d'un ensemble d'hypothèses possibles données a priori s'accorde avec la vision actuelle de l'approche de l'apprentissage artificiel, considérant l'apprentissage comme la sélection des hypothèses les plus performantes par rapport aux observations [Cornuéjols, 2009].

4. Apprentissage actif multi-label

Le but de l'apprentissage actif est de réduire l'effort d'étiquetage et le coût nécessaires à l'entraînement d'un modèle de prévision de haute qualité. Compte tenu d'un pool (groupe) important d'instances non libellées, un apprenant actif sélectionne itérativement la plupart des instances informatives du pool pour interroger par exemple, un annotateur humain (oracle) pour les étiquettes. La plupart des études d'apprentissage actif dans la littérature se sont concentrées sur des problèmes de classification mono-label.

4.1 Travaux connexes sur l'apprentissage actif multi-label

Le principal défi de l'apprentissage actif multi-label est de développer des stratégies efficaces pour évaluer l'informativité unifiée d'une instance non libellée dans toutes les classes. Des travaux d'apprentissage actif multi-label existent, tels que [Brinker, 2006; Li et al., 2004; Esuli et Sebastiani, 2009; Singh et al., 2010; Yang et al., 2009], ils mesurent l'informativité d'une instance non libellée en traitant tous les labels d'une manière

indépendante sans tenir compte de l'information potentielle implicite sur la structure de l'étiquette dans toutes les classes.

L'une des stratégies d'apprentissage actif les plus couramment utilisées est l'échantillonnage d'incertitude, où l'apprenant actif choisit l'instance qui est la plus incertaine à étiqueter pour le modèle de classification entraîné actuel. Bien que les méthodes d'échantillonnage d'incertitude restent myopes sans pouvoir mesurer la valeur informationnelle (informativité) prédictive future de l'instance candidate à partir de grande quantité de données libellées, elles sont calculées efficacement et ont démontré de bonnes performances empiriques [Lewis and Gale, 1994; Luo et al., 2005; Culotta and McCallum, 2005; Settles and Craven, 2008]. Certaines méthodes d'apprentissage actif non myopes plus sophistiquées exploitent des données non libellées pour minimiser une approximation de l'erreur de généralisation [Guo and Greiner, 2007; Guo and Schuurmans, 2007; Roy and McCallum, 2001; Yan et al., 2003; Zhu et al., 2003].

Cependant, ces méthodes sont généralement coûteuses en termes de calcul car elles nécessitent un nouveau modèle de prédiction pour être recyclées pour chaque requête candidate. L'apprentissage actif pour la classification multi-label est toutefois encore à l'état préliminaire. La plupart des méthodes d'apprentissage actif multi-label décomposent la classification multi-label en un ensemble de problèmes de classification binaire et prennent des décisions de sélection d'instance en exploitant les classifieurs binaires indépendamment sans considérer l'information de structure d'étiquette d'une instance révélée dans toutes les classes.

[Brinker, 2006] utilise une simple extension de la stratégie d'échantillonnage d'incertitude en décomposant le problème de classification multi-label en plusieurs classifieurs binaires en utilisant le schéma un-vs-tous et sélectionne l'exemple qui minimise la plus petite marge SVM entre tous les classificateurs binaires.

[Singh et al., 2009] prend simplement la moyenne des scores d'incertitude de tous les classifieurs binaires SVM comme mesure de sélection d'instance. Dans [Li et al., 2004], une méthode d'apprentissage actif SVM a été proposée pour la classification multi-label des images. Elle détermine les labels prédits non libellés en utilisant des classifieurs SVM binaires et prend une décision de sélection d'instance en utilisant des stratégies de perte maximale, Max Loss (*ML*) et de sa moyenne, Mean Max Loss (*MML*) pour compter les pertes de tous les classifieurs binaires. [Yang et al., 2009] présente une stratégie appelée réduction des pertes maximales avec une confiance maximale (Loss reduction with maximal confidence-*MMC*), en utilisant une régression logistique multi-classes pour prédire le nombre de labels pour une instance non libellée, et puis calcule la mesure *MMC* en additionnant les pertes des classifieurs SVM sur tous les labels.

Dans ce travail, on se concentre néanmoins sur le problème de sélection des instances en développant deux stratégies d'échantillonnage d'incertitude multi-label *BinMin* [Brinker, 2006] et *MMC* [Yang et al., 2009] avec une mesure d'erreur de généralisation approximative pour sélectionner efficacement les instances les plus informatives.

Dans le chapitre 5 suivant, ces deux stratégies d'apprentissage actif multi-label, nous permettent l'évaluation de l'incertitude de prédiction sur la marge maximale sous forme de minimisation des pertes maximales sur chaque occurrence non libellée.

Notre étude empirique sur des datasets de classification multi-label tentera de démontrer l'efficacité des stratégies d'apprentissage actif multi-label et éventuellement de l'approche intégrée d'apprentissage actif adaptatif. Evidemment, tout ceci dépendra non seulement de la stratégie d'échantillonnage mais aussi de l'impact de la complexité du dataset considéré en multi-label [Charte et al., 2016]. Cet aspect, sera revu au chapitre 5 suivant, lors de l'interprétation des résultats obtenus.

D'habitude dans les recherches antérieures, la perte est modélisée pour le cas d'un seul label, et ici nous l'étendons au cas multi-label pour différents datasets issus de divers domaines. Nous l'appliquons également pour prédire des labels pour des données multi-label. La perte escomptée est approximée avec la perte associée au résultat le plus confiant de la prédiction du label. Nous montrerons que la méthode de prédiction du label approprié est essentielle pour mesurer la perte pour les données multi-label.

Nous évaluerons, au chapitre 5, empiriquement l'efficacité de la méthode en utilisant plusieurs datasets du monde réel qui sont accessibles au public. Les résultats démontrent que la méthode est concurrente à l'état de l'art des algorithmes d'apprentissage actif pour la classification multi-label de texte ou autres domaines, et peut considérablement réduire la demande de données libellées tout en conservant des résultats prometteurs de classification.

Dans ce chapitre, nous considérons l'apprentissage actif basé pool (ou groupe) qui semble être le scénario le plus répandu pour la recherche appliquée dans l'apprentissage actif. Supposons que nous avons un petit ensemble d'instances multi-label libellées $D_l = \{(x_i, y_i)\}_{i=1}^{N_l}$, mais grand nombre de cas non libellés $D_u = \{(x_i)\}_{i=1}^{N_u}$.

Comme ci-dessus, le vecteur étiquette y_i est un vecteur évalué sur $\{+1, -1\}$ de longueur k . Un apprenant actif sélectionnera itérativement l'instance la plus informative à partir du groupe D_u non libellé à libeller, puis le déplacera vers l'ensemble libellé D_l et recommence de nouveau à recycler le modèle augmenté D_l d'entraînement pour la classification. Nous visons à adopter des stratégies d'apprentissage actif multi-label pour l'apprentissage d'un bon modèle SVM de classification multi-label avec moins d'exemples libellés et donc un moindre coût d'étiquetage.

On rappelle qu'un classifieur binaire est de la forme $f(\mathbf{x}) = \mathbf{w} \times \mathbf{x}$ et que f_i représente le classificateur binaire associé à la classe cible i . Étant donné une instance de test $\mathbf{x}'$, si $f_i(\mathbf{x}') > 0$, alors $\mathbf{x}'$ appartient à la classe i , sinon, les étiquettes de $\mathbf{x}'$ ne seront pas incluses dans la classe i .

Au début, un classifieur est entraîné en utilisant l'ensemble libellé initial D_l , basé sur ce classifieur, l'apprenant sélectionne un échantillon de D_u et requiert ses vraies étiquettes en fonction de certains critères. Ensuite, les données nouvellement libellées sont incorporées dans D_l . L'apprentissage et le processus d'étiquetage fonctionne itérativement après un certain

nombre d'itérations ou lorsque le classifieur atteint une précision suffisante. La question clé de l'apprentissage actif est de savoir comment choisir les exemples les plus informatifs de données à étiqueter, ce qui représente la stratégie de sélection d'échantillon. Le problème de recherche étudié dans ce travail peut être décrit comme suit: afin d'entraîner un classifieur actif multi-label efficace, comment concevoir la stratégie de sélection de l'échantillon pour réduire le coût autant que possible?

On présentera tout d'abord le cadre menant à l'optimisation d'apprentissage actif multi-label, ensuite on décrira la stratégie adoptée de sélection d'échantillons avec SVM.

4.2 Stratégies de sélection d'échantillons avec SVM Multi-label

Dans le cadre de l'apprentissage actif optimal, l'objectif d'optimisation de notre apprenant actif multi-label est d'étiqueter les données susceptibles de contribuer à la réduction attendue de la perte du modèle. Soit $P(\mathbf{x})$ la distribution à l'entrée, on indique par f_{D_l} , la fonction donnée de prédiction multi-label, étant donné l'ensemble d'entraînement D_l . Le jeu de labels prédits de $\mathbf{x}$ est $f_{D_l}(\mathbf{x})$. Supposons que l'ensemble des étiquettes vraies de $\mathbf{x}$ est $\mathbf{y}$, alors la perte estimée sur $\mathbf{x}$, peut être écrite comme $L(f_{D_l}(\mathbf{x}), \mathbf{y})$ (on la simplifiera en écrivant simplement $L(f_{D_l})$ par la suite), et la perte attendue de l'apprenant peut s'exprimer comme suit:

$$\widehat{\sigma}_{D_l} = \int_{\mathbf{x}} \left(\sum_{\mathbf{y} \in Y} L(f_{D_l}) P(\mathbf{y}|\mathbf{x}) \right) P(\mathbf{x}) d\mathbf{x} \quad (4.11)$$

Comme il est plutôt difficile d'estimer $P(\mathbf{x})$ directement, une façon pratique d'estimer $\widehat{\sigma}_{D_l}$ est de la mesurer sur tous les exemples dans D_u , comme D_u est d'habitude de taille assez large. Nous avons donc :

$$\widehat{\sigma}_{D_l} = \frac{1}{|D_u|} \sum_{\mathbf{x} \in D_u} \sum_{\mathbf{y} \in Y} L(f_{D_l}) P(\mathbf{y}|\mathbf{x}) \quad (4.12)$$

L'apprenant actif évaluera chaque ensemble possible de données non libellées D_s pour trouver l'ensemble optimal D_s^* des requêtes pour le jeu des labels associés. Lorsque D_s obtient ses étiquettes, il peut être incorporé à l'ensemble d'entraînement $D'_l = D_l + D_s$, et la perte escomptée pour le classifieur entraîné sur D'_l comme $\widehat{\sigma}_{D'_l}$. Le problème d'optimisation est de trouver le jeu de requêtes optimales D_s^* , qui une fois ajouté, engendrera la plus forte réduction sur la perte escomptée.

$$D_s^* = \arg \max_{D_s} (\widehat{\sigma}_{D_l} - \widehat{\sigma}_{D'_l}) = \arg \max_{D_s} \left(\sum_{\mathbf{x} \in D_u} \sum_{\mathbf{y} \in Y} (L(f_{D_l}) - L(f_{D'_l})) P(\mathbf{y}|\mathbf{x}) \right) \quad (4.13)$$

Comme dans [Campbell et al., 2000], nous supposons que tout $\mathbf{x}$ dans $D_u - D_s$ a un impact égal sur l'apprenant entraîné à partir de D_l et D'_l . Ainsi, nous aurons :

$$D_s^* = \arg \max_{D_s} \left(\sum_{\mathbf{x} \in D_s} \sum_{\mathbf{y} \in Y} (L(f_{D_l}) - L(f_{D'_l})) P(\mathbf{y}|\mathbf{x}) \right) \quad (4.14)$$

Dans le cadre de la stratégie de sélection d'échantillons avec *SVM* multi-label et selon l'équation (4.14), le problème d'optimisation peut être divisé en deux parties: comment mesurer la réduction de perte du classifieur multi-label et la façon de fournir une bonne estimation de probabilité pour la probabilité conditionnelle $p(y|x)$. On abordera ces deux questions respectivement dans les paragraphes suivants.

4.2.1 Estimation de la réduction de perte

Comme nous l'avons vu au sous-paragraphe 3.3 ci-dessus, le problème multi-label peut-être décomposé en des sous-problèmes un-contre-tous avec *SVM* utilisée comme classifieur binaire de base dans l'apprentissage actif. En décomposant le classifieur en plusieurs classifieurs binaires, la perte globale du classifieur peut être mesurée en recueillant la perte de tous les classifieurs binaires.

$$L(f) = \sum_{i=1}^k l(f^i) \quad (4.15)$$

où $l(f^i)$ est la perte du modèle classifieur binaire f^i . Alors le problème devient comment estimer la perte du modèle de chaque classifieur binaire. Comme suggéré par [Tong and Koller, 2002], cette perte du modèle peut-être mesurée par la taille de l'espace de versions d'une *SVM* binaire. Selon les deux auteurs, l'espace de versions d'une *SVM* peut être déduit comme suit:

$$V = \{\mathbf{w} \in W \mid \|\mathbf{w}\| = 1, y_i(\mathbf{w} \cdot \mathbf{x}_i) > 0, i = 1, \dots, n\} \quad (4.16)$$

où W désigne l'espace des paramètres. La taille d'un espace de versions est définie comme la surface de l'hypersphère $\|\mathbf{w}\| = 1$ dans W .

Basé sur le travail de [Tong and Koller, 2002], on peut utiliser la marge *SVM* comme la mesure de la taille de l'espace de versions. Lorsqu'un nouveau exemple est rajouté, on peut approximer la taille du nouvel espace de versions en calculant la marge *SVM* du classifieur mis à jour. Cependant, il est trop coûteux du point de vue calcul lorsque chaque donnée dans le pool non libellé doit être évaluée. Pour le rendre plus pratique, on peut appliquer l'idée heuristique dans [Tong, 2001] pour simplifier l'approximation en faisant correspondre la marge *SVM* du classifieur actuel à la taille du nouvel espace de versions.

Dans les réglages multi-label, on indique par $V_{D_i}^i$, la taille de l'espace des versions du classifieur binaire $f_{D_i}^i$ associé à la classe cible i et entraîné à partir des données libellées D_i . Après l'ajout d'une nouvelle instance de données $(\mathbf{x}, \mathbf{y}^i)$, où $\mathbf{y}^i \in \{+1, -1\}$ est le véritable label pour la donnée $\mathbf{x}$ sur la classe i , la réduction de perte du nouveau modèle classifieur binaire $f_{D_i}^i$, peut être estimée par :

$$\frac{l(f_{D_i \cup \{(\mathbf{x}, \mathbf{y}^i)\}}^i)}{l(f_{D_i}^i)} \approx \frac{V_{D_i \cup \{(\mathbf{x}, \mathbf{y}^i)\}}^i}{V_{D_i}^i} \approx \frac{1 + \mathbf{y}^i f_{D_i}^i(\mathbf{x})}{2} \quad (4.17)$$

Ensuite, la partie de réduction de perte de l'équation (4.14) peut être réécrite comme:

$$L(f_{D_l}) - L(f_{D'_l}) = \sum_{i=1}^k (l(f_{D_l}^i) - l(f_{D'_l}^i)) = \sum_{i=1}^k (l(f_{D_l}^i) \cdot (1 - \frac{l(f_{D'_l}^i)}{l(f_{D_l}^i)})) \quad (4.18)$$

On note que $l(f_{D'_l}^i)$ n'a rien à voir avec l'exemple non libellé sélectionné $\mathbf{x}$, donc on peut se concentrer sur l'optimisation du taux de réduction qui peut être approximé par :

$$\sum_{i=1}^k (\frac{1 - y^i f_{D_l}^i(\mathbf{x})}{2}) \quad (4.19)$$

Intuitivement, l'idée de l'estimation ci-dessus peut être expliquée comme suit. Considérons un exemple d'instances de données non étiquetées $\mathbf{x}$, si $\mathbf{x}$ peut être correctement prédit par le classifieur binaire courant f^i , alors plus la valeur de $|f^i(\mathbf{x})|$ est petite, plus incertain est le classifieur sur $\mathbf{x}$, et $\mathbf{x}$ mérite plus d'être libellé. Ceci est cohérent avec le résultat de la mesure ci-dessus, puisque $\mathbf{x}$ contribuerait davantage à réduire la taille de l'espace de versions, dans le cas où il sera correctement prédit. D'autre part, si le classifieur fournit un résultat erroné de prédiction pour $\mathbf{x}$, alors $|f^i(\mathbf{x})|$ sera d'autant plus large que le classifieur commettra plus d'erreur, d'où la nécessité de l'ajout d'autres $\mathbf{x}$ pour aider le classifieur à se corriger et à réduire grandement la taille de l'espace de versions.

4.2.2 Prédiction des étiquettes

Maintenant, on aborde la question de l'estimation de la probabilité conditionnelle $p(y|x)$, $y \in Y$. A noter que pour k labels, il y a 2^k combinaisons de labels possibles. Il est difficile pour l'apprenant actif d'estimer toutes ces possibilités. En particulier, il deviendra plus difficile encore lorsque les données d'entraînement sont assez limitées, ce qui est courant dans l'apprentissage actif.

Afin de simplifier l'estimation, on approxime la fonction de perte attendue par la fonction de perte sur la combinaison des labels les plus plausibles, puisque les labels prédits avec la plus grande confiance seront les plus probablement corrects. Ceci implique qu'on doit s'attendre à une perte ayant une large réduction puisque l'étiquetage le plus confiant serait le plus vraisemblable. Le problème devient donc comment produire une meilleure prédiction d'étiquette sur les données non libellées. L'approche de prédiction de label a été proposée dans [Yang et al., 2009] pour résoudre ce problème.

Au lieu d'estimer directement les labels possibles pour chaque donnée, on essaie tout d'abord de décider du nombre possible de labels que peut avoir chaque donnée, puis de déterminer les labels finaux basés sur la probabilité sur chaque label obtenue par le classifieur binaire correspondant. A supposer qu'il y ait k classes. En utilisant l'approche un-contre-tous, on peut avoir k classifieurs binaires. Compte tenu de la donnée $\mathbf{x}$, avec $p(y^j=1|\mathbf{x})$ comme la probabilité de $\mathbf{x}$ appartenant à la classe i , on peut obtenir k probabilités de classification sur $\mathbf{x}$ produites par les k classifieurs binaires et ensuite trier ces k probabilités par ordre décroissant.

Si $\mathbf{x}$ possède en fait m labels, les premières m probabilités sont censées être importantes tandis que les autres $k-m$ probabilités devraient être faibles. Sur cette hypothèse, on souhaiterait prédire le nombre de label pour chaque instance en se basant sur les probabilités générées par les classifieurs binaires.

Spécifiquement, le nombre de labels sera prédit en résolvant le problème de classification multi-classe. L'algorithme de la régression logistique est utilisé pour entraîner un modèle multi-classe et prédire les probabilités d'avoir un nombre différent de labels pour chaque donnée. Pour k classes, il existe k nombre possible d'étiquettes: $1, \dots, k$. Donc, on a k classes pour le problème de la classification multi-classe. Avant d'utiliser la régression logistique, on transforme la sortie de décision sur les données d'entraînement en fonction des probabilités de classification. Ici, on utilise la fonction sigmoïde [Lin et al., 2007] pour transformer la sortie SVM en valeurs de probabilités. Pour un exemple de données $\mathbf{x}$, on a :

$$p(y^i = 1|\mathbf{x}) = \frac{1}{1 + \exp(Af^i(\mathbf{x}) + B)} \quad (4.20)$$

où f^i est le classifieur SVM binaire associé à la classe i , A et B sont des valeurs scalaires obtenues par l'estimation du maximum de vraisemblance (voir section ci-dessus 3.3.1).

Le processus de prédiction du nombre de labels peut être décrit comme suit:

1. Utiliser le classifieur SVM pour attribuer des probabilités de classification pour tous les exemples de données.
2. Pour chaque instance $\mathbf{x}$, classer les probabilités de classification en ordre décroissant, $p(y^{i_1} = 1|\mathbf{x}) \geq p(y^{i_2} = 1|\mathbf{x}) \geq \dots \geq p(y^{i_k} = 1|\mathbf{x})$. Normaliser les probabilités de classification et obtenir $q_1(\mathbf{x}), \dots, q_k(\mathbf{x})$, où

$$q_p(\mathbf{x}) = \frac{p(y^{i_p} = 1|\mathbf{x})}{\sum_{t=1}^j p(y^{i_t} = 1|\mathbf{x})} \quad (4.21)$$

3. Entraîner le classifieur par la régression logistique. Pour chaque entraînement de données $\mathbf{x}$, présenter $[1: q_1(\mathbf{x}), 2: q_2(\mathbf{x}), \dots, k: q_k(\mathbf{x})]$ comme les caractéristiques d'apprentissage pour le modèle de régression logistique. Le nombre de labels de $\mathbf{x}$ sont utilisées comme catégorie pour entraîner un classifieur multi-classe.

4. Pour chaque donnée dans le pool non libellé, appliquez la régression logistique pour prédire les probabilités d'avoir des nombres différents de labels, et afficher le vecteur du label correspondant à la probabilité maximale d'être le nombre prédit de labels pour la donnée.

En supposant que le nombre le plus possible de labels pour la donnée $\mathbf{x}$ soit m , et $i_1, \dots, i_m$ sont les m classes associées aux plus grandes probabilités produites par les m SVM classifieurs binaires correspondants. Alors le vecteur du label prédit $\hat{y}$ peut être représenté par le vecteur binaire $[\hat{y}^{i_1} = 1, \dots, \hat{y}^{i_j} = 1, \hat{y}^{i_{j+1}} = -1, \dots, \hat{y}^{i_k} = -1]$; cette approche est connue comme la prédiction basée label par régression logistique.

En intégrant le vecteur du label prédit dans l'estimation de la perte attendue, on obtient ainsi la stratégie recherchée de sélection de données à savoir la réduction des pertes Maximales avec Confidentialité Maximale (MMC). La MMC peut être écrite comme :

$$D_s^* = \arg \max_{D_s} \left(\sum_{x \in D_s} \sum_{i=1}^k \left(\frac{1 - \hat{y}^i f^i(x)}{2} \right) \right) \quad (4.22)$$

Sur la base de la discussion ci-dessus et de la procédure décrite dans [Yang et al., 2009] pour la classification de textes, l'algorithme d'apprentissage actif proposé est décrit dans l'algorithme 1 suivant.

Algorithme 1 Apprentissage actif multi-label

Entrée: Ensemble libellé D_l

Jeu non libellé D_u

Nombre de classes k

Nombre d'itérations T

Nombre d'exemples sélectionnés par itération S

1: **pour** $t = 1$ à T **faire**

2: Entraîner k classifieurs SVM binaires $f^1, \dots, f^k$ basés sur les données d'entraînement D_l

3: **pour** chaque instance x dans D_u **faire**

4: Prédire son vecteur de labels en utilisant la méthode de prédiction basée sur la régression logistique décrite dans la section 4.2.2.

5: Calculer la réduction des pertes attendue avec le vecteur label le plus confiant $\hat{y}$,

$$score(x) = \sum_{i=1}^k \left(\frac{1 - \hat{y}^i f^i(x)}{2} \right)$$

6: Trier $score(x)$ dans l'ordre décroissant pour tout x dans D_u

7: Sélectionner un ensemble S d'exemples D_s^* avec les scores les plus importants,

et mettre à jour l'ensemble d'entraînement $D_l' \leftarrow D_l + D_s^*$

Dans le chapitre 5 suivant, nous évaluerons notre proposition d'apprentissage actif multi-label pour la classification multi-label sur des tâches se rapportant à six datasets du monde réel, relatifs à différents domaines d'application en comparaison avec l'état de l'art des approches d'apprentissage actif et de bien d'autres approches en classification multi-label telles la Binary relevance (BR), la ML- k NN, les cartes topologiques SOM, SOM-Y (basée label), SOM-mixtes, et pour les classifieurs SVM, la BR semi-SVM ou BR-SVM transductive. Pour les méthodes relevant de l'apprentissage actif, on s'est intéressé à trois types : la méthode *BinMin*, la méthode *Random* et la méthode *MMC*, qu'on vient de décrire dans la section 4.2.2.

Sur tous les datasets, la classification un-contre-tous est effectuée pour chaque catégorie et le problème de classification multi-label est traité comme plusieurs problèmes de classification binaire, où les instances de la catégorie cible sont données comme label positif (c'est-à-dire $y = 1$), et le reste des instances reçoit un label négatif (c'est-à-dire $y = -1$). *SVM^{Light}* Package [Joachims, 2002] est téléchargé et utilisé pour entraîner le classifieur

binaire. Le type du noyau (kernel linéaire ou gaussien) et le paramètre de pénalité C sont choisis en fonction du domaine d'application.

Dans nos expériences d'apprentissage actif sur chaque dataset, on sélectionne aléatoirement auparavant un petit ensemble d'instances pour constituer l'ensemble initial libellé, et on laisse les instances restantes comme pool ou groupe non libellé. Ensuite, l'apprenant actif (classifieur) sélectionne un nombre donné d'exemples à partir du pool non libellé dans chaque itération, puis les rajoute à l'ensemble libellé avec leurs labels respectifs.

On effectue plusieurs itérations d'apprentissage actif sur chaque dataset jusqu'à ce que l'apprenant atteigne une précision suffisante. A chaque itération, une fois les données sélectionnées incorporées, l'apprenant actif ré-entraîne le classifieur sur l'ensemble libellé élargi et sa performance est évaluée sur les autres exemples restants de données. Nous avons utilisé le score de micro-average F1 comme mesure d'évaluation, puisqu'il s'agit d'une évaluation standard utilisée dans la plupart des recherches précédentes sur la classification, notamment des textes. Comme défini dans [Yang, 2001], voir aussi paragraphe 6.2 du chapitre 2, équation (2.12), le score de la *micro-F1* dans le cas de plusieurs labels est donnée comme suit :

$$\frac{2 \sum_{j=1}^k \sum_{i=1}^n \hat{y}_i^j y_i^j}{\sum_{j=1}^k \sum_{i=1}^n \hat{y}_i^j + \sum_{j=1}^k \sum_{i=1}^n y_i^j} \quad (4.23)$$

où n est le nombre de données de test, y_i est le vecteur du label vrai de la $i^{\text{ème}}$ instance de données, $y_j^j = 1$ si l'instance appartient à la catégorie j , sinon $y_j^j = -1$. $\hat{y}_i$ est le vecteur du label prédit. Nous calculons la moyenne des scores de *micro-F1* pour chaque itération d'apprentissage actif basée sur 10 expériences.

Dans nos expériences, nous évaluerons et comparons trois méthodes d'apprentissage actif:

- a) *MMC* : La méthode d'apprentissage actif citée précédemment à la section 2.2. de ce chapitre. Elle a surclassé la sélection aléatoire Random et Bin-Min dans le domaine de la classification de textes et réduit significativement le coût d'étiquetage [Yang et al., 2009].
- b) *Random* (aléatoire) : La stratégie de sélection de base de l'échantillon qui consiste à sélectionner aléatoirement des exemples de données du pool non libellé.
- c) *BinMin*. Il s'agit d'une stratégie de sélection d'échantillons proposée dans [Brinker, 2006]. L'approche un-contre-tous est utilisée pour la classification multi-labels, et la *SVM* est utilisée comme classifieur binaire de base. L'exemple optimal non étiqueté est choisi selon :

$$\arg \min_x \min_{i=1, \dots, k} |f^i(x)| \quad (4.24)$$

où f^i est le classifieur binaire sur le problème binaire associé à la classe i . C'est-à-dire qu'il sélectionne des exemples non libellés par rapport au label le plus incertain.

Comme indiqué auparavant, en section 2, cette méthode ne tire pas partie des avantages des informations multi-label.

Ces approches relevant de l'apprentissage actif seront comparées à celles semi-supervisées standard en vue d'évaluer leurs performances sur six datasets de différentes tailles et portant sur des domaines d'application divers.

5. Classifieur d'apprentissage actif topologique proposé

Les modèles *SOM* sont des algorithmes compétitifs de réseaux neuronaux artificiels populaires basés sur l'apprentissage non supervisé et sont largement utilisés dans la classification des données. Après l'apprentissage, la carte produite par l'algorithme *SOM* préserve généralement la propriété topologique des modèles d'entrée, c'est-à-dire que les vecteurs de poids qui sont voisins dans l'espace d'entrée sont mappés autour des neurones voisins dans la carte autour des meilleures unités correspondantes BMU (Best Matching Units). Ce travail tente de profiter de l'apprentissage de clusters de cartes *SOM* afin de combiner les données sur un espace dimensionnel plus petit.

Formellement, l'apprentissage semi-supervisé *SOM* vise à construire une fonction de classification à partir d'un ensemble fini d'échantillons d'entrée partiellement classés, puis à affecter les étiquettes de classe aux échantillons d'entrée, de sorte qu'il vise à généraliser le processus afin qu'émergent les clusters ou sous-ensembles d'échantillons d'entrée qui sont compatibles avec des étiquettes de classes identiques.

Chaque neurone à l'intérieur de la carte *SOM* produit des zones de clusters en fonction de l'approximation de la densité de données connectées aux vecteurs d'entrée durant les itérations de l'apprentissage. Selon [Abbas et al., 2013], *SOM* montre plus de précision dans la classification de la plupart des objets à leurs clusters que d'autres algorithmes de clustering tels que *k-means*.

En outre, les connaissances tirées d'une approche semi-supervisée ont conduit à une meilleure classification des instances non libellées en relation avec le clustering non supervisé et la classification supervisée, d'où l'intérêt croissant de nombreux chercheurs dans l'apprentissage semi-supervisé (SSL) [Abei et al., 2015]. Abaei et al. ont proposé un meilleur modèle de prédiction en utilisant un seuil pour la carte *SOM* lorsque les données sont non libellées. Dans leur étude, la carte *SOM* est utilisée au lieu de *k-means* en raison de plusieurs inconvénients de celui-ci, tels que le problème des minima locaux et la sensibilité aux données bruitées (voir Table 3.1).

Un nouveau modèle de cartes topologiques dédiées à des données mixtes a été proposé par les auteurs [Rogovschi et al., 2008] sur la base de la version par lots des deux algorithmes, celui de Kohonen pour les données continues et celui dédié aux données binaires [Rogovschi et al., 2011]. Ces deux modèles ont été expérimentés pour tenir compte de la pondération associée aux variables catégorielles et continues, en ce qui a trait à l'apprentissage en mono-label non supervisé.

Pour l'apprentissage actif, l'idée est tirée de l'article de [Patra and Bruzzone, 2014] qui ont utilisé une nouvelle technique d'apprentissage actif basée sur *SOM-SVM* pour la télédétection et la classification des images en couleur ainsi que de véritables images multi-spectral et hyper-spectral de télédétection.

De notre côté, on utilisera la carte topologique pour son pouvoir d'abord de réseau neuronal compétitif, de reconfiguration des données en préservant la topologie et surtout de la possibilité d'interprétation des résultats obtenus. Ainsi, on fera appel à la carte *SOM* et ses variantes pour l'apprentissage actif dans le but de sélectionner les instances les plus informatives à libeller en adoptant une stratégie de sélection basée sur l'incertitude des labels et où l'étiquetage inutile des échantillons non informatifs est évité, réduisant ainsi considérablement le coût de l'étiquetage tout en augmentant la qualité de l'ensemble d'apprentissage.

Les auteurs dans [Patra and Bruzzone, 2014] ont présenté une technique d'apprentissage actif qui sélectionne l'échantillon le plus incertain tout en étant le plus proche de l'hyperplan de séparation courant d'un *SVM*. On essaiera, au niveau de la carte *SOM*, de sélectionner l'échantillon le plus incertain tout en étant le plus proche de la zone de séparation entre deux clusters, la région à plus faible densité.

Toutes les stratégies d'échantillonnage se basent d'une part sur les échantillons les plus incertains à chaque itération de l'apprentissage actif et d'autre part, tentent d'incorporer un critère de diversité, pour ne pas introduire ainsi éventuellement une redondance dans les échantillons sélectionnés [Tuia et al., 2013; Patra and Bruzzone, 2012]. Dans l'étape d'incertitude, les échantillons les plus incertains sont sélectionnés en utilisant un critère d'incertitude donnée, puis dans l'étape de diversité, on sélectionne parmi ces échantillons les plus incertains, un certain nombre d'entre eux en appliquant un critère de diversité.

Il faut s'attendre, dès à présent, que les résultats seront tributaires du type du dataset considéré en tenant compte de leur complexité théorique (*TCS*) et de leur densité I_D (voir paragraphe 2.3 au chapitre 5).

Dans le paragraphe 3.2 du chapitre 3, on a présenté une technique d'apprentissage en mode batch basée sur la carte auto-organisée (*SOM*) [Kohonen, 2001]. La proposition exploite l'hypothèse de clusters pour trouver les échantillons les plus informatifs parmi ceux choisis en appliquant un critère d'incertitude et de diversité à chaque itération de l'apprentissage actif. L'hypothèse de clusters est équivalente à l'hypothèse de séparation de faible densité qui stipule qu'une décision entre les classes devrait reposer sur la région de l'espace de caractéristiques de faible densité.

Selon cette hypothèse, on peut dire que deux points dans l'espace de caractéristiques sont susceptibles d'avoir le même label de classe s'il existe un chemin les reliant à travers des régions à haute densité uniquement [Rigollet, 2007]. Dans la méthode d'apprentissage actif, d'abord, un réseau *SOM* est entraîné en mode supervisé avec les instances libellées disponibles ou avec un sous-modèle échantillonné d'instances libellées pour limiter le temps d'apprentissage. Après entraînement, nous calculons la distance moyenne de chaque neurone dans la couche de sortie par rapport à ses neurones voisins en utilisant leurs vecteurs de poids.

Dans l'hypothèse où la *SOM* préserve la topologie des motifs d'entrée, les neurones cartographient les échantillons qui appartiennent à des régions de faible densité de l'espace d'entrée et qui ont une plus grande distance de voisinage moyenne que les neurones qui cartographient des échantillons qui appartiennent à des régions à forte densité. Autrement dit, selon l'hypothèse du clustering, on peut dire que les neurones qui ont une distance voisine

moyenne plus élevée ont une forte probabilité de cartographier les échantillons se trouvant aux limites. À la convergence du traitement de *SOM*, nous commençons la procédure de l'apprentissage itératif actif.

Ensuite, un lot d'échantillons y est choisi qui correspond à la cartographie de *SOM* des neurones ayant la plus grande distance de voisinage moyenne ou la plus faible confiance dans la classification. Cela nous permet d'incorporer la propriété de l'hypothèse de cluster pour sélectionner les échantillons les plus incertains, c'est-à-dire, les plus informatifs pour l'étiquetage. Ainsi, la technique proposée peut facilement utiliser l'hypothèse du clustering et de la diversité dans le processus de sélection de l'échantillon en exploitant les propriétés du réseau neuronal *SOM*. L'avantage principal d'utiliser l'hypothèse du clustering est que, de cette manière, nous pouvons localiser avec précision les échantillons d'entraînement pertinents près de la limite de décision entre les classes quand des échantillons d'entraînement biaisés sont pris en considération au départ.

Les résultats expérimentaux montrent l'efficacité de la méthode proposée qui a été validée sur les six datasets de différents domaines d'application. La figure 4.2 représente le schéma bloc complet de la méthode proposée.

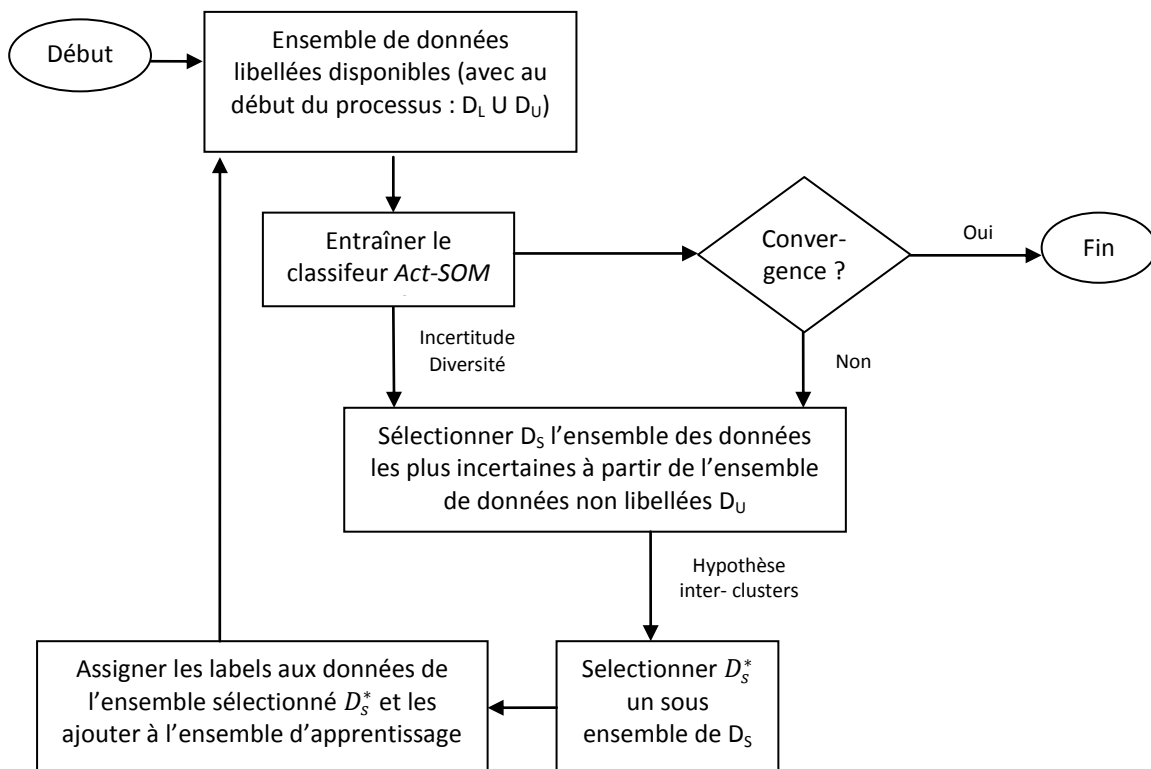


Figure 4.2 – Organigramme de la méthode proposée avec $D_U \rightarrow D_S \rightarrow D_S^*$

La technique proposée comprend deux étapes principales. Dans la première étape, un réseau *SOM* est entraîné de manière supervisée afin d'identifier les échantillons importants disponibles qui appartiennent à l'espace de caractéristiques à faible densité D_S . Ceci est accompli en actualisant le vecteur de poids associé à chaque neurone dans la carte afin que, lorsque le réseau atteint la convergence, les vecteurs poids décrivent une cartographie à partir de la dimension supérieure de l'espace de fonctions d'entrée à un espace inférieur de sortie/carte.

Notre contribution est d'utiliser le paradigme de l'apprentissage actif avec le pouvoir topologique de la carte SOM pour la classification semi-supervisée multi-label, en tenant compte des informations multi-label, et en sélectionnant des données non libellées qui peuvent conduire à la plus grande réduction attendue de la perte du modèle. Dans l'approche proposée, à notre connaissance, c'est une première tentative d'utilisation dans le contexte de classification multi-label en considérant la procédure d'apprentissage actif pour ces modèles topologiques ; on nommera cette proposition *Act-SOM*.

Dans l'approche proposée, nous utilisons comme mesure de similarité, les distances Euclidienne et celle de Hamming $H(\cdot)$, calculées en moyenne respectivement dans le cas continu des variables d'instances et dans le cas binaire des labels. SOM est capable d'atteindre l'optimum global dans l'espace de recherche, plus adapté aux jeux de données bruitées, avec une structure complexe, mais les résultats sont visuels et faciles à analyser, sensibles à la topologie de la couche de sortie et la taille de la carte peut être déterminée (voir Table 3.1 du chapitre 3). En comparaison avec SOM, l'approche SOM-Y utilise la distance euclidienne et permet d'associer à chaque label un prototype en prenant les exemples les plus proches du prototype, puis d'effectuer un vote majoritaire en appliquant également différents masques sur les labels pour voir leurs comportements.

5.1 Formulations mathématiques

Nous définissons la tâche de l'apprentissage multi-étiquettes en donnant en entrée, un ensemble libellé $L = \{\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^l \in \mathcal{R}^D \times \{0,1\}^Q$ et aussi en entrée, un ensemble plus grand non libellé $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$, où chaque observation d'entrée $\mathbf{z}_i$ est composée d'une partie instance $\mathbf{x}_i \in A / \mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ et éventuellement son vecteur de labels correspondant $\mathbf{y}_i \in Y / \mathbf{y}_i = \{y_{i1}, y_{i2}, \dots, y_{iQ}\}$ codé en variable binaire. Ainsi, une observation particulière de L est un vecteur d'entrée pour *Act-SOM*, $\mathbf{z}_i$ (valeurs d'instance et de label) de dimension $d = D + Q$.

Nous définissons la distance $\delta(c, r)$, comme la longueur unique du chemin le plus court sur la grille entre les cellules c et r et le système de voisinage de type gaussien, comme une fonction kernel décroissante et monotone $K(\delta(c, r))$ qui représente l'influence mutuelle des deux cellules (voir Figure 3.6 du chapitre 3). En pratique, comme pour la carte topologique traditionnelle, une fonction lisse est associée pour contrôler la taille du voisinage comme $K^T(\delta(c, r)) = \exp\left(\frac{-\delta(c, r)}{T}\right)$ où T devient un paramètre du modèle décroissant progressivement selon les itérations.

Nous associons un vecteur référent $\mathbf{w}_c = (\mathbf{w}_c^{inst}, \mathbf{w}_c^{lab})$ de même dimension d , où $\mathbf{w}_c^{inst} \in \mathcal{R}^D$ et $\mathbf{w}_c^{lab} \in \{0,1\}^Q$ représentent respectivement les vecteurs de référence de l'instance et sa partie libellée pour chaque cellule c . Nous désignons par $W = (W^{inst}, W^{lab})$, l'ensemble des vecteurs référents pour les parties exemple et labels associés.

Considérant $A = L \cup U$ comme l'ensemble global où sont toutes les l instances libellées par leurs étiquettes y , émises de L et aussi les instances x non libellées de U ($\text{Card}(A) = \text{Card}(L) + \text{Card}(U) = l + u$).

Notre carte *SOM* active, *Act-SOM* utilise une procédure itérative en trois étapes, comme suit:

- 1: Faire apprendre la carte par les échantillons de l'ensemble d'apprentissage L (apprentissage semi-supervisé).
- 2: Trouver les BMU correspondants pour chaque $\mathbf{z}_i$, noté $\phi(\mathbf{z}_i)$ comme neurone gagnant, comme suit:

$$\forall \mathbf{z}_i = (x_i, y_i), \quad \phi(\mathbf{z}_i) = \arg \min_c \left((1-\alpha) \|\mathbf{x}_i - \mathbf{w}_c^{inst}\|^2 + \alpha H(y_i, \mathbf{w}_c^{lab}) \right) \quad (4.25)$$

Dans cette expression, H est la distance de Hamming pour la partie label. Le paramètre de réglage $0 \leq \alpha \leq 1$ détermine combien les poids doivent être biaisés par les labels.

- 3: Affecter, pour chaque $\mathbf{z}_i$, le poids des neurones du vecteur référent $\mathbf{w}_c$ comme suit:

- Pour la partie d'instance $\mathbf{w}_c^{inst}$ comme vecteur moyen:

$$\mathbf{w}_c^{inst} = \frac{\sum_{\mathbf{z}_i \in A} K(\delta(\phi(\mathbf{z}_i), c)) \mathbf{x}_i}{\sum_{\mathbf{z}_i \in A} K(\delta(\phi(\mathbf{z}_i), c))} \quad (4.26)$$

- Pour chaque composante de la partie label $\mathbf{w}_c^{lab} = (w_{c1}^{lab}, \dots, w_{ck}^{lab}, \dots, w_{cQ}^{lab})$, comme centre médian de la partie binaire des observations $\mathbf{z}_i \in A$ pondéré par $K(\delta(\phi(\mathbf{z}_i), c))$.

$$\mathbf{w}_{ck}^{lab} = \begin{cases} 0 & \text{if } \left[\sum_{\mathbf{z}_i \in A} K(\delta(\phi(\mathbf{z}_i), c)) (1 - y_{ik}) \right] \geq \\ & \left[\sum_{\mathbf{z}_i \in A} K(\delta(\phi(\mathbf{z}_i), c)) y_{ik} \right] \\ 1 & \text{otherwise} \end{cases} \quad (4.27)$$

Le processus de minimisation est effectué itérativement par rapport à ϕ (équation (4.25)) et ensuite $\mathbf{W}$ (équations (4.26) - (4.27)) en prenant chaque fois l'un des deux paramètres de la fonction de coût global $\xi(\phi, \mathbf{W})$ à minimiser et qui est donnée par:

$$\xi(\phi, \mathbf{W}) = (1-\alpha) \sum_{\mathbf{z}_i \in A} \sum_{r \in \zeta} K(\delta(\phi(\mathbf{z}_i), r)) \|\mathbf{x}_i - \mathbf{w}_r^{inst}\|^2 + \alpha \sum_{\mathbf{z}_i \in A} \sum_{r \in \zeta} K(\delta(\phi(\mathbf{z}_i), r)) H(y_i, \mathbf{w}_r^{lab}) \quad (4.28)$$

Le problème est donc de savoir comment produire une meilleure prédiction des étiquettes sur des données non libellées, nous pouvons également appliquer l'approche de prédiction de l'étiquette proposée dans [Yang et al., 2009] qui consiste pour chaque donnée dans le pool non libellé à appliquer une régression logistique pour prédire le vecteur d'étiquette correspondant.

La pondération de la distance est très utile surtout dans le cadre de *Act-SOM*, car cela nous permet, au cours de la phase d'apprentissage, d'ajuster l'adaptation en tenant compte de l'importance de chaque attribut, le coefficient de pondération α dans l'équation (4.25) peut être estimé en fonction de la pertinence des attributs. Ainsi, α nous permet de moduler les deux parties d'instances-labels au niveau des vecteurs d'entrée, et aussi, comme on le verra plus loin dans la procédure d'apprentissage actif (voir algorithme 2 de l'apprentissage actif *SOM* ci-dessous), pour pouvoir le considérer dans les instances des vecteurs transductifs afin d'en déduire leurs labels équivalents.

5.2 Stratégies de sélection d'échantillons avec Act-SOM multi-label

Dans notre méthode, à la convergence de la phase d'entraînement, nous calculons la distance de voisinage moyenne de chaque neurone de la carte à ses neurones voisins en utilisant leurs poids correspondants. La distance voisine moyenne du neurone k , notée $d(w_k)$, est calculée comme suit:

$$d(w_k) = \frac{1}{|N_k^r|} \sum_{i \in N_k^r} \|w_k - w_i\|^2 \quad (4.29)$$

Où N_k^r représente l'ensemble des neurones de la carte qui sont dans le système de voisinage d'ordre $r^{\text{ième}}$ du neurone k . Sous l'hypothèse selon laquelle SOM préserve la propriété topologique de l'espace d'entrée, on peut utiliser l'ensemble des mesures des distances de proximité moyennées obtenues pour identifier les échantillons de l'espace de caractéristiques à faible densité. Ces échantillons sont associés aux neurones qui ont des valeurs plus grandes de distance moyenne.

En conséquence, nous pouvons affirmer que, en raison de l'hypothèse de clustering, les neurones qui ont des distances moyennes de proximité plus élevées ont une probabilité plus élevée de cartographier les échantillons aux frontières que les neurones ayant une distance voisine moyenne inférieure. Cette information est alors exploitée dans la deuxième étape de la méthode pour lancer l'itération de l'apprentissage actif.

La deuxième étape de la technique proposée vise à sélectionner les échantillons les plus informatifs D_S^* à chaque itération du processus d'apprentissage actif pour résoudre un problème de classification de classes. Après entraînement, pour chaque échantillon non libellé x dans le pool non libellé U , K distances fonctionnelles $f_k(x)$, $k = 1, 2, \dots, K$, sont obtenues. Ensuite, la valeur de l'incertitude $s(x)$ associée à la fiabilité de la classification de chaque échantillon non libellé $x \in U$ peut être calculée. Il convient de noter que la confiance peut être liée à l'incertitude associée à l'échantillon considéré. La stratégie peut être utilisée pour calculer l'incertitude en se basant sur l'échantillonnage marginal largement utilisé [Demir et al., 2011], où la plus petite distance entre les K décisions, c'est-à-dire $\min_{k=1,2,\dots,K} \{f_k(x)\}$ est considérée pour calculer la valeur de l'incertitude $s(x)$ de chaque échantillon non libellé $x \in U$, c'est-à-dire :

$$s(x) = \arg \max_i \min_k \|x_i - w_{ik}\|^2 \quad (4.30)$$

Avec cette stratégie, l'incertitude de chaque échantillon non libellé $x \in U$ est mesurée selon sa valeur $s(x)$ correspondante. Les échantillons qui ont des valeurs de confiance plus faibles sont les plus incertains étant donné qu'ils ont la confiance de la classification correcte la plus basse.

Après calcul de l'incertitude de chaque échantillon non libellé en utilisant (4.30), nous sélectionnons le lot des échantillons de U qui ont les valeurs de confiance les plus faibles et sont mappés en différents neurones de la carte SOM selon les résultats obtenus lors de la première étape. Cela nous permet de sélectionner un lot d'échantillons parmi les plus incertains qui sont différents les uns des autres parce que des motifs d'entrée similaires sont mappés dans le même neurone. Puis un sous lot d'échantillons D_S^* parmi le lot déjà choisi des

échantillons D_S est sélectionné qui correspond aux neurones de mappage de *SOM* ayant les distances moyennes de voisinage les plus élevées calculées en (4.29). Ce qui permet d'incorporer la propriété d'hypothèse de clustering dans la sélection des échantillons les plus informatifs pour l'étiquetage.

En d'autres termes, nous sélectionnons des échantillons qui sont à la fois incertains, divers et situés dans des régions de faible densité de l'espace de caractéristiques (c'est-à-dire, selon l'hypothèse du clustering sur la limite des régions de décision). Cela peut être particulièrement utile lorsque les ensembles d'entraînement qui ne modélisent pas la distribution réelle des données à la limite de décision sont disponibles. Le processus est réitéré jusqu'à ce qu'un critère d'arrêt (qui peut être lié à la stabilité de la précision ou à sa valeur) soit satisfait. La procédure plus détaillée d'un tel apprentissage actif pour la carte *SOM* est comme suit :

- a) Faire apprendre le réseau neuronal *SOM* en utilisant les deux partitions du dataset disponibles (libellé L et non libellé U).
- b) Calculer la distance moyenne de voisinage de chaque neurone en utilisant (4.29)
- c) Répéter jusqu'à ce qu'un critère d'arrêt est satisfait et/ou l'ensemble d'entraînement final est obtenu :
 - i) Faire apprendre avec les échantillons initiaux disponibles (libellés et non libellés), les K clusters du classifieur *SOM* (chacun associé à une classe informative spécifique) organisée en apprentissage semi-supervisé
 - ii) Calculez la valeur de l'incertitude de chaque échantillon non libellé en utilisant (4.30).
 - iii) Sélectionnez D_S , le pool d'échantillons de U qui ont les valeurs d'incertitude les plus grandes (c'est-à-dire valeurs de confiance les plus faibles) et sont mappés dans des neurones distincts de *SOM* (critère de diversité).
 - iv) Sélectionnez D_S^* , les échantillons de D_S qui correspondent aux neurones de la cartographie de *SOM* ayant les distances moyennes les plus élevées (exploitation de l'hypothèse inter-clusters).
 - v) Assignez les étiquettes aux échantillons sélectionnés, et incluez-les dans le jeu d'entraînement.

L'algorithme 2 suivant concerne la classification en semi-supervisé multi-label relatif à l'apprentissage actif de la carte topologique *SOM* combinant instances et labels.

Dans l'algorithme2, la ligne (3) considère les entrées d'instances-labels (x, y) de L et celles des x de l'ensemble non libellé U non pour calculer tous les centroïdes de la carte *Act-SOM* de $i = 1, \dots, l, \dots, u, \dots, l + u$ pour tout $x_i \in A = LUU$ et seulement $i = 1, \dots, l$ pour les étiquettes y_i selon l'équation (4.25) qui nous donne le mappage des clusters selon les deux modalités (instances, labels). Ensuite, en mettant la pondération à $\alpha = 0$ (c'est-à-dire que nous ne considérons pas la partie des labels pour chaque x de U), nous évaluons les points d'instances les plus éloignés, les étrangers ou outsiders, en fonction de la ligne (5) à ces clusters appris en ligne (3).

Algorithm 2 : Apprentissage actif par Act-SOM

Entrée:Ensemble libellé L Ensemble non libellé U ,Nombre d'itérations $Iter$,Nombre d'exemples sélectionnés par itération S 1: **Pour** $\alpha = 0$ to 1 par pas de 0.1 **Faire**2: **Pour** $t=1$ à $Iter$ **Faire**

Entraîner le classifieur *Act-SOM* basé sur l'ensemble d'apprentissage LUU qui combine la distance Euclidienne pour les instances (x_i) et la distance de Hamming pour les labels (y_i) en utilisant (4.26) et (4.27) aboutissant aux vecteurs W comme référents des clusters

4: **Pour** tout x dans U

5: Calculer l'incertitude, qui est mesurée en accord avec les neurones de la carte *SOM* qui ont la plus grande distance moyenne en utilisant :

$$s(x) = \arg \max_i \min_k \|x_i - w_{ik}\|^2$$

6: Trier l'incertitude $s(x)$ en ordre décroissant pour tous les x dans U 7: Sélectionner un ensemble de données S avec la plus grande incertitude8: Assigner les labels aux données sélectionnées S , et les rajouter à l'ensemble d'apprentissage

Dans le cadre de l'apprentissage actif, nous avons utilisé l'une des stratégies les plus connues, celle du traitement de données par lots, en utilisant des clusters. Nous avons combiné la méthode *Act-SOM* avec l'un des critères d'apprentissage actifs utilisés dans [Patra et al., 2014] où les auteurs ont présenté une technique d'apprentissage actif qui sélectionne l'échantillon le plus incertain tout en étant le plus proche de l'hyperplan de séparation actuel d'un *SVM*.

Au niveau de la carte *SOM*, nous essaierons de sélectionner l'échantillon le plus incertain tout en étant le plus proche de la zone de séparation entre les clusters, la région avec une densité plus faible, comme indiqué dans l'équation (4.30). Les cas les plus incertains sont les étrangers (les moins discriminants) et donc les plus informatifs pour améliorer la prédiction de la classification.

6. Conclusion

Ce chapitre avait pour but la concentration des efforts fournis en matière d'apprentissage semi-supervisé multi-label combiné à l'apprentissage actif. On a essayé de donner un aperçu sur les travaux connexes relatifs à la classification multi-label, ensuite de synthétiser les différentes approches vues au chapitre 2 et qu'on a adoptées pour le problème d'apprentissage multi-label en mode semi-supervisé et enfin les stratégies suivies durant la problématique de d'apprentissage actif multi-label où l'apprenant actif est basé sur la *SVM*. On verra aussi au chapitre 5, côté implémentation, la procédure incrémentale adoptée pour un suivi graduel par

pool de 5% d'instances à libeller afin d'arriver à un apprentissage le moins coûteux possible en restant performant.

L'apprenant actif ou classifieur sélectionne itérativement un échantillon de données à étiqueter sur la base de certaines stratégies de sélection suggérant les données qui méritent le plus d'être étiquetées. Ainsi, l'affectation de labels pour des données multi-label est beaucoup plus complexe que pour les données à un seul label.

Malgré la valeur et la signification de ce problème, la recherche est très limitée sur l'apprentissage actif multi-label. La plupart des recherches sur l'apprentissage actif se focalisent sur le problème de classification soit de textes ou d'images. La stratégie de sélection de l'échantillon suit strictement l'hypothèse que chaque instance a un seul libellé et ne peut donc pas être directement appliquée dans l'apprentissage actif multi-label.

Dans ce travail, nous proposons l'application d'une approche d'apprentissage actif multi-label pour la classification dans différents domaines (biologie, médecine, texte, multimédia). On essaiera de présenter dans ce cadre d'apprentissage actif, des stratégies d'échantillonnage d'incertitude multi-label des instances à partir des perspectives de prédiction.

En plus des stratégies d'apprentissage actif multi-labels elles-mêmes, la façon dont l'évaluation de ces méthodes est effectuée est également une caractéristique importante pour le travail connexe. Certains aspects importants à considérer sont la taille du dataset libellé initial, la taille et la manière du pool à libeller (instances aléatoires ou positives et/ou négatives), l'ensemble des exemples utilisés, la stratégie d'échantillonnage et aussi l'approche d'évaluation. La taille du pool définit le nombre d'exemples demandés dans chaque cycle d'apprentissage actif.

Ensuite, les jeux de données ont des propriétés différentes selon le nombre d'instances, le nombre de labels, peu ou prou de multi-label différents (parcimonie), de fréquence de labels et de degré de cardinalité, entre-autres. Un dataset est plus difficile à étudier qu'il est déséquilibré. Tous ces paramètres seront pris en charge pour arriver à avoir des indices sur la complexité effective d'un dataset avant de pouvoir l'étudier, selon la caractérisation de certaines métriques. En classification multi-label, tout est dans l'estimation par le prétraitement du dataset pour déterminer s'il présente des difficultés et nécessite de ce fait, une investigation poussée pour son apprentissage. Tous ces aspects connexes seront décrits pour chaque dataset.

À cette fin, le chapitre 5, commencera d'abord par décortiquer les six datasets selon ces paramètres qu'on verra en détail, ensuite, on abordera l'apport de l'apprentissage actif sous sa forme classique (habituelle) avec les séparateurs SVM (à base de classifieurs binaires OVA, un-contre-tous) ; et ce qui constitue notre principale contribution, sous forme préliminaire (non encore usitée) de cartes topologiques auto-organisatrices SOM (à base de clusters compétitifs intégrés, tous à la fois). Cette différence d'architecture et de conception au niveau de l'apprentissage est à l'avantage des SOM pour tirer profit dans le cadre multi-label, de la corrélation globale entre les attributs instances et labels associés et aussi de l'informativité unifiée des instances non libellées.

Chapitre 5

Cadre expérimental et résultats

Sommaire

1	Introduction.....	100
2	Caractéristiques des datasets multi-label et métriques associées.....	102
	2.1 Caractéristiques des datasets multi-label.....	103
	2.2 Les métriques d'évaluation adoptées.....	105
	2.3 Score de complexité théorique (TCS).....	108
3	Classifieurs de base multi-label.....	110
4	Résultats de TSVM active en multi-label.....	114
5	Résultats de l'approche proposée de SOM en apprentissage actif multi-label..	119
6	Conclusions.....	125

Chapitre 5 : Cadre expérimental et résultats

Résumé :

La classification multi-label est de plus en plus répandue comme technique d'exploration des données. Son objectif est de catégoriser les modèles dans plusieurs groupes non exclusifs, et elle est appliquée dans des domaines tels que la catégorisation des nouvelles (news), l'étiquetage d'image et la classification musicale, entre-autres, comme on l'a déjà vue au chapitre 2. Comparativement parlant, la classification multi-label est une tâche plus complexe que la classification multi-classe et binaire, puisque le classifieur doit apprendre la présence de différentes sorties à la fois à partir du même ensemble de variables prédictives.

A partir de la littérature et des observations de résultats déjà obtenus, on peut déduire que, plus le dataset multi-label est complexe, est plus les méthodes de classification multi-label deviennent moins performantes. Comme d'habitude lors de l'évaluation d'une tâche de classification multi-label, la performance dépend non seulement de l'algorithme sélectionné, mais aussi des traits et spécificités du dataset multi-label et des mesures d'évaluation choisies qui doivent être prises en compte.

Ce chapitre présente une nouvelle métrique de caractérisation *TCS* (Theoretical Complexity Score) [Herra, 2016], visant à évaluer la complexité intrinsèque d'un dataset multi-label, ainsi qu'un ensemble de résultats allant des :

- 1) Classifieurs de base en multi-label, y compris des cartes topologiques, en passant par les SVM pour procéder à une étude comparative avec l'existant;
- 2) Classifieur transductifs *TSVM* en apprentissage actif avec les trois méthodes d'échantillonnage étudiées au chapitre 4 (aléatoire, *BinMin* et *MMC*), avec des propositions faites à ce niveau et les résultats obtenus de leur pertinence pour la fonction *BinMin* qui semble conçue pour s'adapter aux traits des données multi-libellées de divers domaines d'application ;
- 3) Classifieurs topologiques actifs en multi-label, en adoptant une stratégie s'inspirant des travaux de [Patra, 2014] pour des images hyper-spectrales, relatif à l'évaluation par l'incertitude des labels.

Chaque fois que c'est possible, on essaiera de mener une comparaison avec les résultats de la littérature, mais à notre connaissance, l'apprentissage multi-label avec une étude expérimentale aussi variée des datasets de différents domaines impliquant des cartes topologiques dans ce sens en mode semi-supervisé actif semble être notre pierre d'achoppement.

1. Introduction

Dans le monde de l'apprentissage semi-supervisé, l'apprentissage actif regroupe un ensemble de méthodes de sélection d'exemples utilisées pour construire l'ensemble d'apprentissage du modèle de manière itérative. Toutes les stratégies ont en commun de chercher à utiliser le moins d'exemples possible et de sélectionner les exemples les plus informatifs.

Notre travail a pour finalité d'avoir un meilleur classifieur qui permettrait une meilleure classification avec un coût minimal d'échantillons libellés. Pour se faire, le plus important est de choisir l'échantillon d'apprentissage le plus représentatif où réside l'utilité de l'apprentissage actif.

Alors que la transformation des données est un moyen relativement simple de classer les datasets multi-label dans les classifieurs classiques, l'approche basée sur l'adaptation de ces classifieurs pour aborder les données originales multi-label a également été explorée. Dans ce chapitre, un grand nombre de ces adaptations de méthodes sera introduit, s'appuyant sur des algorithmes traditionnels basés les *kNN*, les cartes topologiques *SOM* et ses dérivées ainsi que les *SVM* et ses dérivées.

L'apprentissage actif permet aussi au modèle de construire graduellement son ensemble d'apprentissage au cours de son entraînement, en interaction et guidé par une heuristique. L'apprentissage débute avec peu de données libellées, ensuite, le modèle sélectionne les exemples (non libellés) qu'il juge les plus "instructifs" pour les labéliser, puis de les incorporer à l'ensemble d'apprentissage. La particularité de l'apprentissage actif réside dans l'interaction du modèle avec son environnement. Contrairement à la stratégie "passive" où les exemples sont choisis avant l'apprentissage, de manière aléatoire, les stratégies "actives" permettent d'accélérer l'apprentissage en considérant d'abord les exemples les plus informatifs. Cette approche est particulièrement avantageuse lorsque les données sont coûteuses à acquérir et à labéliser avec la devise de bien choisir peu mais bien d'où l'intérêt de la stratégie à adopter.

Les approches d'apprentissage actif classiques se concentrent principalement sur les problèmes de classification unique, c'est-à-dire la classification binaire et la classification multi-classe. Dans le cas multi-label, chaque instance peut être associée à plusieurs labels simultanément, où la tâche de classification consiste à affecter un ensemble de labels pour chaque instance.

Pour évaluer la performance prédictive d'un classifieur multi-label, il existe un grand nombre de critères différents dans la littérature. La variété des critères d'évaluation est nécessaire pour fournir un aperçu global sur la performance prédictive de chaque classifieur. 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 (voir mesures d'évaluation au paragraphe 6 du chapitre 2).

Contrairement à la classification multi-classe et binaire, où le classifieur doit prédire seulement une sortie, la classification multi-label doit apprendre les associations entre les caractéristiques des instances d'entrées et plusieurs sorties à la fois. Chaque sortie indique si un certain label est pertinent pour l'échantillon de données ou pas, ainsi les algorithmes sont axés sur un ensemble de prédictions binaires. De nos jours, la classification multi-label est appliquée pour automatiser la suggestion des réponses [Charte et al., 2015], catégoriser des

documents de texte [Klimt and Yang, 2004], étiqueter des images [Duygulu et al., 2002], etc. Une introduction à la classification multi-label et un examen récent des techniques et des thèmes connexes peuvent être trouvés dans [Gibaja and Ventura, 2014-2015], respectivement et ont été déjà résumés au chapitre 3.

La plupart des tâches susmentionnées impliquent de travailler avec de datasets multi-label (*MLD*) ayant un nombre disparate de caractéristiques d'entrée, instances, labels, combinaisons de labels, etc. Certainement, certains de ces traits, comme le nombre des instances, déterminent dans une certaine mesure le temps nécessaire pour faire apprendre un classifieur. Au-delà de ce fait, il serait souhaitable de connaître à l'avance les difficultés que peuvent présenter des *MLD* et comment leur complexité peut affecter la performance du classifieur.

Une deuxième circonstance qui affecte potentiellement la performance des algorithmes de classification multi-label est la façon dont les datasets *MLD* sont partitionnés. Il existe des *MLD* contenant seulement quelques échantillons, parfois seulement un, en tant que représentants de labels rares. L'échantillonnage aléatoire, qui est la stratégie dominante utilisée dans le domaine multi-label, peut jeter ces quelques échantillons en entier en dehors de la partition d'entraînement ou de celle de test. Dans les deux cas, cela diminuera probablement la performance du classifieur.

On étudiera et on utilisera dans ce chapitre à caractère expérimental, comment la complexité des *MLD* et la stratégie d'échantillonnage influence les résultats de la classification. Pour ce faire, on introduira une nouvelle métrique de caractérisation, appelée *TCS* (Score de Complexité Théorique) qui permettra de connaître la complexité d'un *MLD* à l'avance, avant de l'utiliser pour faire apprendre un classifieur. Cette métrique est calculée à partir des traits *MLD* de base [Charte et al., 2016]. Ensuite, on procédera graduellement au prélèvement d'échantillons pour le partitionnement des datasets. On fera appel par la suite aux tests statistiques en utilisant des diagrammes critiques.

2. Caractéristiques des datasets multi-label et métriques associées

La nature propre des données à traiter par le classifieur implique un certain degré de complexité. Comment mesurer ce niveau de complexité strictement à partir des caractéristiques des données serait un objectif intéressant. Dans le même temps, la stratégie de partitionnement des données influence également les schémas d'échantillonnage que l'algorithme a à sa disposition pour entraîner le classifieur. Dans la classification multi-label, l'échantillonnage aléatoire est couramment utilisé pour accomplir cette tâche.

Avant de tenter de construire un modèle de classification pour résoudre un problème spécifique, il est important d'analyser les principales caractéristiques des données disponibles pour la tâche. Comprendre les traits internes des données permettra habituellement la sélection de meilleur algorithme et des paramètres adéquats pour rendre le modèle appris à la fois plus efficace et performant.

Mulan est une bibliothèque *Java* open-source pour l'apprentissage des datasets multi-label *MLD* (voir section 7, chapitre 2 : <http://mulan.sourceforge.net/>). L'ensemble des *MLD* comprend des exemples d'apprentissage d'une fonction cible qui a de multiples variables binaires. Cela signifie que chaque élément d'un *MLD* peut être membre de plusieurs catégories et annoté par de nombreux labels.

On s'intéresse en mode semi-supervisé, à une partie apprentissage peu volumineuse par rapport à celle non libellée, avec les bases de données qui sont représentées comme suit : *X* représentent les exemples ou instances d'entrée, les *Y* sont les labels. On divise la base de données en deux parties, la partie apprentissage (*X_App*, *Y_App*) et la partie Test (*X_Test*, *Y_Test*) (voir Figure 5.1).

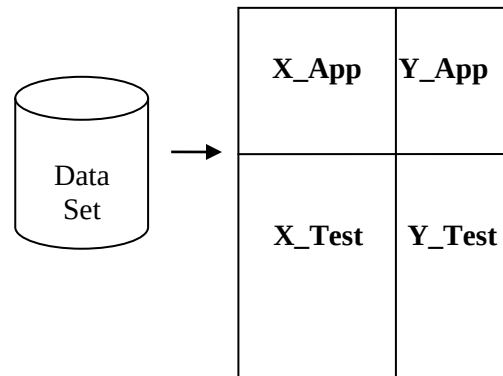


Figure 5.1– Répartition du dataset pour l'apprentissage semi-supervisé multi-label.

Pour la configuration expérimentale, les *MLD* ont été partitionnés suivant une stratégie 2×4. Cela signifie qu'il existe deux répétitions avec 4 répartition, et que pour chaque déroulement, 75% (3/4) des instances sont utilisés pour l'entraînement et 25% (1/4) pour le test. Par conséquent, un total de 8 essais est effectué pour chaque *MLD*. L'échantillonnage aléatoire a été utilisé pour sélectionner les instances dans chaque répartition. L'ensemble complet des répartition pour les six *MLD* mentionnés ci-dessus est disponible dans le livre référentiel de [Charte et al., 2016] à <https://github.com/fcharte/SM-MLC>.

A partir de chaque cycle de déroulement, un ensemble de prédictions est obtenu à partir du classifieur adopté. Celui-ci peut être évalué à l'aide de nombreuses mesures d'évaluation du rendement (elles ont été décrites auparavant dans la section 6 du chapitre 2), obtenant un ensemble de valeurs pour chaque métrique. Ces valeurs sont moyennées, obtenant les indicateurs moyens qui sont habituellement rapportés dans la plupart des articles, parfois avec leurs écarts standards. La table 5.2 montre ces valeurs pour les paramètres d'évaluation adoptés.

2.1 Caractéristiques des datasets multi-label

C'est en fait la nature de nombreux problèmes du monde réel tels que l'annotation sémantique d'images et de vidéo, la catégorisation des pages web, le marketing direct, la génomique fonctionnelle et la catégorisation de la musique en genres et les émotions. Une introduction sur l'exploitation et la fouille de données multi-label est fournie dans [Srivastava and Zane-Ulman, 2005].

Les six *MLD* choisis pour exécuter les différents algorithmes, tels *ML-kNN*, *SOM*, *SVM*, etc., sont répartis comme suit (voir Table 5.1) : deux d'entre eux sont du domaine du texte (*medical* et *tmc2007*), deux autres du domaine multimédia (*emotions* et *scene*), et les deux derniers viennent du domaine de la biologie (*genbase*, *yeast*).

- **A propos de la catégorisation du texte :**

La catégorisation des documents textuels en une ou plusieurs catégories est un besoin très courant. C'est la tâche à la base de la classification multi-label. C'est la raison de l'existence de nombreux datasets avec ce cas d'utilisation. Les études de cas mentionnées ci-dessous ont été utilisées dans une partie considérable de la littérature multi-label.

- **medical:** Les documents traités pour produire ce *MLD* sont des textes cliniques anonymes, en particulier le texte libre où les symptômes du patient sont décrits. Une partie du corpus total décrit dans [Crammer et al., 2007] a été utilisée pour générer le *MLD*, avec la transformée du texte en sac-de-mots (bag of words, BoW) par document. Les labels, au total 45, sont les codes de la Classification Internationale des Maladies, précisément les codes ICD-9-CM, <http://www.cdc.gov/nchs/icd/icd9cm.htm>.

- **tmc2007:** Ce dataset résulte de l'atelier SIAM Text Mining Workshop en 2007 [Srivastava and Zane-Ulman, 2005], <http://web.eecs.utk.edu/events/tmw07/>. Comme beaucoup d'autres datasets textuels, un sac de mots (BoW) booléen a été choisi comme un moyen de représenter les termes apparaissant dans les documents. Il s'agit de rapports sur la sécurité aérienne, dans lequel certains problèmes lors des vols ont été décrits. Le vocabulaire se compose originalement de 49 060 mots différents, utilisés comme caractéristiques d'entrée. Chaque rapport est étiqueté dans une ou plusieurs catégories à partir d'un ensemble de 22 qui sont les labels dans ce *MLD*. On fait référence à *tmc2007-500* dans le cas où on utilise seulement 500 attributs ou mots différents.

- **A propos de l'étiquetage des ressources multimédias :**

Bien que les ressources textuelles aient été les premières à exiger des mécanismes automatisés, récemment, le besoin d'étiqueter d'autres types de données, telles que les images, les sons, la musique et la vidéo, a été expérimenté avec une croissance élevée. Par contraste avec les études de cas énumérées dans la section précédente, dans lesquelles une représentation commune comme BoW (qu'elles contiennent des valeurs booléennes, des fréquences ou des valeurs TF-IDF), ceux-ci recourent à des modes de réalisation disparates.

- **emotions:** L'origine de ce jeu de données est l'étude réalisée dans [Wieczorkowska et al., 2006], dont l'objectif est d'identifier automatiquement les émotions produites par différentes chansons. Une centaine de chansons provenant de chacune de sept styles musicaux ont été prises en compte. Les auteurs ont utilisé l'outil logiciel décrit dans [Tzanetakis and Cook, 2002] pour extraire de chaque enregistrement un ensemble de caractéristiques rythmiques et un autre avec des caractéristiques de timbre. L'union de ces ensembles, après un processus de sélection d'entités, est utilisée comme attributs d'entrée. Les chansons étaient étiquetées par trois experts, en utilisant les six émotions principales du modèle émotionnel abstrait de Tellegen-Watson-Clark. Seules les chansons où les labels assignés coïncident ont été conservées, réduisant le nombre d'instances de l'original de 700 à 593.

- **scene:** Ce *MLD* est également lié à l'étiquetage d'image, spécifiquement à la classification de scènes. L'ensemble des images a été extrait du dataset Corel et certaines données personnelles par les auteurs [Boutell et al., 2004] ont également été incluses. Le *MLD* est composé de 400 images pour chaque concept principal, plage, coucher du soleil, champ, feuillage d'automne, montagne, et urbain. Par conséquent, six labels non exclusifs sont considérés. Les images sont transformées en CIE de l'espace colorimétrique de Luv, connu pour être perceptuellement uniforme, et segmentées en 49 blocs, en calculant pour chacun d'entre eux des valeurs telles que la moyenne et la variance. Le résultat est un vecteur de 294 fonctionnalités de valeur réelle dans chaque instance.

- **A propos de génétique / biologie :**

Il s'agit de la zone où les datasets sont moins accessibles au public, ce qui n'est surprenant de part leur complexité. Il existe deux *MLD*, l'un axé sur la prédiction de la classe des protéines et un autre pour classifier les gènes en fonction de leur expression fonctionnelle.

- **genbase:** Les auteurs de [Diplaris et al., 2005] ont produit cette information de compilation *MLD* pour 662 différentes protéines. Le numéro d'accès du Prosite a été utilisé pour identifier les 1 185 modèles du motif et des profils ont été utilisés comme caractéristiques d'entrée, <http://prosite.expasy.org/prosite.html>. Tous sont nominaux, ne prenant que les valeurs Oui ou Non. De cette façon, les motifs et profils présents dans chaque protéine sont indiqués. 27 classes de protéines différentes sont considérées, chaque protéine étant associée avec un ou plusieurs d'entre elles. Les identificateurs de classe de protéine PDOC sont utilisés comme des noms de labels. Autre chose à prendre en compte lors de l'utilisation de ce *MLD* est la présence d'une caractéristique supplémentaire, la première, qui identifie de manière unique chaque instance.

- **yeast:** Dans ce cas [Elisseff and Weston, 2001], l'objectif a été de prédire l'expression fonctionnelle d'un ensemble de gènes. Les caractéristiques d'entrée pour chaque gène proviennent de l'expression microarray des données, avec un vecteur de 103 valeurs réelles par instance. Un sous-ensemble de 14 classes fonctionnelles, dont l'origine est la base complète du génome de levure, sont sélectionnées et utilisées comme labels. Puisque chaque gène peut exprimer plus d'une fonction à la fois, en fait c'est la situation habituelle, le résultat est un ensemble de données à caractère multi-label,

<http://www.ncbi.nlm.nih.gov/pubmed/15608217>.

2.2 Les mesures d'évaluation adoptées

Pour l'évaluation des performances des classifieurs, la sortie de tout classifieur multi-label se compose du jeu de labels prédits pour chaque exemple de test. Lorsqu'on travaille dans le scénario traditionnel, avec une seule classe comme sortie, la prédiction peut seulement être correcte ou erronée. Une prédiction multi-label, par contre, peut être entièrement correcte, partiellement correcte / erronée (à des degrés différents), ou totalement erronée. Appliquer les mêmes paramètres utilisés dans la classification traditionnelle est possible, mais habituellement elle est excessivement restrictive.

C'est la raison pour laquelle on utilise des mesures d'évaluation spécifiques permettant de prendre en considération les cas entre les deux extrêmes. Actuellement, plus de vingt mesures de performance distinctes ont été définies dans la littérature (au chapitre 2, section 6, on a énuméré les seize mesures les plus usitées), et certaines d'entre elles sont tout à fait spécifiques visant à la classification hiérarchique multi-label. On résume ces métriques en donnant les principales d'entre-elles au sens de notre application. Comme déjà mentionné au paragraphe 6 du chapitre 2, toutes ces métriques d'évaluation multi-label peuvent être regroupées selon deux critères, On résume:

- *Comment la prédiction est calculée:* une mesure peut être faite par instance ou par label, donnant ainsi deux groupes de métriques différents:
 - *Mesures basées sur l'exemple:* ces métriques sont calculées séparément pour chaque instance, puis moyennées par division entre le nombre des échantillons.
 - *Mesures basées sur le label:* contrairement au groupe précédent, les statistiques basées sur les labels sont calculées indépendamment pour chaque label avant d'être moyennées. Pour ce faire, deux stratégies différentes peuvent être appliquées:
 - *Macro-moyennage:* la métrique est calculée individuellement pour chaque label et le résultat est divisé en moyenne par le nombre de label Q .
 - *Micro-moyennage :* les compteurs de présence et d'absence pour chaque label sont d'abord agrégés, puis la métrique n'est calculée qu'une seule fois.
- *Comment le résultat est fourni:* la sortie produite par un classifieur multi-label peut être une bipartition binaire de labels ou un classement de labels. Certains d'entre eux fournissent les deux résultats à la fois :
 - *Bipartition binaire:* une bipartition binaire est un vecteur de 0's et 1's indiquant lesquelles des labels appartenant au *MLD* sont pertinents pour l'échantillon traité. Il existe des métriques qui opèrent sur ces bipartitions, en utilisant les compteurs de Positifs, Vrais Négatifs, Faux Positifs et Faux Négatifs.
 - *Classement des labels:* la sortie est une liste de labels classés selon une certaine mesure de pertinence. Une bipartition binaire peut être obtenue à partir d'un classement de labels en appliquant un seuil, généralement donné par le classifieur lui-même. Toutefois, il existe des métriques qui fonctionnent avec le classement brut pour calculer la mesure, au lieu d'utiliser des compteurs de bonnes et de mauvaises prédictions.

Dans les paragraphes suivants, les métriques basées sur l'exemple sont utilisées et par la suite micro-F1 (voir l'équation (2.12) du chapitre 2, section 6), métrique basée labels sera elle aussi utilisée dans l'apprentissage *TSVM* actif. Par la suite, chaque description de la métrique est complétée par les résultats obtenus par l'expérimentation avec *ML-kNN* dans la section suivante.

Parmi les métriques de performance basées sur l'exemple adoptées en toute circonstance et qui sont d'abord évaluées par chaque instance et puis calculées en fonction du nombre d'instances considérées et par conséquent, le même poids est attribué à chaque instance au niveau du score final, qu'elles contiennent de labels fréquents ou rares. Ces critères ont été

déjà énumérés au chapitre 2, section 6, nous donnons ici de plus amples renseignements pour certaines mesures d'entre-elles qu'on a jugés utiles à savoir *Hamming-Loss*, *accuracy* et *F-mesure*.

- *Hamming-Loss* : La perte de Hamming est probablement la métrique de performance la plus couramment utilisée dans la classification multi-label. Cela n'est pas surprenant, car il est facile à calculer comme on le voit dans l'équation (2.1) du chapitre 2. Puisque le compteur d'erreurs est divisé par le nombre des labels, cette métrique entraîne des évaluations différentes pour la même quantité d'erreurs lorsqu'elles sont utilisées avec des datasets multi-label ayant des longueurs différentes de labels.

C'est la principale raison de la faible valeur de *Hamming-Loss* pour *genbase* (0.0007) par rapport à *emotions* (0.3564) ou à *scene* (0.1134) (voir table 5.2). Le premier dataset a un grand nombre de labels (27), tandis que les autres ont seulement six. Par conséquent, cette métrique est un indicateur d'erreurs commises par le classifieur proportionnel à la longueur du labelset. Nous pouvons comparer les résultats de *emotions* et de *scene*, les deux ont le même nombre de labels, et concluons que ML-kNN s'est mieux comporté avec ce dernier (valeur moindre) que le premier dataset.

- *Accuracy* : Dans le champ multi-label, l'exactitude est définie comme la proportion entre le nombre de labels correctement prédits et le nombre total de labels actifs, à la fois dans les labelsets réel et prédit. La mesure est calculée pour chaque instance et puis moyennée, comme toutes les métriques basées sur l'exemple (voir l'équation (2.2) du chapitre 2).

L'exactitude pour *genbase* est très élevée (0.9916), due principalement à la raison expliquée précédemment. Comme le montre la table 5.2, les valeurs pour *scene* (0.6710) et *tmc2007* (0.6593) sont presque similaires, bien qu'avec un léger avantage au premier dataset ; de même pour *yeast* (0.5290) et *medical* (0.5362). L'exactitude obtenue peut être considérée comme bonne dans le cas de *medical* et encore plus pour *tmc2007*. Il faut se rappeler que ce dernier MLD a le TCS le plus élevé des six cas étudiés (voir table 5.1) et par conséquent, le fait d'obtenir un bon rendement de classification attendue prouve la performance du classifieur ML-kNN.

- *F-mesure* : L'utilisation conjointe de la *precision* et du *recall* (rappel) est si fréquente dans la recherche d'information (IR) qu'une métrique *F-mesure* les combinant est définie, comme on l'a vue au chapitre 2, l'équation (2.5). Elle est calculée comme la moyenne harmonique des deux précédentes. C'est une mesure de pondération du nombre de labels pertinents qui sont prédits et du nombre de labels prédits qui sont pertinents.

Avec *emotions* (0.5465) et *medical* (0.5716), on peut voir que ces valeurs signifient que pour ces MLD, une proportion élevée des labels inclus dans la prédiction sont des labels pertinents, mais qu'il existe de nombreux autres labels vrais qui ne sont pas prédits par le classifieur. En regardant les valeurs de la mesure F, la même corrélation entre la complexité théorique (valeur TCS) de chaque MLD et l'évaluation de la performance de classification peut être déduite. Pour le *tmc2007* (0.7186) dont le TCS est le plus élevé (16.372), constitue

un fait plus important avec son cas de 1172 labelsets et 500 attributs. *genbase* a la valeur la plus élevée (0.9941) et qui dépasse l'unité dans le cas des classifieurs topologiques.

2.3 Score de Complexité Théorique (TCS)

Avant de tenter de construire un modèle de classification pour résoudre un problème spécifique, il est important d'analyser les principales caractéristiques des données disponibles pour la tâche spécifiée. Comprendre les traits internes des données permettra habituellement la sélection de meilleur algorithme et des paramètres, etc. Dans la section 6 du chapitre 2, bon nombre de métriques de caractérisation disponibles ont été définies, en fournissant leurs expressions mathématiques, et en détaillant leur utilité.

On utilisera principalement les trois mesures connues Hamming Loss, Accuracy et F-mesure qui seront associées aux datasets DML et commentées selon la métrique TCS, reflétant le score de complexité théorique [Charte et al., 2016], établi comme suit :

D étant le nombre de caractéristiques d'entrée ou attributs, Q le nombre de labels et $Lset$ est le nombre de jeux de labels uniques et distincts.

$$TCS (MLDataset) = \log (D \times Q \times Lset) \quad (5.1)$$

Théoriquement, plus l'entrée et les espaces de sortie sont élevés, plus le modèle généré sera complexe, ce qui le rendra difficile à ajuster correctement. En travaillant dans le champ multi-label, on peut s'appuyant sur les valeurs TCS pré-calculées des MLD, il est facile de les trier selon leur complexité théorique, comme le montre la table 5.1. Ces informations peuvent être utiles pour répertorier par ordre de complexité des données les six datasets considérées.

Rank	Dataset (domaine)	TCS	Attributs D	Labels Q	l_C	l_D	Labelsets
1	Emotions (audio-musique)	9.364	72	6	1.868	0.485	27
2	Scene (images)	10.183	294	6	1.074	0.179	15
3	Yeast (biologie-texte)	12.562	103	14	4.237	0.303	198
4	Genbase (biologie-protéines)	13.840	1 186	27	1.252	0.046	32
5	Medical (texte)	15.629	1 449	45	1.245	0.028	94
6	tmc2007-500 (texte)	16.372	500	22	2.158	0.098	1 172

Table 5.1 – Description des benchmark de référence en termes du domaine d'application (*domaine*), le nombre de caractéristiques (features) ou d'attributs (D), le nombre total d'étiquettes (Q), la cardinalité de l'étiquette (l_C) et sa densité (l_D) ainsi que le labelset associé. Les problèmes sont ordonnés selon le score de leur complexité théorique TCS.

La principale différence entre la classification traditionnelle et multi-label provient du fait que dans ce dernier, chaque cas est associé à un ensemble de labels. Ce sont les premières métriques spécifiques conçues pour les MLD, dont le but est d'évaluer la polyvalence des labels relatifs aux données, c'est-à-dire de déterminer dans quelle mesure les échantillons dans un dataset MLD contiennent plus d'un label.

- *Cardinalité des labels*: Une manière évidente de calculer une telle mesure consiste à compter le nombre de labels pertinents pour chaque instance du dataset, puis de faire la moyenne de la somme pour avoir le nombre moyen de labels par instance. Cette simple métrique a été introduite dans [Tsoumakas and Katakis, 2007] comme la cardinalité du label ou simplement l_c (voir équation 5.2).

$$l_c = \frac{1}{n} \sum_{i=1}^n |Y_i| \quad (5.2)$$

Dans ce contexte, N désigne le nombre d'instances dans le dataset MLD, Y_i le labelset du $i^{\text{ème}}$ instance, et Q le nombre total de labels considérés dans ce dataset. Plus l_c est élevé est plus le nombre de labels actifs par instance est élevé. En conséquence, un MLD avec une valeur faible de cardinalité, près de 1.0, signifierait que la plupart de ses échantillons ont seulement un label pertinent. Par conséquent, il s'agirait d'un dataset à faible nature multi-libellée. Par contre, les valeurs élevées de l_c indiquent que les données sont véritablement multi-libellées. Comme une règle générale, les valeurs élevées de l_c sont liées à des MLD qui ont de grands ensembles de labels, mais le contraire n'est pas toujours vrai.

- *Densité des labels* : Puisque l_c est une métrique influencée par la taille de l'ensemble des labels utilisés par chaque MLD, et il est exprimé en utilisant le nombre de labels comme unité de mesure, une version (voir équation 5.3) a également été proposée. En divisant l_c par le nombre des labels présents dans le MLD, une métrique sans dimension, connue sous le nom de densité de labels l_D , est obtenue. Habituellement, une valeur élevée de l_D indique que les labels dans le MLD sont bien représentés dans chaque instance. En revanche, les valeurs basses de l_D indiquent une plus grande dispersion, avec seulement un petit sous-ensemble des labels présents dans la plupart des cas.

$$l_D = \frac{1}{k} \frac{1}{n} \sum_{i=1}^n |Y_i| \quad (5.3)$$

La table 5.1 donne les détails des datasets (voir aussi la table 2.2 du chapitre 2) tels que le nombre d'attributs, le nombre de classes et leur densité de labels. On constate que *genbase* ($l_D = 0,046$) et *scene* ($l_D = 0,179$) sont des datasets multi-label assez clairsemés avec moins de 1,5 labels par exemple en moyenne. Par contre, le dataset *yeast* (données de levures) est plus dense ($l_D = 0,303$) avec plus de 4 labels par exemple en moyenne.

Selon la mesure de leur TCS, les datasets *emotions* et la *scene*, classés en positions 1 et 2 de la table 5.1, seraient les cas les plus faciles à explorer. Un peu plus difficile serait *genbase* (4ème), suivi par *medical* (5ème) et enfin *TMC2007-500* (6ème) qui, théoriquement, serait le MLD le plus difficile dans cette collection.

3. Classifieurs de base multi-label

Dans les paragraphes suivants, plusieurs dizaines d'algorithmes de classification multi-label seront appliqués et qui ont été déjà décrits aux chapitre 3 et 4 et dont certains d'entre eux seront encore détaillés selon les circonstances et seront testés expérimentalement. Par conséquent, comment une expérience a été menée, et la manière dont les résultats peuvent être évalués pour tester la performance des algorithmes, sont des aspects fondamentaux. Une fois que les datasets multi-labels *MLD* disponibles et leurs traits principaux sont connus, on commence par l'introduction de l'exécution de l'algorithme fondamental *MLC*, basé sur *kNN* pour traiter les cas des six *MLD*.

Les mesures prédictives de l'évaluation de la performance en multi-label doivent tenir compte de la présence de sorties multiples, en tenant compte de l'existence de prédictions qui sont partiellement correctes ou erronées. Comme cela sera exposé, ces métriques peuvent être regroupées pour nos expériences, nous avons appliqué trois différentes méthodes sur les six datasets *MLD* considérés, à savoir :

- 1) Pour ML-*kNN* : $k=1$ (de base) pour Binary Relevance, ensuite pour différentes valeurs de k selon le dataset mis en jeu.
- 2) Pour les cartes topologiques : *SOM*, *SOM-Y* et *SOM-mixte* considérant à la fois les instances et leurs labels associés, c'est une première tentative pour ces cartes en mode multi-label. *SOM-Y* permet d'associer à chaque prototype des labels, en prenant les exemples les plus proches au prototype et procéder par la suite à un vote majoritaire en appliquant aussi différents masques sur la partie des labels pour voir leurs comportements.

Il ressort de la table 5.2, que chaque méthode a réagi différemment sur chacune des mesures et pour chaque dataset considéré, le but étant d'enrichir nos résultats afin d'avoir par la suite une meilleure interprétation.

- *BR-kNN* : Au chapitre 3, section 2 relative à l'auto-apprentissage, on a donné l'algorithme self-training *kNN* ($k=1$), il s'articule autour d'un classifieur à 1 voisin le plus proche, on arrive à affecter à l'instance sélectionnée, le label de son voisin le plus proche en termes de distance.

La méthode Binary Relevance (*BR*) est l'une des méthodes d'apprentissage par transformation. Les algorithmes traditionnels sont incapables de traiter l'ensemble des instances multi-labels, car ces derniers ont été conçus pour traiter le cas uni-label. La simple solution est de transformer la base de données originale en un ensemble d'instances où chaque ensemble contient tous les attributs, et seulement un des labels sera à prédire, la méthode est connue sous le nom Binary Relevance. La plus simple stratégie pour les problèmes de transformation est d'utiliser la stratégie un contre tous pour convertir le problème multi-labels en un problème de classification binaire [Trohidis et al., 2008]. Il en ressort, comme on l'a déjà noté, que *BR* est la méthode la plus populaire et la plus simple quoiqu'elle ne tient pas de la corrélation des labels (voir section 3 du chapitre 2).

- *ML-kNN* : [Zhang and Zhou, 2005] est une adaptation de l'algorithme d'apprentissage paresseux *kNN* pour les données multi-label. En fait, cette méthode suit le même paradigme et

utilise essentiellement l'algorithme k NN indépendamment pour chaque label l : il trouve les k exemples les plus proches de l'instance de test et considère ceux qui sont étiquetés au moins avec l comme positifs et le reste comme négatifs. Ce qui différencie principalement cette méthode de l'application de l'algorithme k NN original au problème transformé est l'utilisation de probabilités antérieures. ML- k NN a également la capacité de produire un classement des labels comme une sortie.

Pour un nouvel exemple $\mathbf{x}_i$, ML- k NN calcule ses k plus proches voisins puis mesure la fréquence de chaque label dans ce voisinage. Néanmoins, comme toute approche de type k NN, le temps de prédiction croît linéairement avec la taille de l'ensemble d'apprentissage.

Exemple de fonctionnement pour $k = 5$:

$\mathbf{X} \rightarrow \begin{matrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{matrix} \rightarrow \left[\frac{4}{5} \frac{3}{5} \frac{0}{5} \frac{1}{5} \right] \rightarrow [0.8 \ 0.6 \ 0.0 \ 0.2] \rightarrow \text{Seuillage (0.5)} \rightarrow [1 \ 1 \ 0 \ 0] \text{ Vecteur prédit.}$

En figure 5.2, sont donnés pour le dataset *medical*, les résultats pour les deux critères *accuracy* et *Hamming Loss* dans le cas du ML- k NN pour $k=1$, c'est un dataset avec un TCS=15.629 assez difficile à interpréter et à analyser, comme la plupart des datasets textuels.

Bases	Mesures	ML-kNN $k=1$	ML-kNN	SOM	SOM-mixte	SOM-Y	BR-TSVM
Emotions	Ham loss ↓	0.3267	0.3564	0.3432	0.3036	0.3028	0.2343
	Acc ↑	0.3908	0.4372	0.3432	0.2554	0.2409	0.5516
	F-Mesure ↑	0.4635	0.5465	0.5334	0.6345	0.6596	0.6500
		$k=2$					$C=2, ord=12$
Scene	Ham loss ↓	0.1249	0.1134	0.1283	0.1010	0.1099	0.0877
	Acc ↑	0.6377	0.6710	0.5782	0.6626	0.6484	0.6837
	F-Mesure ↑	0.6530	0.7344	0.8273	0.8331	0.7712	0.9610
		$k=4$					$C=1, \sigma=0.25$
Yeast	Ham loss ↓	0.2447	0.2045	0.2321	0.2165	0.2489	0.1929
	Acc ↑	0.4749	0.5290	0.4625	0.4967	0.4529	0.5218
	F-Mesure ↑	0.5722	0.6333	0.5950	0.6085	0.5506	0.6266
		$k=10$					$C=2, \sigma=0.5$
Genbase	Ham loss ↓	0.0009	0.0007	0.0056	0.0032	0.0032	0.0007
	Acc ↑	0.9883	0.9911	0.9343	0.9719	0.9686	0.9916
	F-Mesure ↑	0.9923	0.9901	1.0083	0.9809	0.9951	0.9941
		$k=3$					$C=1, \sigma=0.75$
Medical	Ham loss ↓	0.0228	0.0228	0.0249	0.0210	0.0207	0.0112
	Acc ↑	0.5362	0.5362	0.3625	0.3673	0.3823	0.7395
	F-Mesure ↑	0.5716	0.5716	1.0082	1.1963	1.1709	0.9318
		$k=1$					$C=3, ord=1$
TMC2007	Ham loss ↓	0.0541	0.0541	0.0714	0.0684	0.0690	0.1958
	Acc ↑	0.6593	0.6593	0.4836	0.4861	0.4995	0.5911
	F-Mesure ↑	0.7186	0.7186	0.6358	0.6379	0.6206	0.9814
		$k=1$					$C=1, \sigma=0.75$

Table 5.2– Les résultats obtenus par les différentes méthodes sur les six datasets *MLD*.

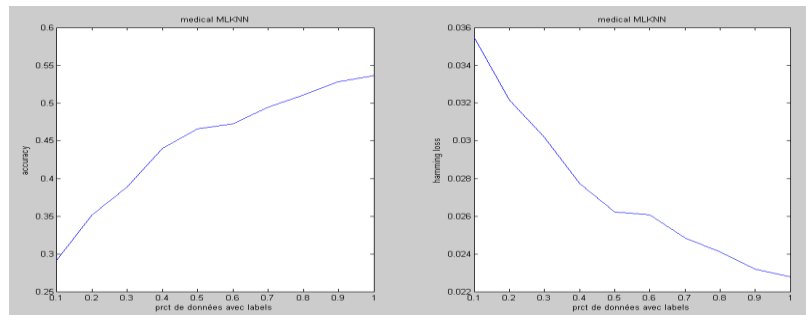


Figure 5.2 – Exemple de résultats ML- k NN obtenus pour la base *medical* (accuracy, Hamming loss)

On remarque dans la table 5.2 que les méthodes ML- k NN et *SOM-mixte* donnent de bons résultats pour les trois mesures (*Ham loss*, *Acc* et *F-Mesure*). *SOM-mixte* est meilleure que *SOM*, et en concurrence avec *SOM-Y* qui elle aussi, a prouvé son efficacité. Par conséquent, *BR-TSVM* donne globalement de meilleurs résultats (en gras dans le tableau) que toutes les autres méthodes.

	<i>emotions</i>	<i>scene</i>	<i>Genbase</i>
HammingLoss ↓	0.1940	0.0869	0.0048
Accuracy ↑	0.5391	0.6667	0.9440
F-measure ↑	0.7776	0.9593	0.9776

La flèche à droite de chaque métrique indique si les valeurs inférieures sont meilleures (↓) ou l'inverse (↑).

Table 5.3 – Résultats de classification obtenus par ML- k NN pour les trois principales métriques relevés de [Herrera et al., 2016] à titre comparatif.

• **SVM Transductif** : Dans l'apprentissage semi-supervisé, nous devons toujours envisager la possibilité qu'un modèle statistique précis conduira à des gains significatifs à partir de données non libellées. De ce fait, nous devrions chercher le modèle "correct" chaque fois que possible. En fait, la recherche en littérature a décrit des situations où un classifieur à structure fixe fonctionne mal, alors que les modèles de recherche flexibles peuvent mener à d'excellents classifieurs (Bruce, 2001; Cohen et al., 2003-2004). En particulier, Cohen et al. (2004) ont discuté et comparé différentes stratégies de recherche de modèles avec des données libellées et non libellées. Contrairement à l'apprentissage d'une règle de prédiction générale, V. Vapnik a proposé le cadre d'apprentissage transducteur où les prédictions ne sont faites qu'à un nombre fixe de connaissances connues d'instances de test. Cela permet à l'algorithme d'apprentissage d'exploiter l'emplacement des instances, ce qui en fait un type particulier de problème d'apprentissage semi-supervisé.

Les *TSVM* transductives mettent en œuvre l'idée d'apprentissage transductif en incluant des instances test dans le calcul de la marge. Elles exploitent la structure géométrique (cluster) ou les vecteurs caractéristiques des exemples de test, ce qui les rend un type particulier de méthode d'apprentissage semi-supervisé. En particulier, les *TSVM* estiment que l'étiquetage des exemples de test maximise la marge conjointement sur l'entraînement et les données d'essai. Intuitivement, cela produit l'étiquetage des exemples de test de sorte que les limites de classes suivent les limites des clusters.

Les résultats empiriques suggèrent que les *TSVM* sont bien adaptées pour la classification de texte et plusieurs autres (typiquement de haute dimension) problèmes d'apprentissage, montrant souvent de gros gains d'exactitude pour les grands ensembles de test. Cependant, sur certains problèmes, le *TSVM* effectue à peu près l'équivalent d'une *SVM* inductive, ou parfois même pire. En partie, l'échec de certaines tâches peut être dû à la difficulté de trouver l'optimum du problème d'optimisation de *TSVM*.

En général, l'entraînement utilisant seulement un sous-ensemble d'échantillons ne fonctionnera pas aussi bien comme dans le cas de l'ensemble des données. Choisir avec soin le sous-ensemble D_S peut aider à limiter cette perte de performance. Même si la sélection aléatoire est certainement la manière la plus simple de choisir les instances en D_S , la *TSVM* a deux inconvénients principaux:

Elle ne peut pas sélectionner des instances dans certaines régions de l'espace, ce qui entraîne une faible approximation dans ces régions.

- Elle peut choisir des instances ou points inintéressants: la région près de la surface de décision est celle où nous sommes plus susceptibles de faire des erreurs en attribuant le mauvais label. Donc, nous aimerions avoir autant de points que possible en S dans cette région, alors que nous n'avons pas besoin de points qui sont loin de cette surface.

- En conséquence, il vaut la peine d'envisager des schémas de sélection de sous-ensembles plus élaborés, pour réfléchir au problème de l'apprentissage à partir de données libellées et non libellées. De nombreuses approches ont été adoptées, nous procéderons graduellement par petites doses des instances non libellées à libeller. Par la suite, dans la section suivante, en mode apprentissage actif, d'autres critères tels que *BinMin*, *MMC* et celui d'incertitude seront utilisés pour le choix des instances à libeller.

L'idée de l'approche transductive proposée par Vapnik est de prendre en compte les données non-libellées pour induire une fonction générale. Cette fonction devrait non seulement minimiser le risque sur l'ensemble d'apprentissage mais aussi sur l'ensemble des données de test. On donne ci-joint, une procédure simplifiée relative à *TSVM* :

Entrées :

D_L : données avec labels P_L

D_U : données sans labels

D_T : données de Test P_T

$SVM = \text{Train}(D_L, P_L)$

$t=0$;

Faire

$P_U^t = \text{Test}(SVM, D_U)$

$SVM = \text{Train}(D_L \cup D_U, P_L \cup P_U^t)$

$t=t+1$;

Jusqu'à ($P_U^{t-1} = P_U^t$)

$P_S = \text{Test}(SVM, D_T)$

Calculer les mesures entre (P_T et P_S)

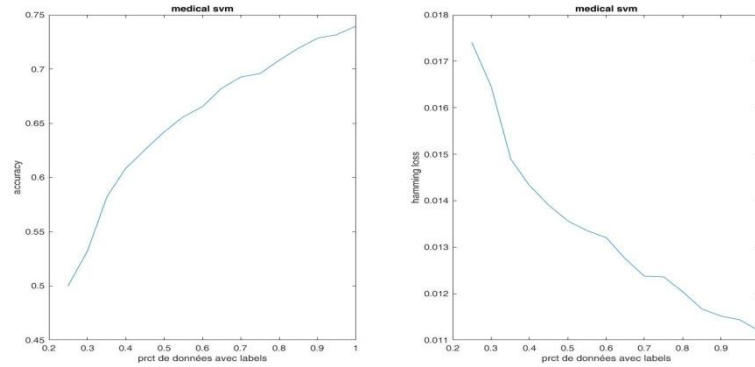


Figure 5.3— Exemple de résultats obtenus pour la base *medical* (accuracy, Hamming loss) pour TSVM

Les résultats obtenus en figure 5.3 pour la TSVM se révèlent meilleurs que ceux de la figure 5.2 pour le cas de Ml-kNN pour le même dataset *medical*.

Plusieurs essais ont été tentés prenant en compte le pourcentage des données libellées rajoutées, sachant qu'un classifieur est d'autant meilleur en termes de performances qu'il fait appel le moins possible à ces données relevant de D_U , d'où on procède généralement par prélèvement à petite dose (représentant le coût de l'étiquetage):

$$D_{\text{Train}} = D_L \cup D_U$$

Expérience 1 : $D_L \leftarrow 10\%$ des données

Expérience 2 : $D_L \leftarrow 20\%$ des données

...

Expérience 10 : $D_L \leftarrow 100\%$ des données

On aimerait bien que ce prélèvement (pourcentage rajouté) à partir de D_U soit aussi informatif que possible afin d'aider le classifieur à mieux prédire, d'où l'intérêt de l'apprentissage actif combiné à TSVM pour une meilleure sélection à partir du sous-ensemble

D_S , d'un sous-ensemble D_S^* plus optimal renfermant des instances semi-libellées (de D_U) et

transductives (de $D_S \subset D_U$) plus informatives: $D_U \rightarrow D_S \rightarrow D_S^*$ (voir algorithme 1 du chap.4).

Nous présentons ici les résultats obtenus selon le classificateur SVM et pour la première fois, selon le classificateur SOM en adoptant l'apprentissage actif basé sur le pool qui est activement utilisé dans la littérature. Le gain obtenu peut être quantifié en désignant les données minimales choisies par chacune des stratégies adoptées. En d'autres termes, dans l'apprentissage actif, la stratégie gagnante est celle qui donne de meilleurs résultats avec le moins de nombre possible de données supplémentaires.

Dans les expériences présentes, nous avons utilisé la version par lot des algorithmes pour faire apprendre les classifieurs. Toutes les méthodes et les algorithmes d'apprentissage actifs ont été implémentés à l'aide des fonctions Matlab (R2013b).

4. Résultats de TSVM actif en multi-label

Les TSVM transductifs mettent en œuvre l'idée d'apprentissage transductif en incluant des instances de test dans le calcul de la marge. En particulier, les TSVM considèrent que

l'étiquetage des exemples de test maximise la marge conjointement sur les données de l'apprentissage et les données de test. Intuitivement, cela produit l'étiquetage des échantillons de test de sorte que les limites de classe suivent les limites des clusters [Liu et al., 2016].

Au chapitre précédant, section 4, on a présenté la stratégie adoptée pour l'optimisation de l'apprentissage actif en multi-label pour la sélection d'échantillons avec SVM. Il s'agissait de en plus de la méthode évidente aléatoire (*random*), des deux stratégies d'échantillonnage d'incertitude multi-label, celle de BinMin de [Brinker, 2006] et de l'estimation de la réduction de perte avec confiance maximale (*MMC*) de [Yang et al., 2009].

La stratégie BinMin utilise une approche unique pour la classification multi-label et k *ML-SVM* classifieurs binaires f^i associés à la classe i et basés sur les données d'apprentissage. L'exemple x non libellé optimal est choisi parmi le label le plus incertain, selon:

$$\arg \min_x \min_{i=1, \dots, k} |f^i(x)| \quad (5.4)$$

Pour la stratégie *MMC*, la réduction de perte prévue pour le vecteur $\hat{y}$ le plus confiant pour les données non libellées x , est alors estimée comme suit:

$$\text{score}(x) = \sum_{i=1}^k \left(\frac{1 - \hat{y}^i f^i(x)}{2} \right) \quad (5.5)$$

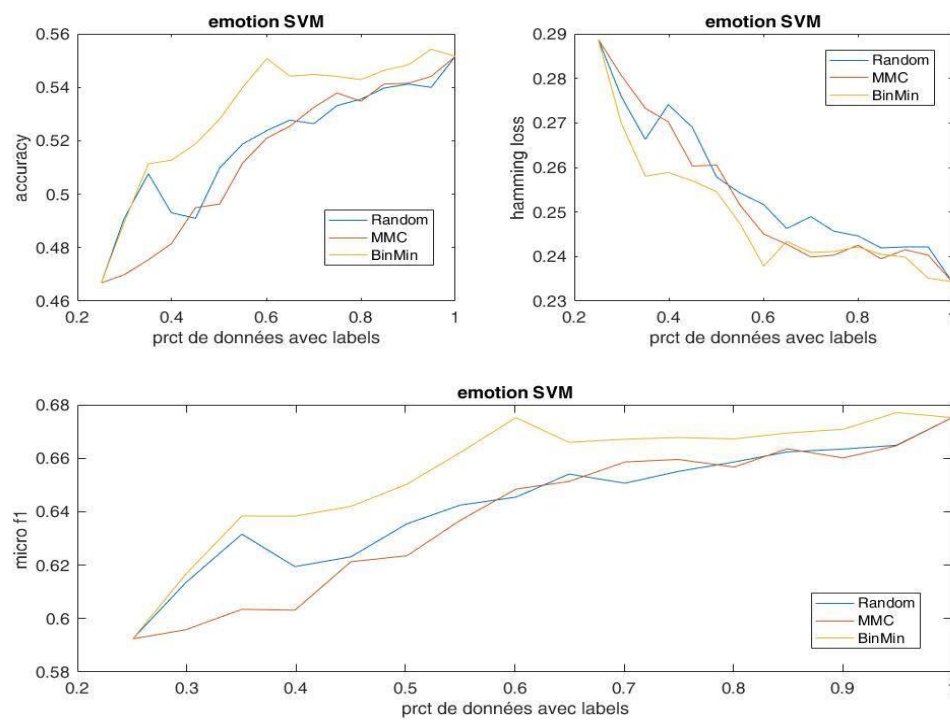
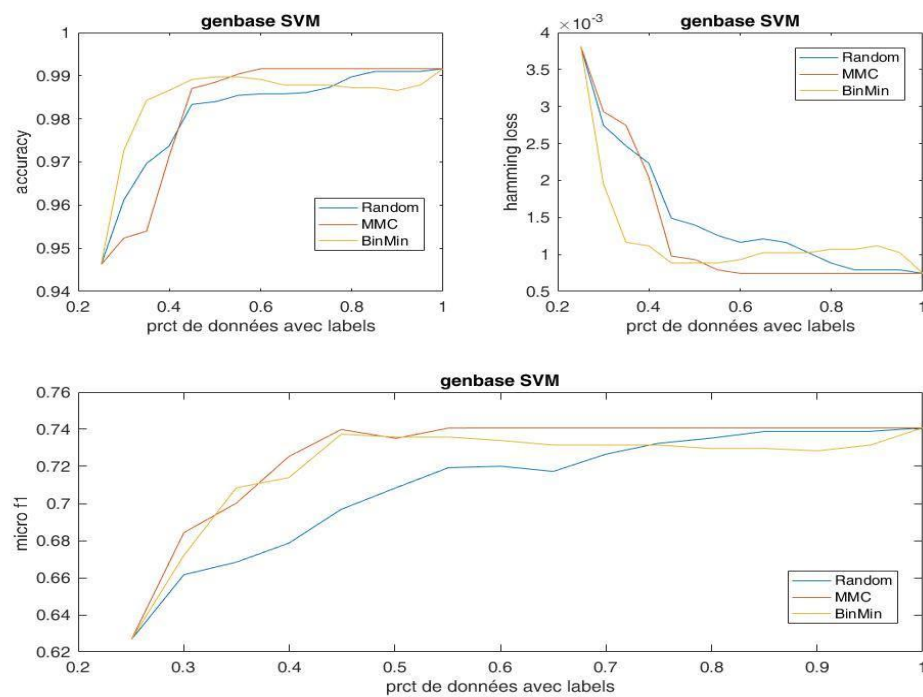
Ensuite, un ensemble sélectionné avec les scores les plus importants est rajouté pour recycler le classificateur multi-label.

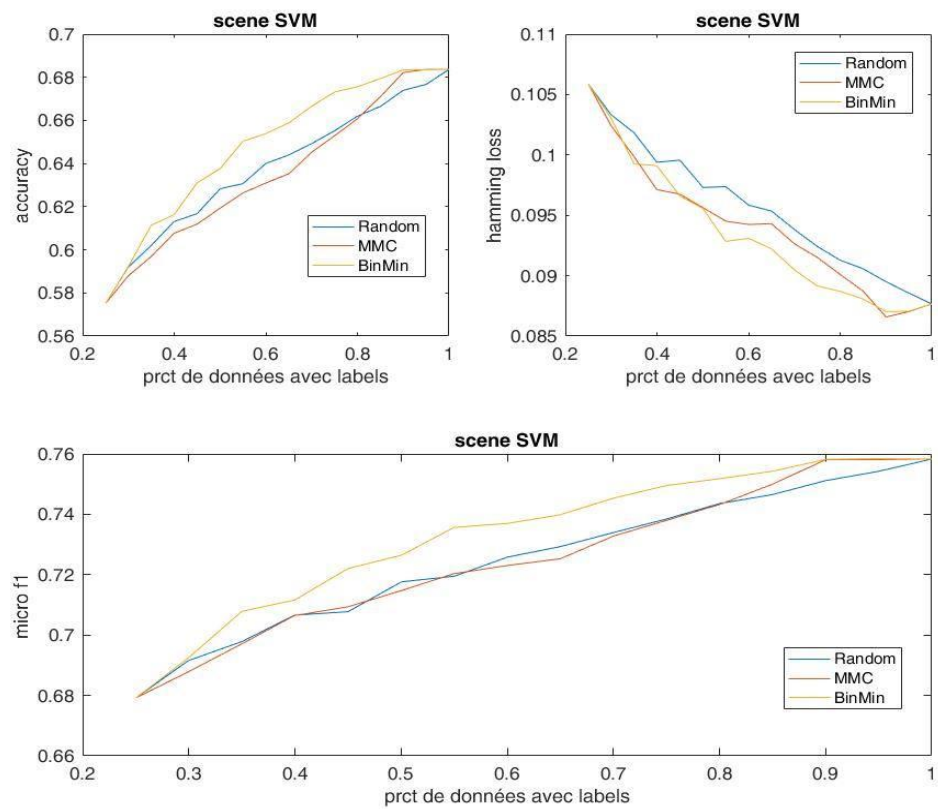
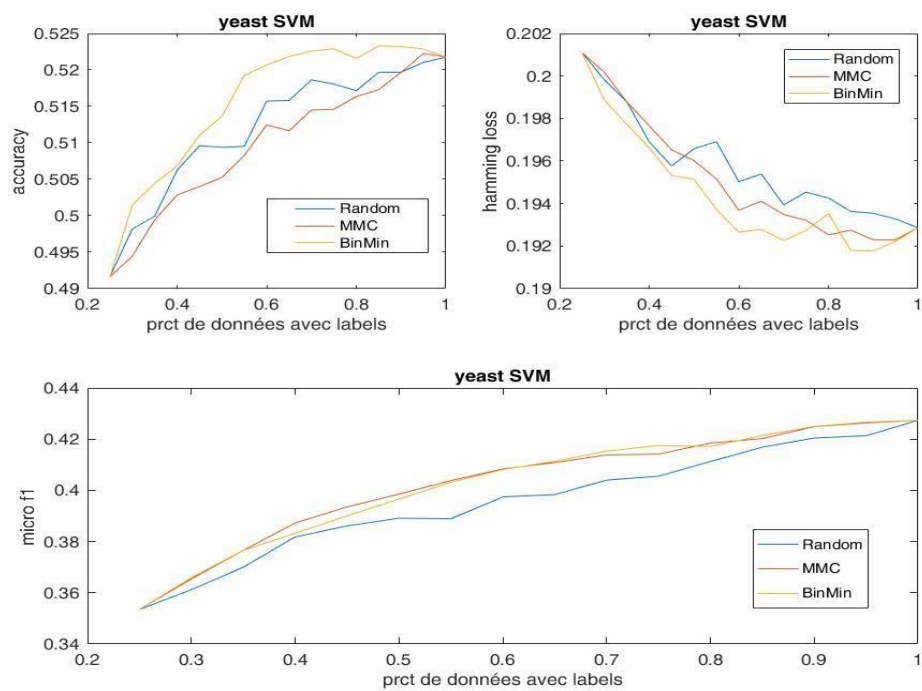
Pour les deux cas, les points les plus proches de la limite de décision sont choisis, c'est-à-dire la stratégie du pire des cas pour sélectionner les instances des outsiders [Cherman et al., 2016].

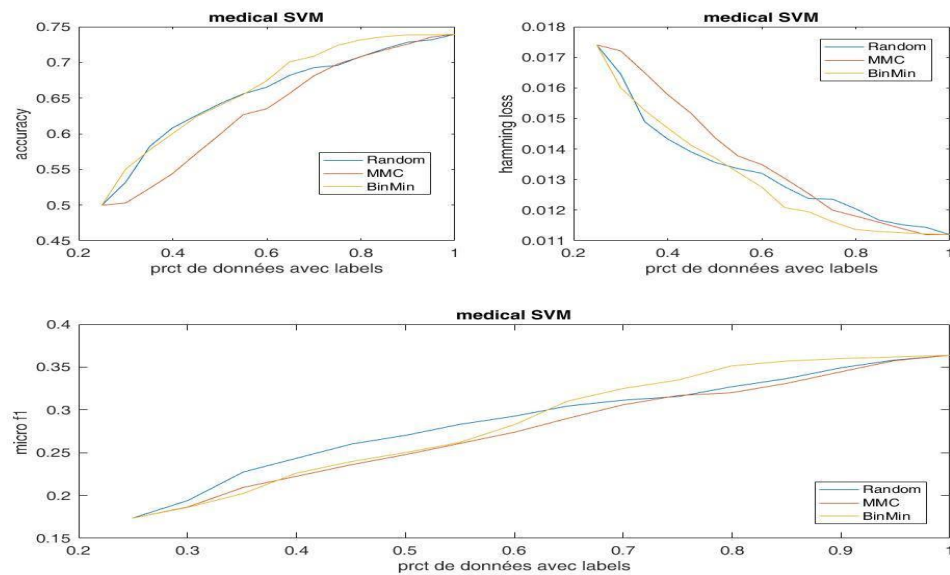
Nous avons commencé à faire apprendre le classifieur avec 20% de la base de données. Les résultats montrent clairement que *BinMin* a donné de meilleurs taux par rapport à *Random* et *MMC* sur les six *MLDatasets* (*yeast*, *emotion*, *scene*, *genbase*, *medical* et *tmc2007-500*). À la fin, les stratégies se rejoignent pour nous donner les mêmes résultats lors de l'utilisation du dataset complet (100%). Le gain obtenu peut être quantifié en désignant les données minimales choisies par chacune des stratégies adoptées. En d'autres termes, en apprentissage actif, la stratégie gagnante est celle qui donne de meilleurs résultats avec le moins de données rajoutées possibles.

Ces *MLDatasets* sont d'autant plus difficiles à interpréter et à analyser que leurs *TCS* sont élevés. Le tableau 5.4 montre ces résultats en gardant leur rang selon leur *TCS*. Les cas du *TSVM-AL* sont illustrés dans les figures 5.4 pour les ensembles de données multi-label avec les trois stratégies données et les trois critères d'évaluation.

Les figures 5.4 représentent l'évaluation des trois méthodes *Random*, *MMC* et *BinMin* sur les cinq bases de données (*Yeast*, *Emotion*, *Scene*, *Genbase* et *Medical*). Nous avons commencé l'apprentissage avec 20% de la base de données. Les résultats montrent clairement que *BinMin* a donné de meilleurs taux par rapport aux *Random* et *MMC*.

(a) Résultats de *TSVM* active pour *emotions*(b) Résultats de *TSVM* active pour *genbase*

(c) Résultats de *TSVM* active pour *scene*(d) Résultats de *TSVM* active pour *yeast*

(e) Résultats de *TSVM* active pour *medical*Figures 5.4 – Les performances de *TSVM* active pour les trois stratégies pour différents MLDatasets

Taux de données rajoutées		40%		100%	
Datasets	Mesures	<i>Random</i>	<i>MMC</i>	<i>BinMin</i>	
<i>Emotions</i>	Ham loss ↓	0.265	0.273	0.258	0.235
	Acc ↑	0.512	0.494	0.525	0.551
	Micro F1 ↑	0.631	0.618	0.642	0.675
<i>Scene</i>	Ham loss ↓	0.099	0.097	0.097	0.087
	Acc ↑	0.615	0.611	0.632	0.682
	Micro F1 ↑	0.704	0.706	0.724	0.758
<i>Yeast</i>	Ham loss ↓	0.196	0.197	0.195	0.193
	Acc ↑	0.509	0.503	0.517	0.522
	Micro F1 ↑	0.381	0.394	0.389	0.423
<i>Genbase</i>	Ham loss ↓	0.020	0.015	0.010	0.008
	Acc ↑	0.978	0.978	0.989	0.992
	Micro F1 ↑	0.687	0.725	0.720	0.740
<i>Medical</i>	Ham loss ↓	0.014	0.016	0.014	0.011
	Acc ↑	0.613	0.583	0.612	0.743
	Micro F1 ↑	0.250	0.221	0.233	0.352
<i>Tmc2007</i>	Ham loss ↓	0.081	0.075	0.061	0.059
	Acc ↑	0.782	0.854	0.913	0.922
	Micro F1 ↑	0.544	0.593	0.780	0.786

Table 5.4 – Résultats de classification obtenus par *TSVM* active

La table 5.4 montre les résultats des six MLDatasets en gardant leur rang selon leur *TCS*. Nous avons commencé le processus d'apprentissage avec 20% de la base de données libellée et, à 40% des données libellées, les résultats sont relativement compétitifs et tendent rapidement à 100%, d'où le gain à valeur ajoutée qui en résulte et le bénéfice de l'efficacité de l'apprentissage actif.

De la table 5.4, chaque approche a réagi différemment sur chacune des mesures et pour chaque ensemble de données considéré, l'objectif étant d'enrichir nos résultats afin d'avoir une meilleure interprétation. Pour *genbase*, les résultats obtenus pour les trois mesures considérées sont les plus compétitifs par rapport à tous les autres MLDatasets et la stratégie *BinMin* est plus compétitive que les deux autres stratégies, sur tous les MLDatasets testés.

Cela démontre globalement que la méthode d'apprentissage actif *TSVM* est une approche concurrentielle par rapport aux algorithmes d'apprentissage actifs de pointe pour la classification multi-label du texte ou d'autres domaines, et peuvent réduire considérablement la demande de données à libeller tout en obtenant une qualité satisfaisante. Nos résultats obtenus donnés dans la table 5.4 sont sensiblement identiques ou même meilleurs dans le cas de la mesure *Accuracy* que les derniers résultats de classification obtenus à partir de la littérature produite par *ML-kNN* qui reste une référence dans la classification multi-label [Charte et al., 2016].

Le but est d'arriver à obtenir des taux meilleurs avec le moindre nombre possible d'exemples d'apprentissage, où réside le principe d'apprentissage actif semi-supervisé. Comme contribution, on pourrait appliquer au départ une méthode non supervisée qui permettrait de choisir au mieux cet ensemble d'apprentissage en vue d'améliorer ces résultats.

5. Résultats de l'approche proposée de *SOM* en apprentissage actif multi-label

Dans ce paragraphe, une technique itérative d'apprentissage actif basée sur le réseau neuronal de la carte auto-organisée (*SOM*) est proposée. La technique exploite les propriétés non pas du classifieur *SVM* cette fois-ci, mais bien celles du *SOM* pour identifier des échantillons incertains et divers, à inclure dans l'ensemble d'entraînement. On sélectionne des échantillons incertains à faible densité dans l'espace de caractéristiques en exploitant les propriétés topologiques de la carte *SOM*. Il en résulte une convergence rapide même lorsque les échantillons d'apprentissage initial disponibles sont médiocres. L'efficacité de la méthode proposée est évaluée en la comparant avec plusieurs méthodes existant dans la littérature utilisant nos six Datasets déjà exploitées dans les paragraphes antérieurs.

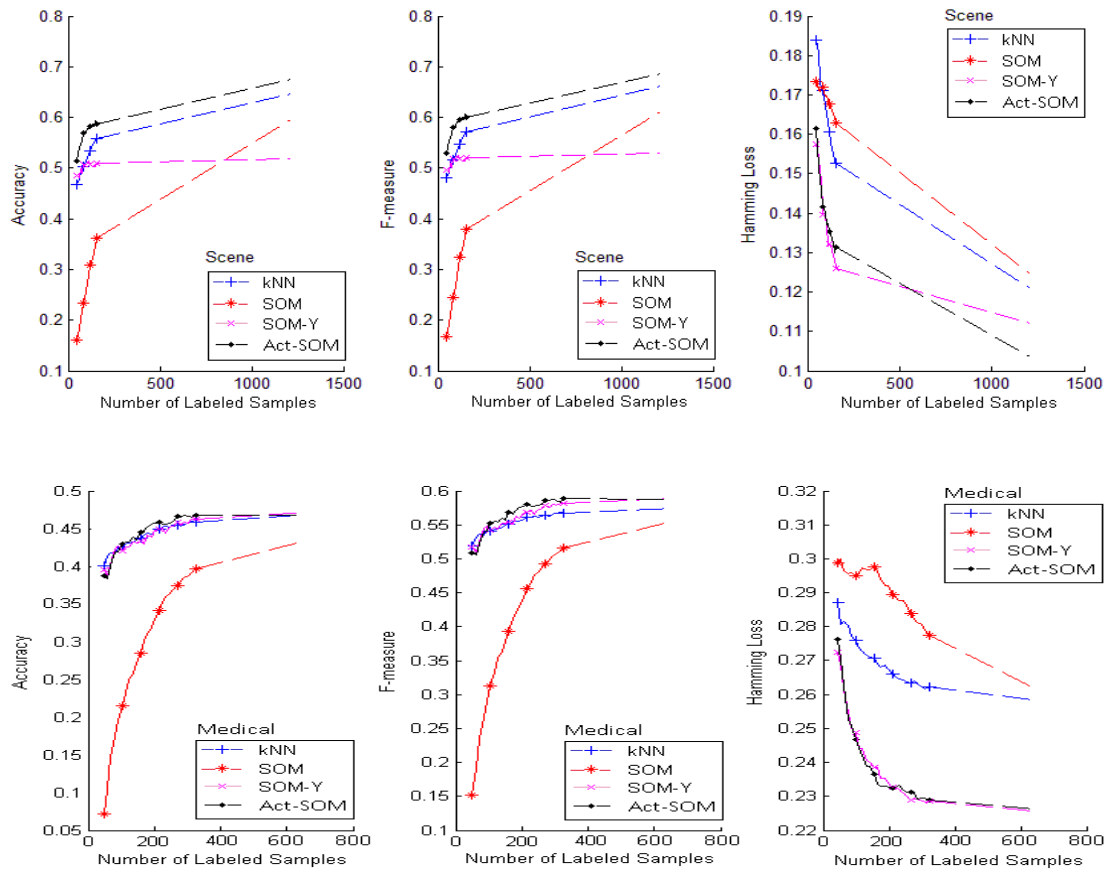
Dans notre approche proposée de *Act-SOM*, le réseau est appris avec toutes les instances d'entrée disponibles, avec et sans labels: $x \in A = U \cup L$. Nous utilisons la stratégie d'incertitude sur $x \in U$ pour choisir les échantillons les plus informatifs.

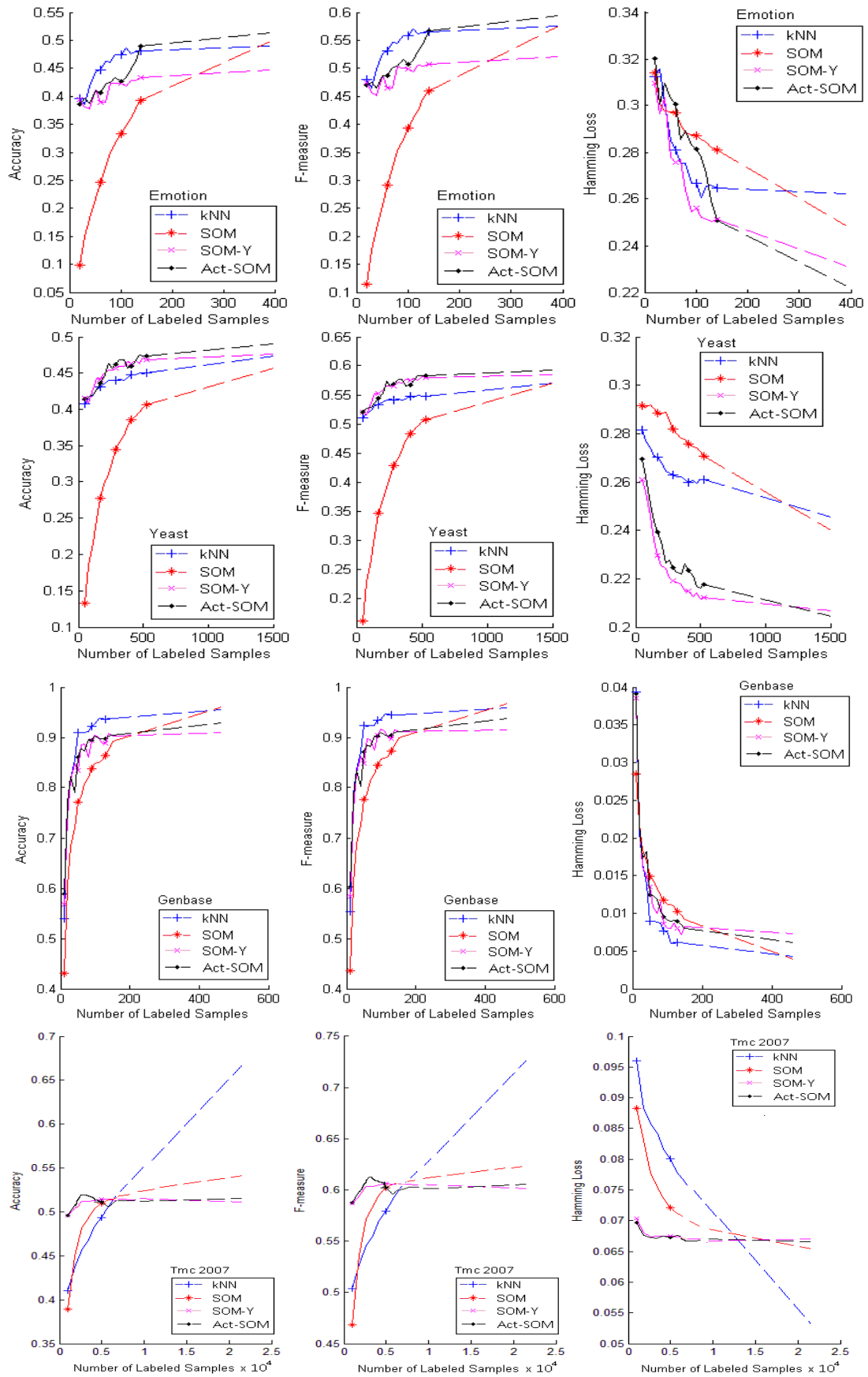
À partir des résultats illustrés avec le nombre d'échantillons additionnels d'apprentissage (voir les figures 1), nous pouvons voir que la performance s'améliore à mesure que la taille d'échantillonnage augmente, d'où l'influence des données libellées rajoutées. Pour les datasets *emotions*, *scene* et *yeast*, notre méthode proposée *Act-SOM* est clairement meilleure que *SOM*, *SOM-Y* et *kNN*, sur la précision (*Accuracy*) et la F-mesure et légèrement en concurrence avec

SOM-Y pour la perte de Hamming (*Hamming loss*), d'où une petite différence des taux pour *scene* et *yeast*.

La méthode d'apprentissage active est effectuée pour 8 itérations, en fonction de la taille et des propriétés de chaque ML Datasets en sélectionnant (le nombre d'échantillons étiquetés initiaux pour l'entraînement, le nombre maximal d'échantillons libellés ajoutés, le pool d'échantillons ajoutés à chaque fois et le coefficient α pour l'algorithme Act-SOM) comme suit: *emotions* (20, 120, 10, 0,9), *scene* (50, 110, 10, 0,7), *yeast* (50, 500, 30, 0,7), *genbase* (10, 150, 10, 0,9), *medical* (50, 340, 20, 0,8), *tmc2007* (1000, 6500, 500, 0,3).

Pour *genbase*, qui est plus complexe (4ème classement TCS) avec 27 étiquettes et moins de densité (0.046), notre méthode proposée est encore meilleure que SOM et SOM-Y, mais légèrement moins bien que kNN et ceci pour les 3 mesures. Pour le plus grand dataset *tmc2007* avec TCS=16.372 le plus élevé, nous voyons clairement que Act-SOM et SOM-Y sont équivalents, mais ils sont meilleurs que SOM et kNN pour les 3 mesures.





Figures 5.5 – Résultats sur les six Datasets pour les trois mesures (Accuracy, F-mesure et Hamming loss).

L'objectif de notre approche est d'avoir une meilleure qualité avec moins de données libellées. Dans la Table 5.5, nous notons que, pour chaque base de données, il est possible de réaliser de très bons taux de classification proche de ceux à 100%, avec seulement le tiers de la base.

Nos résultats expérimentaux montrent que *Act-SOM* a dépassé des méthodes comme le ML-kNN tant dans le cas de l'apprentissage inductif que dans l'apprentissage transductif (c'est-à-dire l'apprentissage actif), dans une large gamme de métriques et de jeux de données. Nous pouvons dire que la conjonction de l'apprentissage actif avec *Act-SOM* a également montré une plus value (compétence) en relation avec d'autres conjonctions actives dans la littérature.

Les résultats montrent que les échantillons obtenus à partir d'un cluster *SOM* semi-supervisé peuvent relativement effectuer, d'une façon similaire à l'ensemble de données en entrée, en préservant la topologie et peuvent être utilisés comme un dataset libellé approprié pour la classification ultérieure de clusters non libellés.

Il s'avère que lorsque la taille de l'ensemble de données est faible, la taille des clusters représentatifs nécessaires à l'apprentissage du classifieur n'est pas assez grande, d'où l'utilisation dans l'approche semi-supervisée, de datasets plus larges pour récupérer un nombre relativement élevé de clusters assez représentatifs des caractéristiques générales des données d'entrée.

Data sets		Emotions			Scene			Yeast		
Echantillons libelés		20	120	391	10	140	1500	50	450	1500
<i>Act-SOM</i>	Ham loss↓	0.321	0.253	0.224	0.162	0.132	0.104	0.257	0.219	0.215
	Acc↑	0.396	0.495	0.518	0.518	0.596	0.652	0.418	0.483	0.489
	F-Measure↑	0.472	0.574	0.591	0.537	0.601	0.683	0.520	0.584	0.586
<i>SOM-Y</i>	Hamloss↓	0.318	0.252	0.232	0.158	0.127	0.114	0.248	0.218	0.213
	Acc↑	0.397	0.426	0.439	0.481	0.501	0.501	0.432	0.458	0.482
	F-Measure↑	0.478	0.500	0.511	0.499	0.512	0.512	0.543	0.566	0.579
<i>SOM</i>	Hamloss↓	0.316	0.283	0.256	0.174	0.162	0.128	0.295	0.278	0.240
	Acc↑	0.009	0.386	0.482	0.152	0.364	0.586	0.201	0.398	0.449
	F-Measure↑	0.122	0.453	0.561	0.178	0.382	0.586	0.251	0.499	0.552
<i>kNN</i>	Hamloss↓	0.316	0.261	0.264	0.185	0.153	0.124	0.279	0.258	0.248
	Acc↑	0.388	0.484	0.480	0.478	0.562	0.613	0.428	0.449	0.481
	F-Measure↑	0.480	0.576	0.576	0.483	0.576	0.620	0.532	0.549	0.550

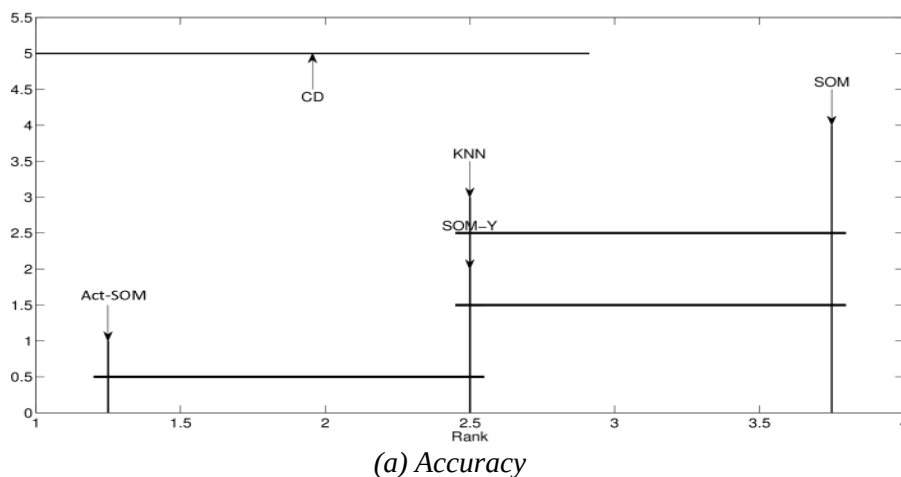
Data sets		Genbase			Medical			Tmc 2007		
Echantillons libelés		10	150	463	50	340	645	1000	6500	21519
<i>Act-SOM</i>	Ham loss↓	0.039	0.007	0.006	0.276	0.228	0.225	0.069	0.065	0.065
	Acc↑	0.591	0.901	0.912	0.382	0.473	0.473	0.492	0.532	0.518
	F-Measure↑	0.600	0.900	0.932	0.513	0.586	0.585	0.583	0.624	0.601
<i>SOM-Y</i>	Hamloss↓	0.038	0.008	0.008	0.272	0.229	0.225	0.071	0.065	0.065
	Acc↑	0.583	0.901	0.901	0.391	0.463	0.474	0.492	0.518	0.517
	F-Measure↑	0.591	0.910	0.910	0.516	0.574	0.574	0.584	0.600	0.601
<i>SOM</i>	Hamloss↓	0.028	0.010	0.005	0.299	0.278	0.265	0.096	0.072	0.065
	Acc↑	0.432	0.895	0.952	0.071	0.399	0.436	0.386	0.518	0.549
	F-Measure↑	0.432	0.895	0.963	0.150	0.511	0.543	0.475	0.601	0.625
<i>kNN</i>	Hamloss↓	0.039	0.006	0.004	0.289	0.263	0.260	0.096	0.080	0.053
	Acc↑	0.544	0.948	0.951	0.400	0.451	0.452	0.418	0.516	0.663
	F-Measure↑	0.556	0.953	0.958	0.521	0.562	0.563	0.504	0.601	0.736

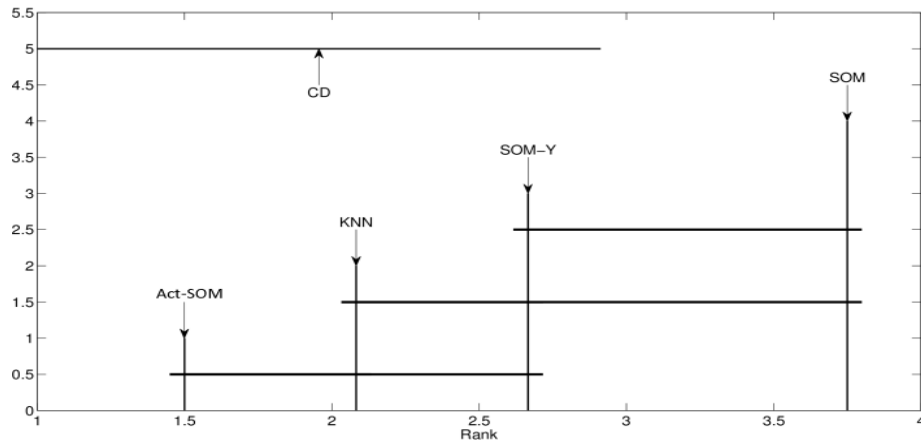
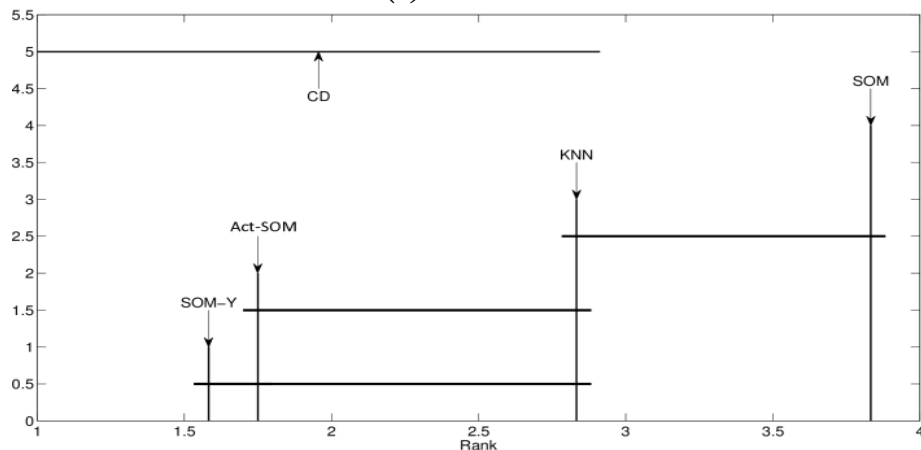
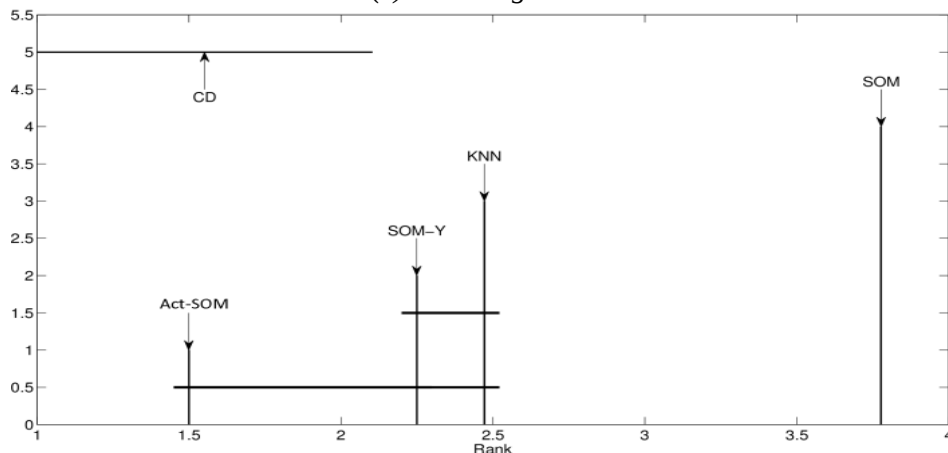
Table 5.5 – Performance de l'apprentissage actif multi-label sur les six MLDatasets

Parmi les différentes figures 5.5 et la Table 5.5, nous pouvons constater que notre approche proposée *Act-SOM* d'apprentissage actif a généralement produit de meilleurs résultats par rapport aux 3 autres approches (*SOM*, *SOM-Y* et *kNN*) qui sont tous des méthodes à base de clusters. L'utilisation des six Datasets avec leurs différentes propriétés et domaines d'application, confirme l'approche proposée et renforce l'hypothèse du clustering pour la sélection d'échantillons informatifs.

- *Evaluations statistique :*

Nous présentons à partir de la statistique de Friedman (distribution de χ^2 avec $k-1$ degrés de liberté où k étant le nombre d'approches testées, voir pour tests de Friedman en [Friedman, 1940] et pour comparaison de distributions [Nemenyi, 1963]) qui est un test non paramétrique pour les tests d'hypothèses multiples, les résultats du test post-hoc de Nemenyi avec des diagrammes de rangs moyens [Demsar, 2006], où un diagramme critique (CD) représente une projection des approches de rangs moyens sur l'axe énuméré (voir les figures 5.6). Le test Nemenyi permet de comparer tous les classifieurs entre eux. Les approches sont ordonnées de gauche (la meilleure) à droite (la plus mauvaise) et une ligne épaisse qui relie les approches avec les rangs moyens non significativement différents (pour un niveau de signification de 5%). Les figures 5.6 représentent respectivement, en tenant compte des trois critères, les graphiques de test global : *NemenyiTestPlotHigh* (Global) qui confirme que la méthode proposée *Act-SOM* est la meilleure, sur les 3 critères. Les autres graphiques représentent respectivement *NemenyiTestPlotHigh* (.) Pour (a) Précision, (b) F-measure, (c) HamLoss et confirment que la méthode proposée *Act-SOM* est la meilleure. Seul le critère HamLoss montre que *Act-SOM* et *SOM-Y* sont très proches, mais globalement, selon les 3 critères, le graphique global montre clairement que *Act-SOM* est le plus performant comme classifieur.



(b) *F-mesure*(c) *Hamming Loss*(d) *Test global*

Figures 5.6 – Les diagrammes critiques (CD) pour les trois évaluations des quatre approches de classification

Nous pouvons également noter que *Act-SOM* est la méthode la plus performante, suivie de *SOM-Y*, *kNN* et *SOM*. L'approche *Act-SOM* est performante en termes de précision (*Accuracy*) et de *F-mesure*, se place deuxième selon *Hamming Loss*.

L'approche par simple *SOM* présente des performances médiocres selon toutes les mesures d'évaluation. Nous supposons que la structure topologique simple de la carte *SOM* n'optimise pas la performance prédictive et nécessite un espace caractéristique plus riche, comme c'est le

cas avec *Act-SOM* qui traite un mélange d'instances avec leurs étiquettes à l'entrée. Par conséquent, plus l'espace d'entrée est informatif, plus la performance prédictive du classifieur est élevée.

6. Conclusions

L'apprentissage actif pour la classification multi-label est encore à l'état préliminaire. Toutes les stratégies adoptées en apprentissage actif, ont en commun pour essayer d'utiliser le moins d'exemples possibles et de sélectionner les exemples les plus informatifs.

Le principal défi de l'apprentissage actif multi-label est de développer des stratégies efficaces pour évaluer l'information unifiée d'une instance non libellée dans toutes les classes. Des travaux d'apprentissage actif multi-label certes existent, mais ils mesurent l'information d'une occurrence non libellée en traitant tous les labels indépendamment sans tenir compte d'informations implicites potentielles sur la structure du label dans toutes les classes. Le défi de classement multi-label est la façon dont nous pouvons faire face à la dépendance de la prédiction du label pour tirer parti des informations de pertinence entre labels.

La technique d'apprentissage active proposée *Act-SOM* exploite l'hypothèse du clustering pour trouver et sélectionner les échantillons les plus informatifs en utilisant la stratégie d'incertitude mesurée selon la cartographie de *SOM* ayant la distance moyenne la plus élevée des voisins. L'amélioration significative de la performance sur certains datasets populaires a démontré l'efficacité de l'*Act-SOM* actif pour la classification multi-label. L'analyse statistique a confirmé globalement la supériorité de la méthode proposée.

À notre connaissance, il n'existe aucun autre document sur la tâche intéressante et à venir de la classification multi-label utilisant *Act-SOM* en apprentissage actif. Nous prévoyons poursuivre notre travail selon deux directions principales.

Tout d'abord, nous avons l'intention d'étudier d'autres combinaisons d'algorithmes semi-supervisés pour des datasets plus importants, et en particulier nous nous concentrerons sur une compréhension théorique plus approfondie de l'approche proposée par rapport à l'apprentissage actif *SVM* en capturant la régularisation relationnelle multi-cluster. Cette interdépendance des clusters donne un avantage palpable en appliquant l'approche machine développée *Act-SOM* sur la perte globale multi-classe en considérant toutes les classes simultanément.

Conclusion générale

L'apprentissage des données multi-label est une tâche très difficile. Un large éventail de métriques de caractérisation multi-label existe dans la littérature, dont certaines ont été décrites dans ce manuscrit, et plusieurs méthodes ont été définies pour traiter et classifier des données multi-label. Le travail de cette thèse a concerné initialement une possible contribution de ma part à l'apprentissage semi-supervisé en contexte multi-label. Il en ressort d'abord l'intérêt grandissant attaché par la communauté scientifique à un tel sujet combinant les trois paradigmes principaux, à savoir :

✓ L'apprentissage multi-label a reçu une attention significative dans la communauté de la recherche au cours des dernières années: cela a permis de développer une variété de méthodes d'apprentissage multi-label. L'apprentissage multi-label est un paradigme d'apprentissage supervisé très récent. Dans notre travail, on s'intéresse à l'apprentissage semi-supervisé où dans de nombreuses applications, les données sont non étiquetées ou l'étiquetage est coûteux ou peu pratique. Ce fait est encore plus difficile dans l'apprentissage multi-label où l'effort d'étiquetage devrait être moindre avec à l'appui un choix rigoureux des exemples à étiqueter parmi de grandes quantités de données non étiquetées. L'importance et la portée réelle de l'apprentissage multi-label se mesurent au quotidien avec la variété intra et inter-domaines d'application tenant compte de toutes les éventualités et choix possibles pouvant nous renseigner sur la diversité des classes et leurs corrélations. Un exemple concret est celui d'un individu qui peut être atteint à la fois de plusieurs déficiences ou maladies et dont le traitement nécessite leur prise en compte simultanée. La diversité fait la richesse et la multi-labellisation est une nécessité pour mieux appréhender à l'avenir le traitement multidisciplinaire de grandes bases de données.

✓ L'étude de l'apprentissage semi-supervisé est motivée par deux facteurs: sa valeur pratique dans la construction de meilleurs algorithmes informatiques, et sa valeur théorique dans la compréhension de l'apprentissage dans les machines. L'apprentissage semi-supervisé a une valeur pratique considérable. Dans de nombreuses tâches, il y a une pénurie de données libellées. Les labels y peuvent être difficiles à obtenir car ils nécessitent des annotateurs humains, des dispositifs spéciaux ou des expériences coûteuses et lentes. L'apprentissage semi-supervisé fournit également un modèle computationnel de la façon dont les humains apprennent à partir de données étiquetées et non étiquetées pour faciliter l'apprentissage conceptuel. L'étude de l'apprentissage semi-supervisé est donc une occasion de relier l'apprentissage machine et l'apprentissage humain. Il s'avère quelquefois préférable à un apprentissage supervisé, tel un élève qu'on ne doit jamais tout lui montrer et c'est à lui d'explorer et de découvrir de lui-même le reste à sa façon et selon ses capacités et objectifs (apprentissage transductif) en devenant plus performant.

✓ Comme on l'a souligné auparavant (au chapitre 3), il ressort clairement que l'apprentissage semi-supervisé guidé par un apprentissage actif aboutit à de meilleurs résultats et qu'à l'avenir ces deux modes d'apprentissage seront fortement liés. Cet apprentissage actif s'impose de lui-même s'agissant d'un apprentissage semi-supervisé offrant le meilleur

compromis entre la performance de la classification et le choix du nombre minimal d'exemples requis pour atteindre une bonne convergence. Donc, cet apprentissage apporte une valeur ajoutée certaine à l'apprentissage semi-supervisé en lui choisissant l'information utile, pertinente et aussi courte que possible, car l'opération d'étiquetage (l'annotation) est onéreuse et gourmande en temps. Malgré l'importance du problème, la recherche actuelle sur l'apprentissage actif pour la classification multi-label reste à l'état préliminaire. Au chapitre 4, l'idée d'espace des versions, comme ensemble de toutes les hypothèses cohérentes avec les données d'apprentissage, s'accorde a priori avec la vision actuelle de l'approche de l'apprentissage artificiel, considérant l'apprentissage comme la sélection des hypothèses les plus performantes par rapport aux observations.

La plupart des études d'apprentissage abordant ces trois paradigmes, réunis ou disjoints, se servent des classifieurs de type *SVM* qui ont été majoritairement employés et développés initialement pour traiter des problèmes binaires, surtout en mono-label. La plupart des méthodes d'apprentissage actif multi-label décomposent la classification multi-label en un ensemble de problèmes de classification binaire et non globale et prennent des décisions de sélection d'instance en exploitant ces classifieurs binaires (un-contre-tous) indépendamment sans considérer l'information de structure d'étiquette d'une instance révélée dans toutes les classes. Certes, comme il a été souligné, cela donne de bons résultats provenant du bon niveau de généralisation induit par les *SVM* et en sélectionnant les exemples qui minimisent la plus petite marge *SVM* entre tous les classifieurs binaires ou de prendre la moyenne des scores de leur incertitude, mais n'empêche que la décision prise est naïve au sens de séparer pour mieux prédire, alors que pour le cas multi-label multi-classe, il faut plutôt penser à une stratégie de raisonnement collaborative (tous-pour-un).

Pour parer à ce problème, on a utilisé de notre côté, la carte topologique pour son pouvoir d'abord de réseau neuronal compétitif, de reconfiguration des données en préservant la topologie et surtout de la possibilité d'interprétation des résultats obtenus. Ainsi, on a fait appel à la carte *SOM* et ses variantes pour l'apprentissage actif dans le but de sélectionner les instances les plus informatives à libeller en adoptant une stratégie de sélection basée sur l'incertitude des labels et où l'étiquetage inutile des échantillons non informatifs est évité, réduisant ainsi considérablement le coût de l'étiquetage tout en augmentant la qualité de l'ensemble d'apprentissage.

A partir de la technique d'apprentissage actif qui sélectionne l'échantillon le plus incertain tout en étant le plus proche de l'hyperplan de séparation courant d'un *SVM*, on a essayé, au niveau de la carte *SOM*, de sélectionner l'échantillon le plus incertain tout en étant le plus proche de la zone de séparation entre deux clusters, la région à plus faible densité, d'où notre principale contribution. Au fait, au niveau de notre carte *Act-SOM*, chaque cluster réagit comme un classifieur en présence des autres clusters sans pour autant les ignorer, donc à base de clusters compétitifs, intégrés et en simultanée, tous à la fois. Cette différence d'architecture et de conception au niveau de l'apprentissage est à l'avantage des *SOM* pour tirer profit dans le cadre multi-label, de la corrélation globale entre les attributs instances et labels associés et aussi de l'informativité unifiée des instances non libellées.

- *Perspectives*

De nouvelles connaissances sont issues de l'analyse collective des données avec l'avènement de grandes bases de données (big data) et la mise à l'échelle nous interpellent à reconsidérer nos stratégies de partitionnement et de classification.

✓ Pour les *SVM* : Les méthodes basées sur *SVM* fonctionnent mieux pour les bases de données réduites, car le noyau gaussien peut gérer très bien un petit nombre d'exemples, mais lorsque le nombre d'exemples augmente, la performance du noyau approche de celle d'un noyau linéaire. En outre, les méthodes basées sur *SVM* sont meilleures pour les domaines avec un plus grand nombre de features, où chaque caractéristique peut jouer un rôle crucial dans la prévision correcte, ainsi les *SVM* exploitent les informations de toutes les caractéristiques.

✓ Pour les *SOM* : Il convient de noter que, comme toute autre technique non linéaire de réduction de dimensionnalité, un réseau neuronal *SOM* ne garantit pas la préservation de la topologie quelque soit la nature du problème donné, en particulier lorsque la caractéristique dimensionnelle est très élevée (voir chapitre 3) et cela se répercute sur le choix des clusters les plus informatifs et représentatifs.

Il convient de souligner qu'on tourne de plus en plus vers des problèmes de grande envergure où le nombre de labels est extrêmement important d'où la nécessité de méthodes de cartographie telles les cartes topologiques compétitives *SOM*, pour la réduction des espaces entrée-sortie (instances-labels) qui se traduit par la réduction de complexités informatique et spatiale.

La classification multi-label est le domaine approprié où des méthodes dédiées sont étudiées et proposées pour remplir la tâche d'étiquetage des ressources en plusieurs catégories et où l'information est un atout essentiel pour améliorer la qualité de vie et le progrès social.

Liste des travaux :

- 1) A. Benyettou, Y. Bennani, A. Benyettou, A. Bendahmane, G. Cabanes, *Semi-Supervised Multi-Label Classification Through Topological Active Learning*, International Journal on Communications Antenna and Propagation (IRECAP), Vol.7, N°3, pp.222-232, June 2017 (ISSN : 2039-5086, Doi : 10.15866/irecap.v7i3.12742).
- 2) Assia Benyettou, Abderrahmane Bendahmane, Khalid Benabdeslem, *Sélection de variables pour l'amélioration de l'accès à l'information sur le Web*, Colloque sur l'Optimisation et les Systèmes d'Information, COSI 2015, 1-3 juin 2015, Oran, Algérie.
- 3) Assia Benyettou, Abderrahmane Bendahmane, Abdelhadi Lotfi, *Variables selection by Support Vector Machines for web pages classification*, Conférence Internationale des Télécommunications et des TIC, ICTTelecom'15, 16-17 mai 2015, Oran.
- 4) Freha Mezzoudj, Assia Benyettou, *On the Optimization of Multiclass Support Vector Machines Dedicated to Speech Recognition*, 19th International Conference On Neural Information Processing, ICONIP 2012, Doha, Qatar, November 12-15, 2012. Part.II, LNCS 7664, pp.1-8, 2012.

REFERENCES

- Abaei G, Selamat A, Fujita H. An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction, *Knowledge-Based Systems* 74 (2015), 28-39.
- Abbas OA. Comparisons between data clustering algorithms, *Int. Arab J. Inform. Technol.* 5 (2008) 320-325.
- Abbas Q, Celebi M, Serrano C, García IF, Ma G. Pattern classification of dermoscopy images: a perceptually uniform model. *Pattern Recogn* 2013, 46:86-97.
- Adankon MM, Cheriet M. Learning semi-supervised SVM with genetic algorithm. In *Proceedings of the International conference on Neural Networks*, 2007.
- Agarwal R, Aggarwal CC, Prasad V. A tree projection algorithm for generation of frequent item sets. *J Parallel Distr Com* 2001, 61:350-371.
- Aggarwal CC, Kong A, QuanguanGu, Han J, Yu PS. Active learning: a survey. In *Data classification: algorithms and applications*, CC Aggarwal, Ed. CRC Press, 2014, 571-606.
- Aha DW. (ed.): *Lazy Learning*. Springer, 1997.
- Aha DW, Kibler D, Albert MK. Instance-based learning algorithms. *Mach. Learn.* 6(1), 1991, 37-66.
- Antenreiter M, Ortner R, Auer P. Combining classifiers for improved multilabel image classification. In: *Proceedings of the 1st Workshop on Learning from Multilabel Data (MLD) Held in Conjunction with ECML/PKDD*, Bled, Slovenia; 2009, 16-27.
- Atkinson RD, Castro DD. Digital quality of life. Technical report, Information Technology and Innovation Foundation, October 2008.
- Ávila J, Gibaja E, Ventura S. Evolving multi-label classification rules with gene expression programming: a preliminary study. In: *Hybrid Artificial Intelligence Systems (HAIS)*, Lecture Notes in Computer Science, vol. 6077; 2010, 9-16.
- Ávila JL, Gibaja EL, Zafra A, Ventura S. A gene expression programming algorithm for multi-label classification. *J Mult-Valued Log S* 2011, 17:183-206.
- Azcarraga AP, Hsieh MH, Pan SL, Setiono R. Extracting salient dimension for automatic SOM labeling, *IEEE Trans. Systems, Man, and Cybernetics – Part C: Applications and Reviews*, Vol.35, N°4, Nov.2005, 595-600.
- Barutcuoglu Z, Schapire RE, Troyanskaya OG. Hierarchical multi-label prediction of gene function. *Bioinformatics* 2006, 22:830-836.
- Bhowmick PK, Basu A, Mitra P, Prasad A. Sentence level news emotion analysis in fuzzy multi-label classification framework (special issue on natural language processing and its applications). *Res Comput Sci* 2010, 46:143-154.
- Bielza C, Li G, Larrañaga P. Multi-dimensional classification with Bayesian networks. *Int J Approx Reasoning* 2011, 52:705-727.
- Blockeel H, Raedt LD, Ramon J. Top-down induction of clustering trees. In: *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, San Francisco, CA, USA; 1998, 55-63.
- Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*, 2001.
- Boutell M, Luo J, Shen X, Brown C. Learning multi-label scene classification. *Pattern Recogn* 2004, 37:1757-1771.
- Breiman L, Friedman J, Olshen R, Stone CJ. *Classification and Regression Trees*, Chapman & Hall/CRC, 1984.
- Breiman L. Bagging predictors, *Machine Learning*, (1996) 24(2):123-140,
- Breiman L. Random forests, *Machine Learning*, 45 (2001) 5-32.
- Brinker K, Fürnkranz J, Hüllermeier E. A unified model for multilabel classification and ranking. In: *Proceeding of the ECAI 2006: 17th European Conference on Artificial Intelligence*; 2006, 489-493.

- Brinker K. *On active learning in multi-label classification, "from data and information analysis to knowledge engineering" of bookseries "studies in classification, data analysis, and knowledge organization"*, Springer, 2006, 1, 2.
- Bruzzzone L, Chi M, Marconcini M. A novel transductive SVM for semi-supervised classification of remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* 44 (11), 2006, 3363-3373.
- Burges CJC. A tutorial on support vector machines for pattern recognition. *Data Mining Knowl. Discov*; vol. 2, no. 2, 1998, 121-167.
- Campbell N, Cristianini N, Smola AJ. Query learning with large margin classifiers. In *Proceedings of the 7th International Conference on Machine Learning (ICML'00)*, pages 111-118, 2000.
- Cano A, Zafra A, Galindo ELG, Ventura S. A grammar-guided genetic programming algorithm for multi-label classification. In: *16th European Conference, EuroGP*, Lecture Notes in Computer Science, vol. 7831; 2013, 217-228.
- CDC/National Center for Health Statistics, International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM), <<http://www.cdc.gov/nchs/icd/icd9cm.htm>> (2011).
- Chan A, Freitas AA. A new ant colony algorithm for multi-label classification with applications in bioinformatics. In: *GECCO'06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, New York, USA; 2006, 27-34.
- Chang CC, Lin CJ. LIBSVM: a library for support vector machine, 2001. <http://csie.ntu.edu.tw/~cjlin/libsvm>.
- Chapelle O, Zien A. Semi-supervised classification by low density separation. In *Proceedings of the 10th intern. workshop Artificial Intelligence and Statistics*. 2005, 57-64.
- Charte F, Rivera AJ, del Jesus MJ, Herrera F. Multilabel classification. Problem analysis, metrics and techniques book repository. <https://github.com/fcharte/SM-MLC>
- Charte F, Rivera AJ, del Jesus MJ, Herrera F. QUINTA: a question tagging assistant to improve the answering ratio in electronic forums. In *EUROCON 2015 - International Conference on Computer as a Tool (EUROCON)*, IEEE, 2015, 1-6. doi:[10.1109/EUROCON.2015.7313677](https://doi.org/10.1109/EUROCON.2015.7313677)
- Charte F, Rivera AJ, del Jesus MJ, Herrera F. On the impact of dataset complexity and sampling strategy in multi-label classifiers performance. In: *Proceedings of 11th International Conference on Hybrid Artificial Intelligent Systems, HAIS'16*, vol. 9648, Springer, 2016, 500-511.
- Chen Y, Wang G, Dong S. Learning with progressive transductive support vector machine. *Pattern Recognit. Letters*. 24(12), 2003, 1845-1855.
- Cheng W, Hullermeier E. Combining instance-based learning and logistic regression for multilabel classification, *Machine Learning* 76 (2009) 211-225.
- Cherman EA, Tsoumakas G, Monard MC. Active learning algorithms for multi-label data, *IFIP: Intern. Conf. on Artificial Intelligence and Innovations*, Springer Intern. Publishing, 2016, 267-279.
- Ciarelli PM, Oliveira E, Badue C, Souza AF. Multi-label text categorization using a probabilistic neural network. *Int J Comput Inf Syst Ind Manage Appl* 2009, 1:133-144.
- Ciarelli PM, Oliveira E. An enhanced probabilistic neural network approach applied to text classification. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, vol. 5856, chap. 78. Berlin/Heidelberg: Springer; 2009, 661-668.
- Clare A, King RD. Knowledge discovery in multi-label phenotype data. In: *PKDD'01 Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, vol. 2168; 2001, 42-53.
- Cohn DA, Ghahramani Z, Jordan MI. Active learning with statistical models. In *Advances in Neural Information Processing Systems*. The MIT Press, 1995, vol.7, 705-712.
- Cong H, Tong LH. Grouping of TRIZ inventive principles to facilitate automatic patent classification. *Expert Systems with Applications* 2008, 34:788-795.
- Cornuéjols A. Apprentissage supervisé et espace des versions. Cours, ENSTA/CNAM-IIE et L.R.I., Université de Paris-Sud, Orsay, France, 2009.

- Cortes C, Vapnik V. Support-vector networks. *Machine Learning*, 1995, 20(3): 273-297.
- Crammer K, Singer Y. A family of additive online algorithms for category ranking. *J Machine Learning Research*; 2003, 3:1025-1058.
- Christianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines and other kernel based learning methods*, Cambridge University Press, 2000.
- de Carvalho A, Freitas A. A tutorial on multi-label classification techniques. In: *Foundations of Computational Intelligence*, vol. 5, Berlin/Heidelberg: Springer; 2009, 177-195.
- de Comité F, Gilleron R, Tommasi M. Learning multi-label alternating decision trees from texts and data. In: *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'03)*, Berlin/Heidelberg: Springer; 2003, 35-49.
- Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 2002, 3:182-197.
- Dembczyński K, Cheng W, Hüllermeier E. Bayes optimal multilabel classification via probabilistic classifier chains. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*; 2010, 279-286.
- Demir B, Persello C, Bruzzone L. Batch-mode active-learning methods for the interactive classification of remote sensing images, *IEEE Trans. Geosci. Remote Sens.*, Mar. 2011, vol. 49, no. 3, pp. 1014-1031.
- Dempster AP, Laird N.M, Rubin D.B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 1977, 39:1-38.
- Demsar J. Statistical comparisons of classifiers over multiple data sets, (2006) *Journal of Machine learning Research* 7 1-30.
- Diplaris S, Tsoumakas G, Mitkas P, Vlahavas I. Protein classification with multiple algorithms. In: *Proceedings of the 10th Panhellenic Conference on Informatics (PCT 2005)*, Volos, Greece (2005) 448-456.
- Donmez P, Carbonell JG, Bennett PN. Dual strategy active learning. In *Proc. of ECML*, 2007.
- Duwairi R, Kassawneh A. A framework for predicting proteins 3D structures. In: *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '08)*, Washington, DC, USA; 2008, 37-44.
- Duygulu P, Barnard K, de Freitas JFG, Forsyth D. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part IV. LNCS*, vol. 2353, Springer, Heidelberg, 2002, 97-112. doi:[10.1007/3-540-47979-1-7](https://doi.org/10.1007/3-540-47979-1-7)
- Elisseeff A, Weston J. A kernel method for multi-labelled classification. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 14; 2001, 681-687.
- Esuli A, Sebastiani F. Active learning strategies for multi-label text classification. In *Proceedings of the 31th European Conf. on IR research on advances in information retrieval, ser. ECIR'09*. Berlin, Heidelberg: Springer-Verlag, 2009, 102-113.
- Fan RE, Lin CJ. A study on threshold selection for multi-label classification. Technical Report, National Taiwan University, 2007.
- Frank A, Asuncion A. UCI Machine Learning Repository, University of California, School of Information and Computer Sciences, 2010. <<http://archive.ics.uci.edu/ml>>.
- Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 1997, 55:119-139.
- Fürnkranz J, Hüllermeier E, Loza menca E, Brinke K. Multilabel classification via calibrated label ranking. *Machine Learning*, 2008, 73:133-153.
- Fürnkranz J. Round robin classification, *Journal of Machine Learning Research*; 2002, 2:721-747.
- Gibaja E, Ventura S. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisc. Rev. Data Min. Knowl. Discovery* 4(6), 2014, 411-444. doi:[10.1002/widm.1139](https://doi.org/10.1002/widm.1139)
- Gibaja E, Ventura S. A tutorial on multilabel learning. *ACM Comput. Surv.* 47(3), 2015, 1-38. doi:[10.1145/2716262](https://doi.org/10.1145/2716262)

- Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification. In: *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*; 2004, 22–30.
- Gonçalves T, Quaresma P. The impact of NLP techniques in the multilabel text classification problem. In: *Proceedings of Intelligent Information Processing and Web Mining (IIPWM'04), Advances in Soft Computing*; 2004, 424–428.
- Gonçalves EC, Plastino A, Freitas AA. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI)*; 2013, 469–476.
- Guo Y, Greiner R. Optimistic active learning using mutual information. In *Proc. of IJCAI*, 2007.
- Guo Y, Schuurmans D. Discriminative batch mode active learning. In *Proc. Of NIPS*, 2007.
- Guo Y, Schuurmans D. Adaptive large margin training for multi-label classification. In *Proc. of AAAI*, 2011.
- Hajjar C. Cartes auto-organisatrices pour la classification de données symboliques mixtes, de données de type intervalle et de données discrétisées. Supélec, 2014. NNT : 2014SUPL0066, 16 apr.2015; <https://tel.archives-ouvertes.fr/tel-01142849>
- Haykin S. *Neural Networks—A Comprehensive Foundation*. Singapore: Pearson Education, 2003.
- Hüllermeier E, Fürnkranz J, Cheng W, Brinker K. Label ranking by learning pairwise preferences. *Artif Intell* 2008, 172:1897–1916.
- Hung CW, Lin HT. Multi-label active learning with auxiliary learner. In *3rd Asian conf. on Machine Learning*. Taoyuan, Taiwan, 2011.
- Ioannou M, Sakkas G, Tsoumakas G, Vlahavas IP. Obtaining bipartitions from score vectors for multi-label classification. In: *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*; 2010, 409–416.
- Joachims T. *Text categorization with support vector machine: Learning with many relevant features*, Springer Verlag, 1998, 137-142.
- Joachims T. Transductive inference for text classification using support vector machines. In *Proceedings of the seventeenth international Conf. on Machine Learning*. 1999, 200-209.
- Joachims T. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- Kajdanowicz T, Wozniak M, Kazienko P. Multiple classifier method for structured output prediction based on error correcting output codes. In: *Intelligent Information and Database Systems, Lecture Notes in Computer Science*, vol. 6592; 2011, 333–342.
- Katakis I, Tsoumakas G, Vlahavas I. Multilabel text classification for automated tag suggestion. In: *Proceedings of the ECML/PKDD 2008 Discovery Challenge*; 2008.
- Klimt B, Yang Y. The enron corpus: a new dataset for email classification research. In Boulicaut JF, Esposito F, Giannotti F, Pedreschi D. (eds.) *ECML 2004. LNCS (LNAI)*, vol. 3201, 2004, 217–226. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-30115-8-22](https://doi.org/10.1007/978-3-540-30115-8-22)
- Kocev D, Vens C, Struyf J, Džeroski S. Ensembles of multi-objective decision trees. In: *Proceedings of the 18th European Conference on Machine Learning (ECML'07)*, Berlin/Heidelberg: Springer; 2007, 624–631.
- Kocev D. Ensembles for predicting structured outputs, PhD. Thesis, IPS Jozef Stefan, Ljubljana, Slovenia, 2011.
- Kohonen T. *Self-organizing Maps*. Springer Berlin, 2001.
- Kumar N, Berg AC, Belhumeur PN, Nayar SK. Attribute and simile classifiers for face verification. In: *IEEE International Conference on Computer Vision (ICCV)*, 2009.
- Lawrence RD, Almasi GS, Rushmeier HE. A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. *Data Mining Knowl. Discov.*, vol. 3, no. 2, 1999.

- Lebbah M, Thiria S, Badran F. Topological map for binary data. In *Proceedings European Symposium on Artificial Neural Networks-ESANN 2000, Bruges, April 26-27-28, 2000*, 267–272.
- Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'94)*, 1994, 3-12.
- Lewis DD, Yang Y, Rose TG, Li F. RCV1: a new benchmark collection for text categorization research. *J Mach Learn Res* 2004, 5:361–397.
- Li X, Wang L, Sung E. Multilabel SVM active learning for image classification. In: *International Conference on Image Processing (ICIP'04)*; 2004, 2207–2210.
- Lin HT, Lin CJ, Weng RC. A Note on Platt's Probabilistic Outputs for Support Vector Machines. Technical report, 2003.
- Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering, *Internet Computing, IEEE* 2003, 7(1): 76-80.
- Liu G, Lin Z, Yu Y. Multi-output regression on the output manifold. *Patterns Recognition* 2009, 42: 2737–2743.
- Lo H, Wang J, Wang H, Lin S. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *IEEE Trans Multimedia* 2011, 13:518–529.
- López VF, de la Prieta F, Ogihara M, Wong DD. A model for multi-label classification and ranking of learning objects. *Expert Syst Appl* 2012, 39: 8878–8884.
- Luo T, Kramer K, Goldgof DB, Hall LO, Samson S, Remsen A, Hopkins T. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 2005, 6:589-613.
- Luo X, Heywood Z. Evaluation of two systems on multi-class multi-label document classification. In: *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*; 2005, 161–169.
- MacQueen JB. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*. Berkeley, CA, University of California Press, 1967, 281-297.
- Madjarov G, Kocev D, Gjorgjevikj D, Džeroski S. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*; 2012, 45:3084–3104.
- Mammadov MA, Rubinov AM, Yearwood J. The study of drug-reaction relationships using global optimization techniques. *Optim Method Softw* 2007, 22:99–126.
- McCallum AK. Multi-label text classification with a mixture model trained by EM. In: *AAAI 99 Workshop on Text Learning*; 1999.
- Melgani F, Bruzzone L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.*, vol.42, no8, Aug. 2004, 1778-1790.
- Mencía EL, Fürnkranz J. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD-2008)*, Springer-Verlag; 2008, 50–65.
- Mencía EL, Fürnkranz J. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In: *Semantic Processing of Legal Texts*, Lecture Notes in Computer Science, vol. 6036, Berlin/Heidelberg: Springer; 2010, 192–215.
- Mencía EL, Park SH, Fürnkranz J. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing* 2010, 73:1164–1176.
- Mitchell T. Generalization as search, *Artificial Intelligence*, 1982, 28:203-226.
- Montejo-Ráez A, Ureña López L. Selection strategies for multi-label text categorization. In: *Advances in Natural Language Processing*, Lecture Notes in Computer Science, vol. 4139; 2006, 585–592.
- Muslea I, Minton S, Knoblock C. Active + semi-supervised learning = robust multi-view learning. *Proceeding of ICML-02, 19th International Conference on Machine Learning*, 2002, 435–442.
- Nair Benrekia NY. Classification interactive multi-label pour l'aide à l'organisation personnalisée des données. Thèse de Doctorat en Informatique et applications de l'Univ. de Nantes, novembre 2015.

- Nardiello P, Sebastiani F, Sperduti A. Discretizing continuous attributes in adaboost for text categorization. In: *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 2633, Berlin/Heidelberg: Springer; 2003, 320–334.
- Nasierding G, Kouzani A. Image to text translation by multi-label classification. In: *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence*, Lecture Notes in Computer Science, vol. 6216, Berlin/Heidelberg: Springer; 2010, 247–254.
- Nguyen CD, Dung TA, Cao TH. Text classification for DAG-structured categories. In: *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, vol. 3518, chap. 36, Berlin/Heidelberg: Springer; 2005; 1–18.
- Nigam K. *Using unlabeled data to improve text classification* (Technical Report CMU-CS-01-126). Carnegie Mellon University. Doctoral Dissertation, 2001.
- Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. *Ninth International Conference on Information and Knowledge Management*, 2000, 86–93.
- Oza N, Castle JP, Stutz J. Classification of aeronautics system health and safety documents. *IEEE Trans Syst Man Cybern C Appl Rev* 2009, 39:670–680.
- Ozonat K, Young D. 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*. ACM.16,30, 2009, 1295-1304.
- Pachet F, Roy P. Improving multilabel analysis of music titles: a large-scale validation of the correction approach. *IEEE Trans Audio Speech Lang Proc* 2009, 17:335–343.
- Patra S, Bruzzone L. A fast cluster-assumption based active learning technique for classification of remote sensing images. *IEEE J. Sel. Topics Signal Process.*, vol.49, n°5, May 2011, 1617-1626.
- Patra S, Bruzzone L. A novel SOM-based active learning technique for classification of remote sensing images with SVM. in *Proc. IGARSS*, 2012, 6879-6882.
- Patra S, Bruzzone L. A novel SOM-SVM-based active learning technique for remote sensing image classification. *IEEE Trans. on Geoscience and Remote sensing* , vol.52, n°11, Nov. 2014, 6899-6910.
- Parpinelli R, Lopes H, Freitas A. Data mining with an ant colony optimization algorithm. *IEEE Trans. Evol. Comput.* 2002, 6:321–332.
- Peters S, Denoyer L, Gallinari P. Iterative annotation of multi-relational social networks. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*; 2010, 96–103.
- Platt J. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, 1999, 61-74.
- Quinlan JR. C4.5: Programs for Machine Learning. *Morgan Kaufmann*, 1993.
- Rabiner L. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 1989, 257–285.
- Rak R, Kurgan L, Reformat M. A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation. *Data Knowl Eng* 2008, 64:171–197.
- Rand WM. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66, 1971, 846-850.
- Read J. A pruned problem transformation method for multi-label classification. In: *Proceedings of the NZ Computer Science Research Student Conference*; 2008, 143–150.
- Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification, in: *Proceedings of the 20th European Conference on Machine Learning*, 2009, 254-269.
- Read J. Scalable multi-label classification. PhD Thesis, University of Waikato, 2010.
- Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Machine Learning* 2011, 85:1–27.
- Rigollet P. Generalization error bounds in semi-supervised classification under the cluster assumption, *Journal of Machine Learning Research*, 2007, vol. 8, pp. 1369-1392.
- Riloff E, Wiebe J, Wilson T. Learning subjective nouns using extraction pattern bootstrapping. *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*.
- Rogovschi N, Lebbah M, Bennani Y. Probabilistic mixed topological map for categorical and continuous data. In *ICMLA*, 2008, 224–231.

- Rogovschi N, Lebbah M, Grozavu N. Pondération et classification simultanée de données binaires et continues, in Proc. of the *EGC'11*, Brest, 25-28 janvier 2011–*RNTI, Revue des Nouvelles Technologies de l'Information*, Editions Hermann. 2011, 65-70.
- Rosenberg C, Hebert M, Schneiderman H. Semi-supervised selftraining of object detection models. *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.
- Roy N, McCallum A. Toward optimal active learning through sampling estimation of error reduction. In *Proc. of ICML*, 2001.
- Rubin T, Chambers A, Smyth P, Steyvers M. Statistical topic models for multi-label document classification. *Mach Learn* 2012, 88:157–208.
- Sapozhnikova E. ART-based neural networks for multi-label classification. In: *Advances in Intelligent Data Analysis VIII*, Lecture Notes in Computer Science, vol. 5772, Berlin/Heidelberg: Springer; 2009, 167–177.
- Schapire RE, Singer Y. BoosTexter: a boosting-based system for text categorization. *Machine Learning*; 2000, 39(2-3):135–168.
- Schohn G, Cohn D. Less is more: active learning with support vector machines. In *Proceedings of the seventeenth international Conf. on Machine Learning*. 2000, 839-846.
- Settles B. Active learning literature survey. Computer sciences technical report 1648. University of Wisconsin-Madison, 2012.
- Settles B, Craven M. An analysis of active learning strategies for sequence labeling tasks. In *Proc. of EMNLP*, 2008.
- Seung HS, Oppor M, Sompolinsky H. Query by committee. In *Proc. of the 5th annual workshop on computational learning theory (COLT'92)*, 1992, 287-294.
- Shao H, Li G, Liu G, Wang Y. Symptom selection for multi-label data of inquiry diagnosis in traditional Chinese medicine. *Sci China Ser F-Info Sci* 2010, 1:1–13.
- Shi C, Kong X, Yu P, Wang B. Multi-objective multi-label classification. In: *Proceedings of the SIAM International Conference on Data Mining*, Anaheim, CA, USA; 2012, 355–366.
- Singh M, Brew A, Greene D, Cunningham P. Score normalization and aggregation for active learning in multi-label classification, University College Dublin, Tech. Rep., 2010.
- Singla A, Patra S, Bruzzone L. A novel classification technique based on progressive transductive SVM learning. *Pattern recognition letters*, 42 (2014) 101-106.
- Sindhwani V, Keerthi SS. Large scale semi-supervised linear SVMs. in *Proceedings of the SIGIR*, 2006.
- Snoek CG, Worring M, Van Gemert JC, Geusebroek JM, Smeulders AW. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia*. ACM.16,30, 2006, 421-430.
- Sobol-Shikler T, Robinson P. Classification of complex information: inference of co-occurring affective states from their expressions in speech. *IEEE Trans Pattern Anal Mach Intell* 2010, 32:1284–1297.
- Sorower MS. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*.16,30, 2010.
- Spat S, Cadonna B, Rakovac I, Gütl C, Leitner H, Stark G, Beck P. Enhanced information retrieval from narrative german-language clinical text documents using automated document classification. In: *eHealth Beyond the Horizon—Get IT There, Proceedings of MIE2008, The XXIst International Congress of the European Federation for Medical Informatics*, Göteborg, Sweden; 2008, 473–478.
- Specht DF. Probabilistic neural networks. *Neural Networks*, 1990, 3:109–118.
- Spyromitros E, Tsoumakas G, Vlahavas I. An empirical study of lazy multi-label classification algorithms. In: *SETN'08: Proceedings of the 5th Hellenic Conference on Artificial Intelligence*, Berlin, Heidelberg; 2008, 401–406.
- Srivastava A, Zane-Ulman B. Discovering recurring anomalies in text reports regarding complex space systems, in: *Proceedings of the IEEE Aerospace Conference*, 2005, 55-63.
- Tang L, Liu H. Scalable learning of collective behavior based on sparse social dimensions. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*, New York, NY, USA; 2009, 1107–1116.

- Tang L, Rajan S, Narayanan VK. Large scale multi-label classification via metalabeler. In: *Proceedings of the 18th Intern. Conf. on World Wide Web (WWW'09)*, New York, NY, USA; 2009, 211–220.
- Tawiah CA, Sheng VS. A study on multi-label classification. In: *Advances in Data Mining. Applications and Theoretical Aspects*, Springer, 2013, 137–150.
- Thabtah FA, Cowling P, Peng Y, Rastogi R, Morik K, Bramer M, Wu X. MMAC: a new multi-class, multi-label associative classification approach. In: *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004*; 2004, 217–224.
- Thabtah FA, Cowling PI. A greedy classification algorithm based on association rule. *Appl Soft Comput* 2007, 7:1102–1111.
- Tian Y, Shi Y, Liu X. Recent advances on support vector machines research. *Technol. Econ. Div. Econ.* 18(1), 2012, 5–33.
- Tong S. *Active learning: theory and applications*. PhD thesis, Stanford university, CA, 2001.
- Tong S, Koller D. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2002, 2:45–66.
- Trohidis K, Tsoumakas G, Kalliris G, Vlahavas I. Multi-label classification of music into emotions. In: *Proceedings of the 9th Intern. Conf. on Music Information Retrieval (ISMIR 2008)*; 2008, 325–330.
- Tsoumakas G, Katakis I. Multi label classification: an overview. *Int J Data Warehouse and Mining*, 3(3):1–13, 2007.
- Tsoumakas G, Katakis I, Vlahavas I. Effective and efficient multilabel classification in domains with large number of labels. In: *Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008.
- Tsoumakas G, Dimou A, Spyromitros E, Mezaris V, Kompatsiaris I, Vlahavas I. Correlation-based pruning of stacked binary relevance models for multi-label learning. In: *Proceedings of the 1st International Workshop on Learning from Multi-Label Data (MLD'09)*, Bled, Slovenia; 2009, 101–116.
- Tsoumakas G, Katakis I, Vlahavas I. Random k-Labelsets for Multi-Label Classification. *IEEE Trans Knowledge Data Engineering*, 2010, 23:1079–1089.
- Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: *Data Mining and Knowledge Discovery Handbook, Part 6*. Springer, 2010, 667–685.
- Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, vol. 12, 2011, 2411–2414.
- Tuia D, Volpi M, Copa L, Kanevski M. A survey of active learning algorithm for supervised remote sensing image classification, *IEEE Journal of selected topics in signal processing*, sept. 2013.
- Tzanetakis G, Cook P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* 10(5), 2002, 293–302.
- Ueda N, Saito K. Parametric mixture models for multi-labeled text. In *Proceedings on Neural Information Processing Systems (NIPS)*; 2002, 721–728.
- Ukwatta E, Samarabandu J. Vision based metal spectral analysis using multi-label classification. In: *Canadian Conference on Computer and Robot Vision (CRV '09)*; 2009, 132–139.
- Vapnik VN. *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer-Verlag, 2001.
- Veloso A, Meira W Jr, Gonçalves MA, Zaki MJ. Multi-label lazy associative classification. In: *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Warsaw, Poland; 2007, 605–612.
- Villmann T, Der R, Hermann M, Martinetz TM. Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Trans. Neural Netw.* vol.8, n°2, mar. 1997, 256–266.
- Wagstaff K, Rogers S. Constrained K-means clustering with background knowledge, *Proceedings of the Eighteenth Intern. Conf. on Machine Learning*, 2001, 577–584.
- Wang H, Huang M, Zhu X. A generative probabilistic model for multi-label classification. In: *ICDM'08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Washington, DC, USA; 2008, 628–637.

- Wang J, Zhao Y, Wu X, Hua XS. A transductive multi-label learning approach for video concept detection. *Pattern Recogn* 2010, 44:2274–2286.
- Wang A, Wen J, Alam S, Jiang Z, Wu Y. Semi-supervised learning combining transductive support vector machine with active learning. *Neurocomputing*, 2015.
- Wieczorkowska A, Synak P, Ras Z. Multi-label classification of emotions in music, in: *Intelligent Information Processing and Web Mining*, Springer, Berlin/Heidelberg, 2006, 307–315.
- Wu TF, Lin CJ, Weng RC. Probability estimates for multi-class classification by pairwise coupling, *Journal of Machine Learning Research*; 2004, 5: 975–1005.
- Xu H, Xu J. Designing a multi-label kernel machine with two-objective optimization. In: *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI'10): Part I*, Berlin, Heidelberg; 2010, 282–291.
- Yan R, Yang J, Hauptmann A. Automatically labeling video data using multi-class active learning. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03)*, page 516, 2003.
- Yan Y, Fung G, Dy JG, Rosales R. Medical coding classification by leveraging inter-code relationships. In: *Proceedings of the 16th Intern. Conf. on Knowledge Discovery and Data Mining (KDD'10)*, New York, NY, USA; 2010, 193–202.
- Yang B, Sun JT, Wang T, Chen Z. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD Intern. Conf. on Knowledge Discovery and data mining, ser. KDD'09*. New York, NY, USA: ACM, 2009, 917-926.
- Yang Y. A study of thresholding strategies for text categorization. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, New York, NY, USA; 2001, 137–145.
- Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 1995, 189–196.
- Yearwood J, Mammadov M, Banerjee A. Profiling phishing emails based on hyperlink information. In: *International Conference on Advances in Social Networks Analysis and Mining*; 2010, 120–127.
- Yu K, Yu S, Tresp V. Multi-label informed latent semantic indexing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.16,30, 2005, 258-265.
- Zhang ML, Zhou ZH. Multi-label neural networks with applications to functional genomics and text categorization, *IEEE Transactions on Knowledge and Data Engineering*; 2006, 18(10): 1338–1351.
- Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*; 2007, 40(7):2038–2048.
- Zhang ML, Zhou ZH. A Review On Multi-Label Learning Algorithms. *IEEE Trans Knowledge Data Engineering* 2014, 26:1819–1837.
- Zhou T, Tao D, Wu X. Compressed labeling on distilled labelsets for multi-label learning. *Machine Learning* 2012, 88:69–126.
- Zhou ZH, Zhang ML, Huang SJ, Li YF. Multi-instance multi-label learning. *Artif Intell* 2012, 176:2291–2320.
- Zhu X. *Semi-supervised learning with graphs*. Doctoral dissertation, Carnegie Mellon University. CMU-LTI-05-192, 2005.
- Zhu X, Lafferty J, Ghahramani Z. *Semi-supervised learning: From Gaussian fields to Gaussian processes* (Technical Report CMU-CS-03-175). Carnegie Mellon University, 2003.
- Zhu X. *Semi-supervised learning literature survey* (Tech. Rep. 1530). Computer Sciences, University of Wisconsin – Madison, 2007.