REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE UNIVERSITE MOULOUD MAMMERI TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire

En vue de l'obtention du diplôme de master

Domaine : Mathématiques et Informatiques
Filière: Informatique
Spécalité : Conduite de projets informatique

Thème

Extraction de connaissance à partir des données multi-labels en utilisant l'analyse de concepts formels

Présenté par :

Melle BENZIANE Massissilia

Devant le jury composé de :

Promoteur : *Mr RADJA Hakim*

Président : *Mr* CHEBOUBA Lokmane

Examinateur : *Mr* SAIDANI Fayeçal

Année universitaire 2019/2020

Remerciements

Je tiens tout d'abord à remercier le bon dieu de m'avoir donnée le courage et la volante pour la réalisation de ce modeste travail.

Je tiens à remercier vivement mon enseignant et promoteur Mr RADJA Hakim, d'avoir accepté de m'encadrer tout au long de mon projet. Ainsi que pour sa disponibilité, son aide, ses conseils précieux et ses critiques constructives, ses explications et suggestions pertinentes.

Je remercie chaleureusement les membres de jury de m'avoir honoré en acceptant de juger ce projet de fin d'études.

J'exprime enfin mon infinie gratitude à mes chers parents et ma famille en reconnaissance de leurs sacrifices, aides, soutien et encouragements, et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Ces dernière années ont vu émerger différentes extensions du problème de classification classique. Parmi ces extensions se trouve la classification multi-labels[1]. Dans cette dernière, chaque instance peut appartenir à une ou plusieurs classes simultanément. Très souvent dans le cas d'application réelle, les classes ne sont pas indépendante les unes des autres. Chaque objet pouvant avoir plusieurs labels (classes), il semble pertinent de commencer par déterminer s'il existe un lien entre ces labels, cela en étudiant les corrélations entre ces classes.cette première étude permet de voir a quel point les classes sont structurés. Pour ensuite exploiter cette information. Entre autres, pour choisir une méthode de classification adaptée, prenant en compte ou non le lien entre les labels, ou bien corriger quelques résultats d'une première classification.

Pour atteindre cet objectif nous allons utiliser l'analyse de concepts formels (ACF)[10] qui est un outil mathématique très puissant permettant d'induire des paires de sous-ensembles ([objets],[propriétés]), appelées concepts formels, a partir d'une relation binaire entre un ensemble d'objets et un ensemble de propriétés. Elle a été utilisé dans divers domaines : psychologie, sociologie, biologie, médecine, linguistique, mathématiques, informatique. . .etc. les connaissances induites appelées concepts formels sont hiérarchisées et représentées sous la forme d'un treillis de Galois. Les treillis de Galois constituent un moyen efficace permettant d'avoir une représentation exhaustive de la réalité étudiée (contexte formel).

Dans la classification multi-labels la dépendance entre les classes existe, pour extraire ces relations existantes et les exploiter on utilise les règles d'associations triadiques (conditionnelles) extraites.

Le présent mémoire, comporte outre l'introduction générale, la conclusion et la bibliographie, les trois chapitres suivants :

Le chapitre 1 intitulé : «la classification multi-labels» nous introduisons la classification multi-labels et ses différentes approches. On va mettre en évidence l'existence de dépendances éventuelles entre les classes.

Dans le chapitre 2 intitulé : «l'analyse de concepts formels» nous introduisons l'ACF et nous présentons les mathématiques de cette théorie.
nous présentons une extension a l'ACF qui est l'ACF triadique. Le but d'une telle analyse est d'identifier des concepts et des implications triadiques que nous allons utiliser dans notre prochain chapitre.

Le troisième chapitre est réservé à notre approche. Dans ce chapitre nous allons utiliser l'analyse de concept formel triadique pour extraire toutes les règles d'associations triadiques(conditionnelles) capturant les relations entre les classes dans le domaine de la classification multi-labels .

Enfin, notre mémoire se termine par une conclusion générale qui rappelle la problématique que nous avons traité et les contributions que nous avons proposées pour la résoudre. Par la suite nous dressons une liste de quelques perspectives.

Chapitre 1

La classification multi-labels

1.1 Introduction

Ces dernières années, diverses extensions du problème de classification classique sont apparues[2], qui consiste à apprendre à partir d'un ensemble d'instances (objets, exemples) associés à un seul label w dans un ensemble disjoint de label Y, |Y|>1. Lorsque |Y|=2, on parle de classification binaire. Tandis que si |Y|>2, le problème d'apprentissage est appelé problème de classification multi-classes[3].

Parmi les extensions du problème de classification se trouvent les problèmes de classification multi-labels où chaque objet peut appartenir simultanément à plusieurs classes contrairement aux problèmes de classification mono-labels dans lesquels un objet appartient à une seule classe.

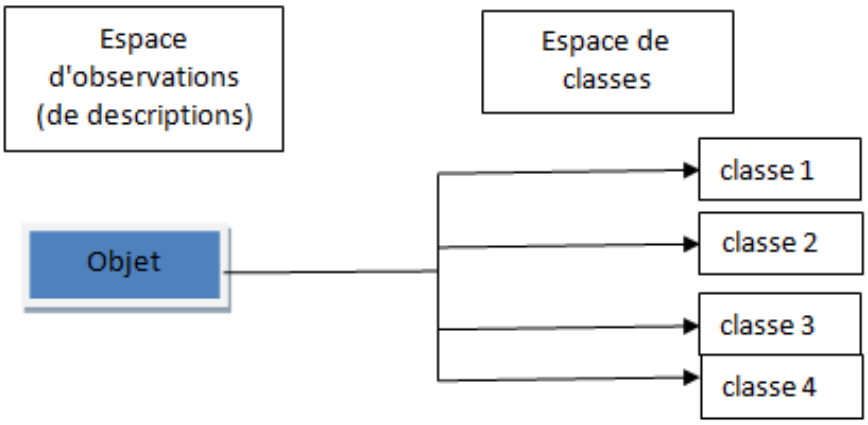
Dans la classification multi-labels, un objet peut être associé à un ensemble de label $\hat{Y} \subseteq Y$ où Y est l'ensemble fini de labels possibles. Dans le passé, la classification multi-labels a été principalement motivée par les tâches de catégorisation de textes et de diagnostic médical.

Les documents tests appartiennent généralement à plus d'une classe. Par exemple, un article de journal sur les réactions de l'église chrétienne à la sortie du film Davinci code peuvent être classées en deux catégories société/ religion et culture/films. De même dans le diagnostic médical, un patient peut être atteint de diabète et du cancer de la prostate simultanément[2].

Aujourd'hui, les méthodes de classification multi-labels sont de plus en plus utilisées dans les applications modernes, tels que la classification de la

fonction des protéines, la classification d'images, une photographie peut appartenir à plus d'une classe telles que couché de soleil, arbres, montagnes et plages simultanément . De la même façon, dans la classification de chanson, chaque morceau peut évoquer plusieurs émotions.

Donc pour résumer, la classification permet de catégoriser et de hiérarchiser des objets, ces objets appartiennent à un espace d'observation où ils sont décrits par des attributs ou des variables. Il seront localisés dans un espace de classes. Ce problème n'a de sens que si on dispose d'une correspondance entre les deux espaces observations et classes. Le problème de classification revient à estimer cette correspondance inconnue, ainsi la classification mono-label consiste à associer à chaque observation une et une seule classe tandis que la classification multi-labels consiste à associer à chaque observation une ou plusieurs classes simultanément. Alors le but sera de créer un classifieur à partir d'un ensemble d'apprentissage qui permet de prédire la ou les classes d'un nouvel exemple.

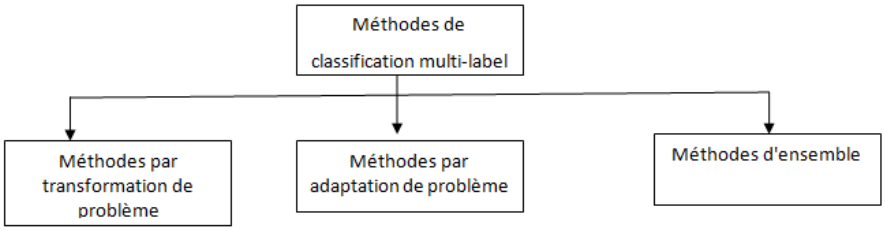


Ce type de classification est largement requise par plusieurs applications modernes telles que la classification d'images, annotation de vidéo...etc.
Dans la suite de ce rapport nous allons utiliser les notations suivantes :
Notation :
-Un objet x∈X,x=[x$_1, x_2, \ldots, x_d$] *où* x_i *est une variable d'observation qui peut être discrète ou continue.*
$-$ *Ensemble fini de classes possibles* $Y = (w_1, w_2, \ldots, w_Q)$
$-$*Ensemble de données d'apprentissage,* $D = ((x_i, Y_i)i = 1, \ldots, n)$ *où* $x_i \in X$ *et* $Y_i \subseteq Y$

$-\ Ensemble\ \ de\ \ données\ \ test\ \ I = ((x_j, Y_j)\ \ j = 1, \ldots, m)$
$-Classifieur\ \ multi-labels\ \ H : X \longrightarrow 2^Y (2^Y\ \ est\ \ l'ensemble\ \ des\ \ parties\ de\ \ Y).$
$-\ Prédiction\ \ Y = H(x)\ \ ensemble\ \ de\ \ classes\ \ prédites\ \ pour\ \ l'individux$
$.-Fonction\ de\ score\ (prédire\ le\ rang)\ pour\ associer\ un\ score\ pour\ chaque$
$classe\ \ prédite\ \ f : X \times Y \longrightarrow [0, 1]$

1.2 Méthodes de classification multi-label :

Le domaine de la classification multi-label est relativement récent. Il y a plusieurs méthodes proposées dans ce domaine, ces méthodes peuvent être divisées en trois grandes familles[4] : les méthodes par transformation de problème, méthodes par adaptation de problème et méthodes d'ensemble.



1.2.1 Méthodes par transformation de problème :

Ces méthodes transforment le problème d'apprentissage multi-labels en un ou plusieurs problèmes d'apprentissage mono-label. Pour illustrer ces méthodes, on va utiliser l'ensemble des données de la table 1.1 Il se compose de quatre instances (documents dans ce cas) qui appartiennent à une ou plusieurs classes : Sport, Religion, Sciences et Politique.

	Sport	Religion	Science	Politique
1	×			×
2			×	×
3	×			
4		×	×	

TABLE 1.1 – Données d'apprentissage

Il existe deux méthodes de cette approche[4] qui forcent le problème d'apprentissage multi-labels en un problème de classification mono-label. La première (PT1), sélectionne subjectivement ou de manière aléatoire l'un des labels de chaque instance et ignore le reste. Tandis que la seconde (PT2), ignore toute instance multi-labels. Les table 1.2 et table 1.3 montrent l'ensemble de données de la table 1.1 transformé en utilisant les méthodes PT1 et PT2.

	Sport	Religion	Science	Politique
1	×			
2				×
3	×			
4			×	

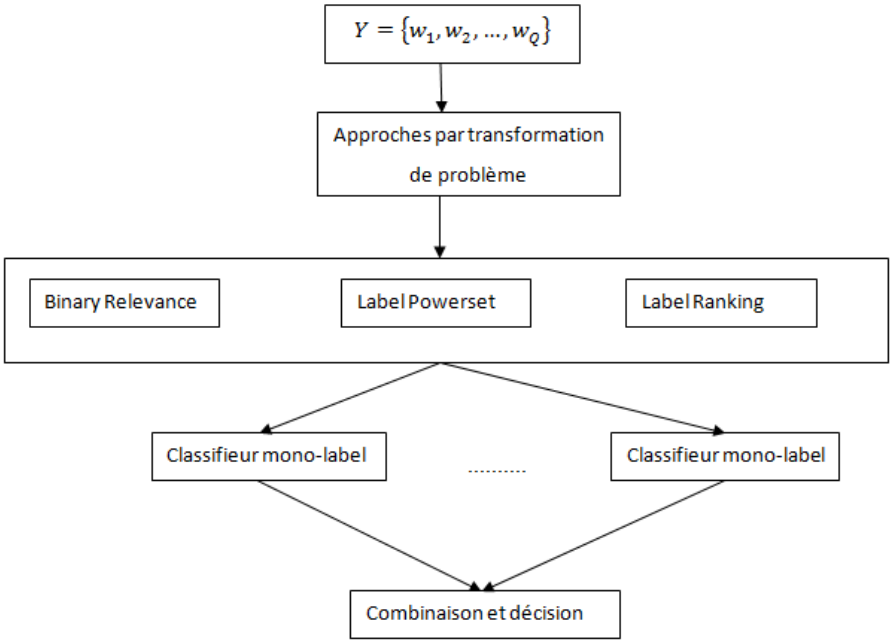
Table 1.2 – Données obtenues en utilisant PT1

	Sport	Religion	Science	Politique
3	×			

Table 1.3 – Données obtenues en utilisant PT2

Ces deux méthodes présentent un grand inconvénient puisqu'elles engendrent une perte d'information considérable[5]. On remarque en effet, que dans la Table 1.2 obtenue en appliquant la méthode PT1, l'instance 1 n'appartient pas à la classe Politique ce qui fausse l'information contenue dans les données initiales représentées par la TABLE 1.1 Idem pour les instances 2 et 4 avec les classes Science et religion respectivement. le même constat s'applique aux données de la TABLE 1.3 obtenue en appliquant la méthode PT2.
De ce fait, nous nous intéressons dans le cadre de ce rapport à trois méthodes de transformation de problème particulièrement répandues :

- Binary relevance,
- Label powerset, et
- Label ranking.



Ainsi pour chaque classifieur mono-label, un sous ensemble d'apprentissage est choisi de l'ensemble d'apprentissage initial, ensuite les sorties des différents classifieurs sont combinées pour prédire l'ensemble de classes pour un nouvel exemple x.

Binary Relevance :

C'est la méthode de transformation de problème la plus utilisée [5,6]. Elle associe à chaque classe possible un classifieur binaire. Cette méthode apprend |Y| classifieurs binaires
$H_w : X \longrightarrow (w, \neg w)$, *un pour chaque label w dans Y. Elle transforme l'ensemble d'apprentissage initial en* $|Y|$ *ensembles de données* D_w *qui contient toutes les instances de l'ensemble d'apprentissage initial étiquette par w si l'instance contient w et* $\neg w$ *sinon.*

Cette figure montre les quatre ensembles d'apprentissage construits par cette méthodes lorsqu'elle est appliquée à l'ensemble d'apprentissage initial représenté par la figure 1.1 :

	Sport	¬ Sport
1	X	
2		X
3	X	
4		X

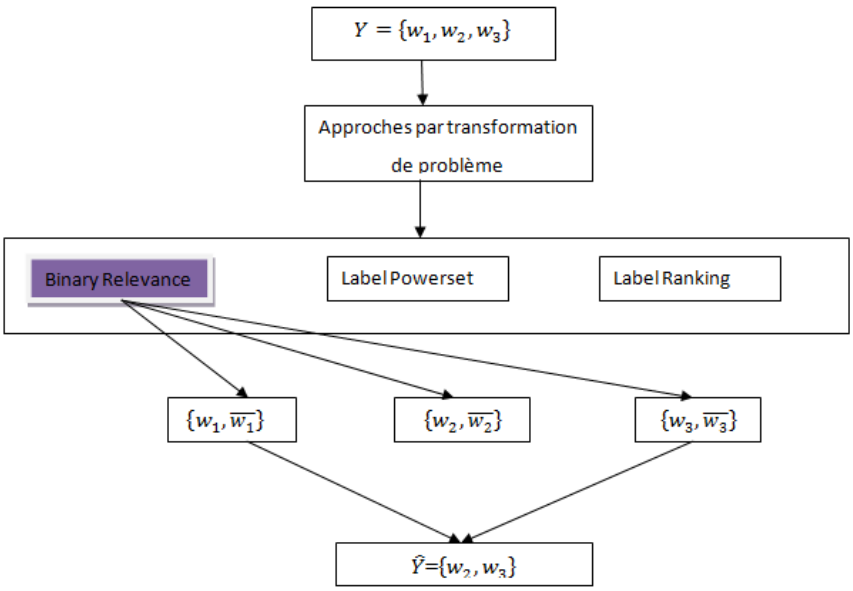
	Science	¬ Science
1		X
2	X	
3		X
4	X	

	Religion	¬ Religion
1	X	
2	X	
3		X
4		X

	Politique	¬ Politique
1	X	
2	X	
3		X
4		X

Pour la classification d'une nouvelle instance x, cette méthode retourne en sortie un ensemble de labels constitué de l'union des labels qui sont en sortie des |Y| classifieurs :

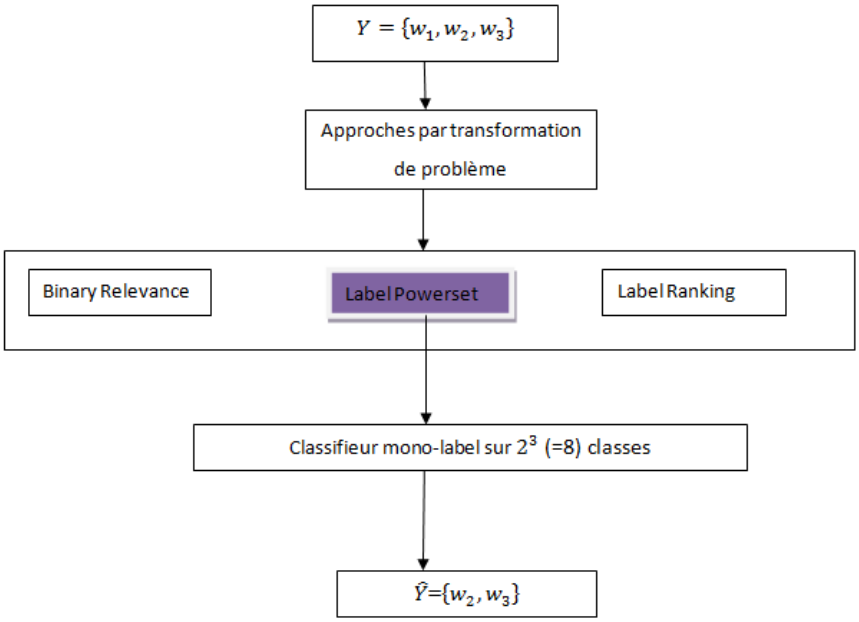
$$H_{BR} = \bigcup_{w \in Y} \{w\}, H_w(x) = w$$



Par exemple, si on dispose d'un ensemble de classes qui contient trois classes w_1, w_2 *et* w_3 *alors Binary Relevance crée trois classifieurs mono−label pour chaque label qui existe ainsi* :
le premier classifieur mono−label (binaire) permet de prédire si w_1 *appartient ou non* à *l'ensemble des classes prédites pour* x, *les autres classifieurs feront de même pour* w_2 *et* w_3.
Les trois sorties des classifieurs sont combinés pour former l'ensemble de classes Y.
La méthode Binary Relevance (BR) *a l'avantage d'*être *simple et peu couteuse en terme de temps de calcul mais son problème majeur réside dans le fait qu'elle ne tient pas compte des corrélations éventuelles entre les classes puisque cette méthode transforme le problèmes de classification multi−labels en Q problèmes de classification mono−labels indépendants*[6].

Label Powerset[4] :

La deuxième méthode de transformation de problème est Label Powerset. Cette méthode considère chaque combinaison de labels existant dans l'ensemble d'apprentissage comme une nouvelle classe. C'est à dire si on dispose de Q classes, l'ensemble de toutes les combinaisons possibles des labels qui existent dans l'ensemble d'apprentissage est au maximum égale à 2^Q :
$H : X \longrightarrow 2^Q Exemple$:
Supposons qu'on dispose de trois classes w_1, w_2 *et* w_3 *donc l'ensemble de toutes les combinaisons possibles dans l'ensemble d'apprentissage est* égale à $2^3 = 8(2^{|Y=3|})$.



La méthode Label Powerset traite le problème de classification multi label de la façon suivante :
Elle considère chaque ensemble de classes qui existe dans l'ensemble d'apprentissage multi label comme étant une seule classe dans une nouvelle tâche de classification mono label, comme le montre la figure suivante :

Exemple	Ensemble de labels
x_1	$\{w_1, w_4\}$
x_2	$\{w_3, w_4\}$
x_3	$\{w_1\}$
x_4	$\{w_2, w_3, w_4\}$

→

Exemple	Ensemble de labels
x_1	$w_{1,4}$
x_2	$w_{3,4}$
x_3	w_1
x_4	$w_{2,3,4}$

Pour une nouvelle instance, le classifieur mono-label de LP fournit en sortie la classe la plus probable, qui est en fait un ensemble de classes. Si ce classifieur peut fournir en sortie une distribution de probabilité sur toutes les classes, alors LP peut également classer les labels suivant la méthode suivante :

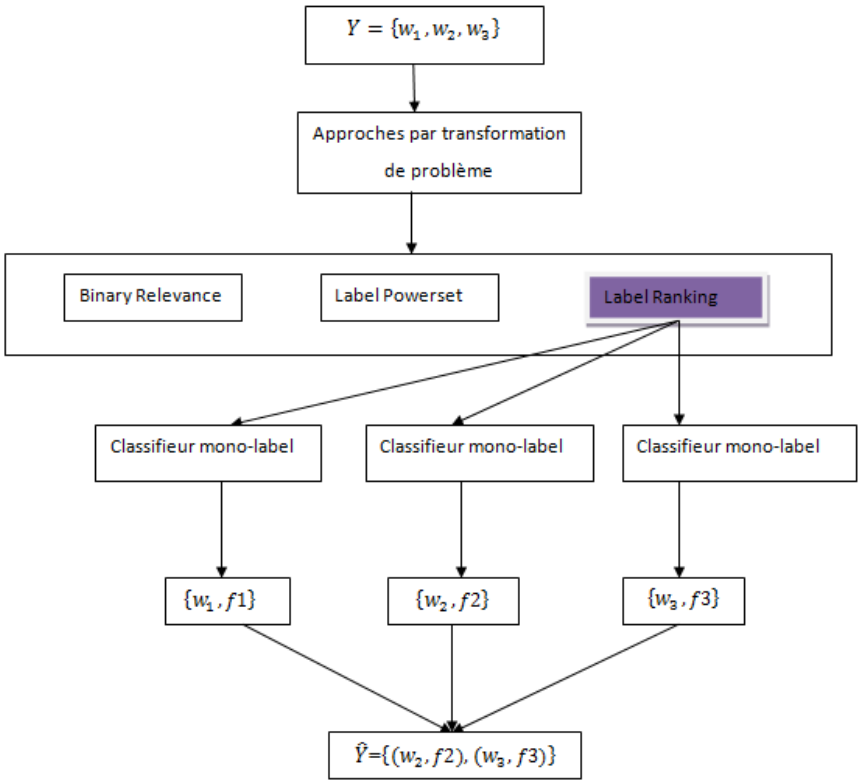
Pour un exemple donné, avec un ensemble de labels inconnu, pour obtenir un classement des labels (labels ranking), nous calculons la somme des probabilités des classes qui le contiennent. Comme le montre l'exemple suivant :

Classes (C)	$P(C/x)$	w_1	w_2	w_3	w_4
$w_{1,4}$	0,7	1	0	0	1
$w_{3,4}$	0,2	0	0	1	1
w_1	0,1	1	0	0	0
$w_{2,3,4}$	0,0	0	0	0	0
	$\sum_C P(C/x)\, w_j$	0,8	0	0,2	0,9

Cette approche a comme avantage de prendre en compte les corrélations entre les différentes classes, mais sa complexité est exponentielle au nombre de classes.

Label Ranking[7] :

Appartient aussi à l'ensemble des méthodes par transformation de problème, elle fait l'apprentissage sur des classifieurs mono-label. Par contre, cette approche permet un classement pour tout les labels appartenant à l'ensemble des labels.



Exemple :
Etant donné un ensemble de cinq documents possibles (champs d'application) suivants : Mathématique, Physique, Chimie, Biologie et Informatique, (M,P,C,B,I) un document peut relever de plusieurs catégories comme Mathématique, Physique et Informatique(M ,P,I). Du point de vue du classement (Ranking), on peut associer à tout document un ordre sur l'ensemble du champs d'application. Par exemple un document x peut appartenir plus à la classe M que P que I.

Pour résoudre ce problème, différentes méthodes existent. une approche particulière est celle dite préférence par paires qui consiste à apprendre pour chaque couple de labels $(\mathrm{w}_i, w_j)telque\ i < j$, *un classifieur binaire* H_ij *permettant de prédire pour une entrée* x *si* w_i *est préféré ou pas à* w_j. *A noter que n'importe quel classifieur binaire peut être utilisé dans ce cas et que la sortie du classifieur n'est pas nécessairement* $(0, 1)$, *elle est généralement comprise dans l'intervalle* $[0, 1]$. *H*ûllermier *propose d'associer à chaque entrée* x *une relation floue de préférence* R_x *définie comme suit :*

$$R_x = \begin{cases} H_{ij} & \text{si} \quad W_i > W_j, \\ 1 - H_{ij} & \text{sinon .} \end{cases}$$

Moyennant ces relations de préférences, on calcule pour chaque label $\mathrm{w}_i, la fonction$

$$\mathrm{S}_x(W_i) = \sum_{i\#j} R_x(W_i, W_j)$$

Pour une instance x, l'ordre total est obtenu en triant par ordre décroissant les fonctions $\mathrm{S}_x(w_i)$ *calculées pour chaque élément de Y.*

Méthode Pairwise binary[5] :

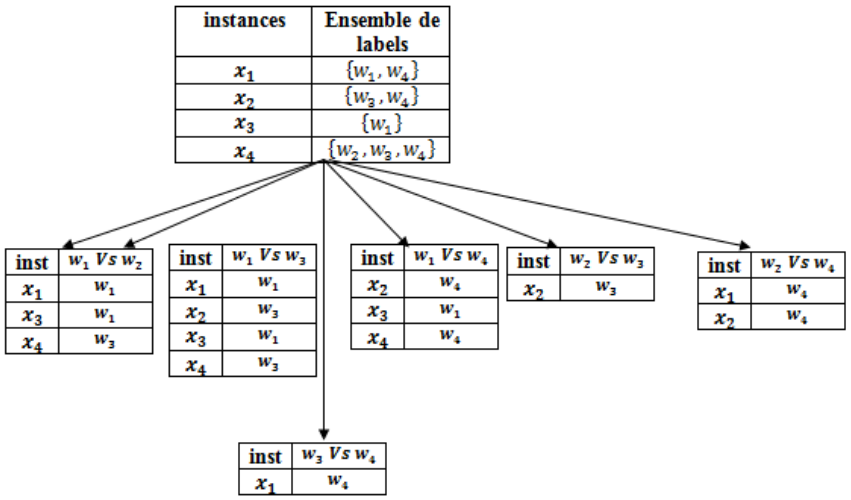
C'est une autre méthode de classification multi-labels par transformation de problème, qui procède de la façon suivante :

Partitionnement du problème en (Q(Q-1))/2 (Q est le nombre total de classes) sous problèmes impliquant seulement deux labels.

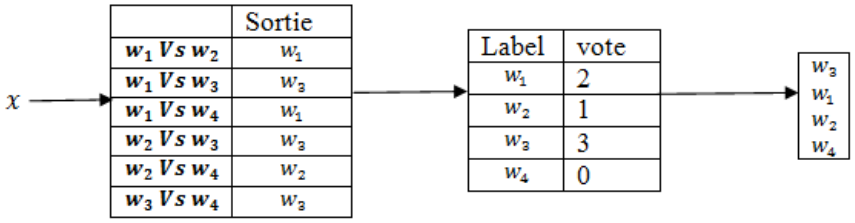
Construire un classifieurs sur toutes les paires de labels, avec seulement les individus de l'un ou l'autre label, c'est à dire que chaque modèle est formé par au moins un label mais pas les deux.

Combiner les sorties des (Q(Q-1))/2 classifieurs binaires pour obtenir les probabilités a postériori des labels.

Exemple

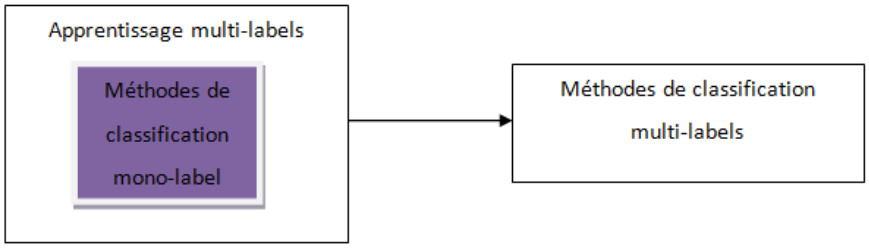


Pour une nouvelle instance x, le schéma suivant nous montre les étapes de la classification en appliquant la méthode Ranking by pairwise comparison :



1.2.2 Méthodes d'adaptation de problème

Ces méthodes permettent d'adapter les méthodes de classification mono-label existantes au domaine d'apprentissage multi labels afin de fournir des méthodes de classification multi-labels.



Ainsi,
KNN⟶ MLKNN
SVM⟶ Rank SVM
C4.5⟶ ML C4.5
Par exemple, l'algorithme C4.5 a été adapté pour l'apprentissage multi-labels. La formule de calcul de l'entropie a été modifiée de la façon suivante :

$$Entropie = \sum_{i=1}^{N} P(C_i)\log P(C_i) + q(C_i)\log q(C_i)$$

Où P(C_i) *est la fréquence relative de la classe* $q(C_i)$ *et* $q(C_i) = 1 - P(C_i)$. *Et plusieurs labels sont prévus dans les feuilles de l'arbre. Adaboost.MH et Adaboost.MR*[9] *sont deux extensions de Adaboost pour la classification multi−labels. Les deux appliquent Adaboost sur les classifieurs simple de la forme* $H : X \times Y \longrightarrow R$. *Dans Adaboost.−MH si le signe de la sortie du classifieur simple est positive pour une nouvelle instance* x *et un label* w *alors nous considérons que cette instance peut être étiquetée par* w, *par contre si la sortie est négative alors l'instance ne peut pas être étiquetée par le label* w. *Dans Adaboost.MR la sortie des classifieurs est considérée pour classer chacun des labels dans* Y.

Bien que des deux algorithmes sont des adaptations d'une approche spécifique de l'apprentissage, nous remarquons qu'ils utilisent en fait une transformation du problème, chaque exemple (x_i, Y_i) *est décomposé en* $|Y|$ *exemples* $(x_i, w, Y_i[w])$, *pour tout* $w \in Y$ *où* $Y_i[w] = 1$ *si* $w \in Y_i$ *et* $Y_i[w] = -1$ *sinon. La table suivante montre*

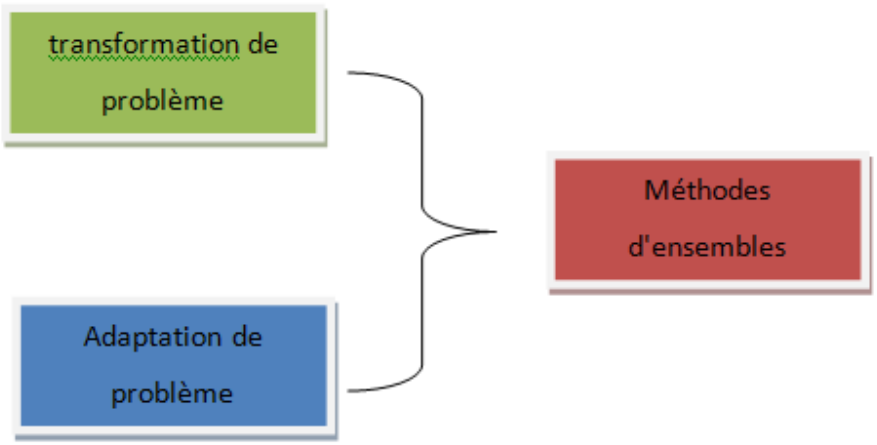
un exemple de cette transformation.

Ex	Label w	$Y[w]$
1	Sports	1
1	Religion	-1
1	Sciences	-1
1	Politique	1
2	Sports	-1
2	Religion	-1
2	Sciences	1
2	Politique	1
3	Sports	1
3	Religion	-1
3	Sciences	-1
3	Politique	-1
4	Sports	-1
4	Religion	1
4	Sciences	1
4	Politique	-1

ML-kNN[8] est une adaptation de l'algorithme kNN pour les données multi-lables. Cette méthode suit le paradigme de Binary Relevance (BR), ML-kNN utilise l'algorithme kNN indépendamment pour chaque label w, il trouve les k exemples les plus proches de l'instance test et considère ceux qui sont étiquetés avec w comme positifs et le reste négatifs. Ce qui différencie cette méthode de l'algorithme kNN original appliqué au problème ; transformé à l'aide de BR est l'utilisation des probabilités a priori. ML-kNN a également la capacité de produire un classement des labels en sortie.

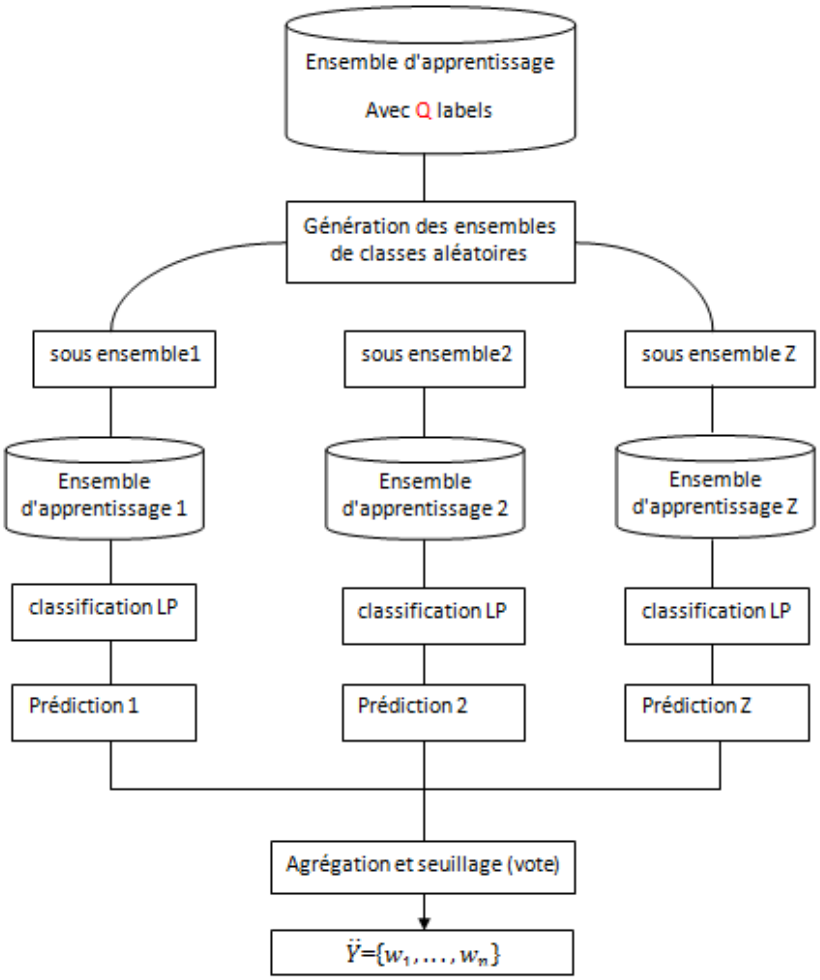
1.2.3 Méthodes d'ensemble :

utilisation des méthodes de transformation et d'adaptation de problème afin de fournir un classifieur multi-labels alors on combine les deux groupes qui existent avant :



Méthode RaKel (Random k label set)[1,7]

Le principe de cette méthode est de générer à partir d'un ensemble d'apprentissage de Q labels des sous ensembles choisis aléatoirement de l'ensemble total de labels de petite taille par rapport à Q (qui est la taille de l'ensemble de tous les labels). Ainsi pour chaque sous ensemble aléatoirement choisi, un sous ensemble d'apprentissage est extrait de l'ensemble d'apprentissage initial, une méthode de classification basée sur l'approche Label Powerset (LP) est appliquée sur l'ensemble d'apprentissage choisi afin de fournir la prédiction dans le même sous ensemble aléatoire, une fois que toutes les combinaisons sont prédites, on utilise la stratégie de vote pour combiner les sorties des différents classifieurs ainsi on peut prédire l'ensemble de classes Y pour un individu x.



Supposons que le nombre total de classes est égale à 6 de w_1àw_6 :
Le nombre de classes choisies dans chaque sous ensemble aléatoirement est égale à 3, $k = 3$.
Le nombre de classifieurs est $m = 5$.
Le seuil de vote $= 0,5$

Dans cette approche, nous avons besoin de trois paramètres :
1 Le nombre de classifieurs utilisé ici est Z, il faut l'optimiser au début,
2 Le nombre de classes dans chaque sous ensemble aléatoire,
3 Le seuil lors de la stratégie de vote.

Exemple
Supposons

Classifieurs	Ensemble aléatoire de labels	w_1	w_2	w_3	w_4	w_5	w_6
h_1	$\{w_1, w_2, w_3\}$	1	-1	1	0	0	0
h_2	$\{w_2, w_3, w_5\}$	0	1	1	0	-1	0
h_3	$\{w_3, w_5, w_6\}$	0	0	-1	0	-1	1
h_4	$\{w_1, w_4, w_6\}$	1	0	0	-1	0	1
h_5	$\{w_4, w_5, w_6\}$	0	0	0	-1	1	1
Moyenne des votes		$^2/_2$	$^1/_2$	$^2/_3$	$^0/_2$	$^1/_3$	$^2/_3$
$\hat{Y}$		1	1	1	-1	-1	1

Par exemple pour le classifieur h_1, *trois classes sont choisies aléatoirement* w_1, w_2 *et* w_3, *un classifieur basé sur l'approche LP permet de nous prédire si x appartient ou non* à *une de ces trois classes. Par exemple* h_1 *nous dit que x appartient aux classes* w_1 *et* w_3 *mais pas* à *la classe* w_2 *et il n'a pas décidé pour* w_4, w_5 *et* w_6. *Pareil pour* h_2 *qui a fait l'apprentissage pour les classes* w_2, w_3 *et* w_5, *il nous a dit que* w_2 *et* w_3 *appartiennent* à *l'ensemble des classes de x mais pas* w_5 *et pas de décision pour les autres classes.*
On fait de manière à *faire apparaitre toutes les classes au moins dans un classifieur. Ici nous avons choisi cinq classifieurs, etant donné un individux, pour prédire l'ensemble des classes, le nombre de votes positif est moyenné parle nombre total de vote pour chaque classe* :

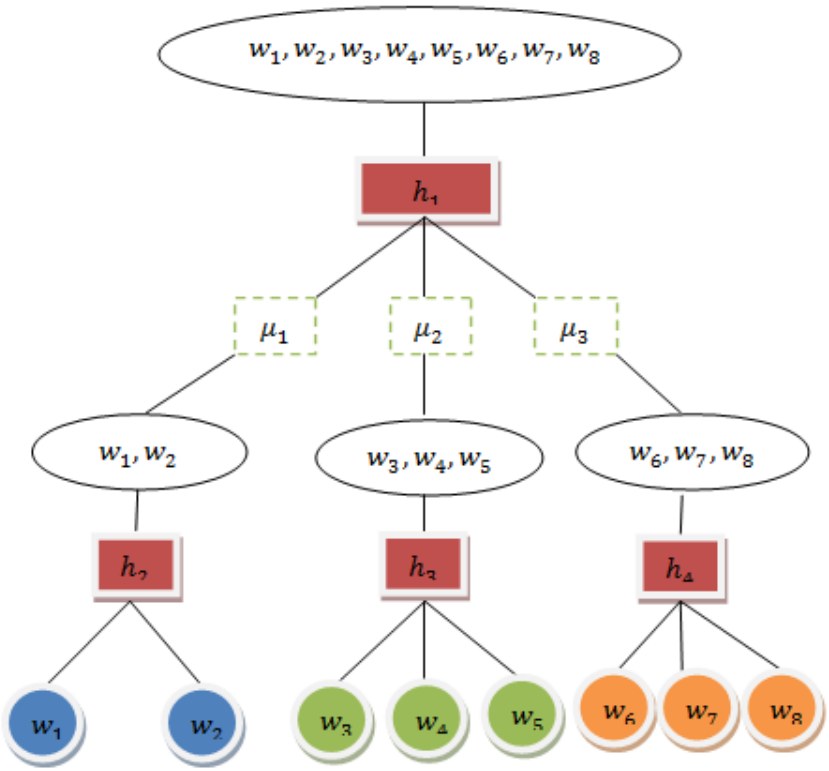
+*Par exemple pour* w_2, *deux votes sont donnés par les classifieurs* h_1 *et* h_2 *dont un est positif alors on a* $1/2$.
+ *Pour* w_3 *nous avons* $2/3$ *votes qui sont positifs.*
En utilisant un seuil $= 0,5$, *on peut prédire l'ensemble des classes pour l'individu x.*

Cette méthode réduit la complexité de la méthode Label Powerset qui crée un classifieur mono label sur au maximum 2^Q (Q *étant le*

nombre total de classes).
La méthode Rakel crée des classifieurs sur 2^K, $K \ll Q$ (K *très petit par rapport* à Q), *aussi Rakel a l'avantage de tenir compte des corrélations éventuelles entre classes.*
Par contre cette méthode induit une perte d'informations puisqu'elle réalise l'apprentissage sur des sous ensembles de labels. Un autre inconvénient est qu'on a trois paramètres à *optimiser*

Méthode HOMER :

L'algorithme HOMER[4,2] construit une hiérarchie de classifieurs multi-labels, chacun traitant avec un ensemble beaucoup plus petit de labels comparé à l'ensemble de tous les labels Y avec une répartition plus équilibrée des instances.
L'idée principale de cette méthode est la transformation d'une tâche de classification multi-labels avec un grand ensemble de labels de taille |Y| en une hiérarchie, en forme d'arbre, de tâches de classification multi-labels plus simple, chacune traitant un nombre plus petit de labels K, K≪|Y|.
Chaque nœud n de cet arbre contient un sous ensemble de labels $Y_n \subset Y, il\ y\ a\ |Y|\ feuilles$, *chacune contenant un singleton* $w_j, j = 1, \ldots, |Y|$ (*chaque feuille représente un label* w_j *de* Y. *Chaque noeud interne* n *contient l'union des ensembles de labels de ses enfants*, $Y_n = \cup Y_c,\ c \in enfant(n)$. *La racine contient tous les labels* $Y_root = Y$.
Le concept de méta−label d'un noeud n, μ_n, *est défini comme étant la disjonction des labels contenus dans ce noeud*, $\mu_n = \bigvee w_j,\ w_j \in Y_n$. *Les méta − labels ont la sémantique suivante* :
une instance d'apprentissage peut être *annotée avec un méta − label* μ_n, *si elle est annotée avec au moins un labels de* Y_n.
Chaque noeud interne n *de la hiérarchie contient* également *un classi− fieur multi − labels* h_n. *La tâche de* h_n *est la prédiction de un ou plusieurs des méta−labels de ses enfants. Par conséquent*, *l'ensemble de labels pour* h_n *est* $M_n = (\mu_c\ \ c \in enfant(n))$. *La figure suivante montre une simple hiérarchie pour un problème de classi− fication multi − labels avec* 8 *labels* $w_1, \ldots, w_8$.



Pour une nouvelle instance x, HOMER commence avec h_{root} *et suit un processus récursif et transfère* x *au classifieur multi − label d'un noeud enfant* h_c *uniquement si* μ_c *est parmi les prédictions de* $h_{(parent(c))}$.
Finalement, ce processus peut conduire à la prédiction d'un ou de plusieurs labels par le classifieur multi labels juste au dessus de la feuille correspondante, l'union de ces labels prédits forme la sortie.

1.3 Les métriques d'évaluation[4] :

Nous introduisons d'abord les concepts de cardinalité et de densité de labels :
Soit D un ensemble de données multi-labels constitué de |D| instances multi-labels $((x_i, Y_i)i = 1, \ldots, |D|)$.

1.3.1 Définition 1 :

La cardinalité de label de D est le nombre moyen de labels dans les instances de D :

$$\text{LC(D)}=\frac{1}{|D|}\sum_{i=1}^{|D|}|Y_i|$$

1.3.2 Définition 2 :

La densité est le nombre moyen de label divisé par le nombre total de labels :

$$\text{LC(D)}=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i|}{Q}$$

La classification multi-labels nécessite des métriques différentes de celles utilisées dans la classification mono-label traditionnelle. Nous présentons dans ce qui suit quelques métriques utilisées :

1.3.3 Coût de Hamming (Hamming Loss) :

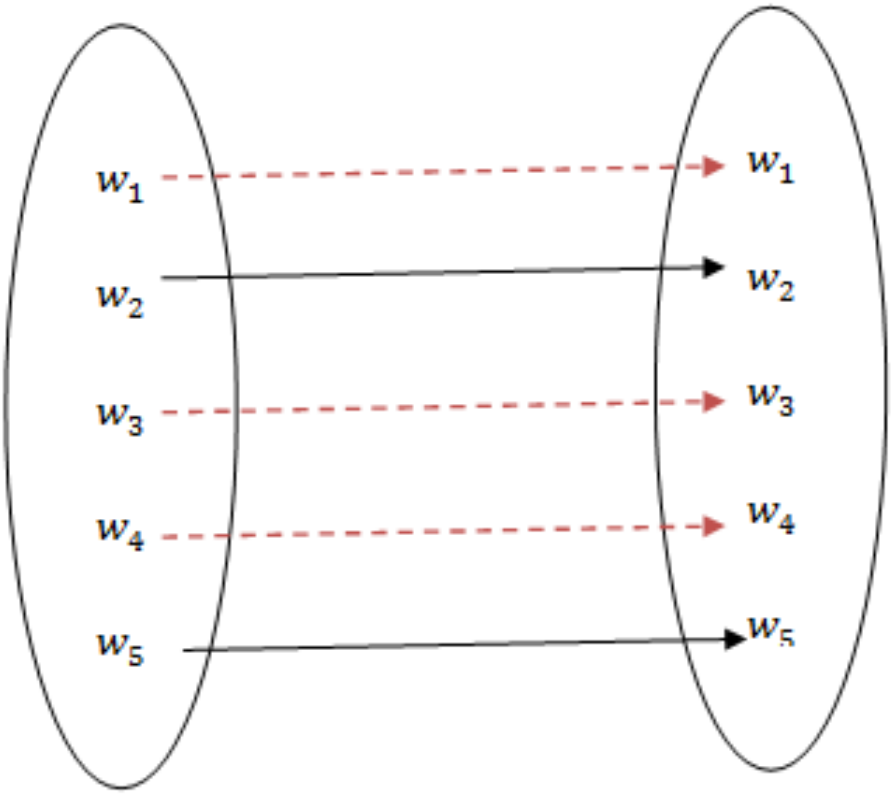
Ce critère évalue combien de labels sont mal classés (un label n'appartenant pas à Y_i *est prédit ou bien un label appartenant* à Y_i *qui n'est pas prédit*)

$$\text{Hamming Loss(H,D)}=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i\Delta\hat{Y}_i|}{Q}$$

Où : Δ est la différence symétrique entre deux ensembles

Remarque

Plus petite est la valeur du coût de Hamming, plus grande est la performance.
Dans l'exemple suivant, un individu x. L'ensemble de vraies classes Y est w_1, w_2 *et* w_5 *tandis que l'ensembles de classes prédites contient* w_2, w_3, w_4 *et* w_5
Le classifieur a commis trois erreurs, on a :
La classe w_1 *appartient* à *l'ensemble des vraies classes et qui n'est pas prédite par le classifieur.*
Les classes w_3 *et* w_4 *qui sont prédites par le classifieur tandis qu'elle n'appartiennent pas* à *l'ensemble des vraies classes.*



Donc le coût de Hamming pour un seul exemple x est égale à :

$$\text{Hamming Loss(H,D)} = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta \hat{Y}_i|}{Q} = \frac{3}{5}$$

1.3.4 Précision

Cette métrique ùesure le degré de similarité entre $Y_i et(\hat{Y}_i)$

$$\text{Precision(H,D)} = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}$$

Dans l'exemple précédant la précision est égale à : 2/5
Remarque :
Contrairement au coût de Hamming, plus grande est la valeur de la précision, plus grande est la performance.

1.3.5 Recall

$$\text{Recall(H,D)} = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i|}$$

Dans l'exemple précédent

Recall(H,D)= $\frac{2}{3}$

1.4 Etudes de corrélations entre les classes pour chaque méthode[5]

Ci-dessous un tableau (Figure 1.1) qui explique l'idée principale de chaque méthode (classifieurs) des trois approches de classification multi-labels et aussi la corrélation entre les classes.

On modélise la corrélation par un signe plus (+) c'est a dire le moins (-) modélise « le classifieur ne prend pas en compte la corrélation (dépendance) entre les classe » ; signe (+) signifie « le classifieur prend en compte la corrélation entre les classes » ;et le (++) signifie «forte exploitation des dépendance entre les classes».

APPROCHE	METHODE	Idée principale	Modélisation des corrélations entre les classes
Par transformations	BR	Apprend q classifieurs binaires	–
	CC	Apprend q classifieurs binaires dans une chaine	++
	LP	Apprend un seul classifieurs multi-classes	++
	CLR	Apprend q(q+1)/2 classifieurs binaires	+
	HOMER	Apprend une hiérarchie de classifieurs multi-classes	++
Par adaptation	ML-KNN	Combine kNN avec une inférence bayésienne	–
	IBLR_ML	Combine kNN avec une régression logistique	++
Ensembles	RAkEL	Apprend N classifieurs multi-classes	++
	ERB	Apprend N classifieurs BR	–
	ECC	Apprend N classifieurs CC	++
	RF-PCT	Apprend N arbres de décisions multi-label	++

FIGURE 1.1 – principe et études de corrélations pour chaque méthode

1.5 Conclusion

Nous avons décrit dans ce chapitre la classification multi-labels, on a cité ses différentes approches et on a parlé des métriques d'évaluations.
Ce travail a permis de mettre en évidence l'existence de relations (corrélations) entre les classes et l'étude des méthodes des approches proposées sur le fait qu'elles tiennent en compte ou pas des relations éventuelles entre les classes.

Chapitre 2

Analyse de Concepts Formel (ACF)

2.1 Introduction

L'Analyse de Concepts Formelle (ACF)[10] (appelée aussi Analyse Formels de Concepts (AFC)) est un formalisme mathématique pour l'analyse de données, la représentation de connaissances et la visualisation de connaissances. L'idée de base de l'ACF est d'extraire des concepts regroupant des objets et leurs propriétés/attributs à partir de données et de construire une hiérarchie à partir de ces concepts.

L'Analyse de Concepts Formels (ACF) a été introduite par Wille en 1982, puis consolidée mathématiquement par Ganter et Wille[10].

L'ACF consiste à induire des paires de sous-ensembles ((objets),(propriétés)), appelées concepts formels, à partir d'une relation binaire entre un ensemble d'objets et un ensemble de propriétés. Elle a été utilisée dans divers domaines : psychologie, sociologie, biologie, médecine, linguistique, mathématiques, informatique, etc...

Nous commençons notre étude par une présentation intuitive de l'ACF sans pour autant introduire de formalisme mathématique. Nous consacrerons par la suite une section complète pour la présentation théorique de l'ACF sur la base de fondements mathématiques (algébriques).

Nous allons rappeler l'analyse de concepts formels triadique en mettant davantage l'accent sur les implications triadiques(conditionnelle) qui traite

les relations ternaires ou plus généralement les relations n-aires mettent en lien trois ou plusieurs dimensions (axes d'analyse) et peuvent cacher des motifs intéressants sous forme de groupes homogènes ou associations entre les éléments des dimensions contrairement à l'ACF qui mit en lien deux dimensions et qui utilise des implications classique.Ensuite, nous allons présenter la solution pour l'extraction des règles d'associations triadiques et de ces implications.

2.2 Analyse de concepts formels (ACF)

L'idée de Wille[10] était d'illustrer dans un premier temps les paires (objets/propriétés) ainsi que la relation d'incidence selon une représentation mathématique appelée contexte formel. Ce dernier est considéré comme un outil de description des situations élémentaires sous la forme : l'objet "x" possède la propriété "y". Dans un deuxième temps, il s'agit de découvrir les ensembles maximaux d'objets satisfaisant un certain ensemble de propriétés.
Les connaissances induites appelées concepts formels sont hiérarchisées et représentées sous la forme d'un treillis de Galois. Les treillis de Galois constituent un moyen efficace permettant d'avoir une représentation "condensée" de la réalité étudiée (contexte formel) tout en gardant son informativité.

Exemple
Soit un ensemble de planètes du system solaire "mercure, venus, terre, mars, Jupiter, saturne, Uranus, Neptune, pluton" décrit par rapport à certaines de leurs propriétés taille (petite, moyenne, grande), distance au soleil (proche, loin), satellite (oui, non). Le contexte formel noté C (autrement dit la relation binaire) est représenté sous forme d'une table (TABLE 2.1) avec en lignes les planètes (correspondant aux objets) et en colonnes les propriétés, telle que si l'objet "xi" vérifie (resp. ne vérifie pas) la propriété "aj" alors la cellule "cij" est marquée par une croix (resp. reste vide).

Pour expliquer la notion de concept formel, nous considérons toutes les propriétés du Venus (taille :petite, distance au soleil :proche) et posons-nous la question : quelles sont les planètes satisfaisant ces propriétés ? Nous obtenons alors l'ensemble X= (venus, mercure, terre, mars).
Nous constatons que l'ensemble X est l'ensemble maximal des planètes satisfaisant toutes les propriétés de l'ensemble A= (taille :petite, distance au soleil :proche). Il résulte que X est l'ensemble de tous les planètes vérifiant toutes les propriétés de A et A est l'ensemble de toutes les propriétés véri-

Objet \ Attribut	Taille			Distance au soleil		Satellite	
	petite	moyenne	grande	proche	loin	oui	non
Mercure	×			×			×
Vénus	×			×			×
Terre	×			×		×	
Mars	×			×		×	
Jupiter			×		×	×	
Saturne			×		×	×	
Uranus		×			×	×	
Neptune		×			×	×	
Pluton	×				×	×	

TABLE 2.1 – Un contexte formel représentant les planètes du système solaire.

fiées par toutes les planètes de X. La paire(X,A) est appelé **concept formel**. Tandis que X est appelé **extension** et A est appelé **intension**.

Hiérarchie entre concepts formels Entre les concepts formels, il y a une relation d'ordre partiel c.à.d. la relation "Sous-concept, Super-concept". Etant donné deux concepts formels (X,A) et (Y,B). On dit que (X,A) est un sous-concept de (Y,B), (dualement (Y,B). est un super-concept de (X,A)) si l'extension de (X,A) est un sous ensemble de l'extension de (Y,B). c.à.d. $X \subseteq Y$(dualement : l'intension de (X,A) est un sur-ensemble de l'intension de (Y,B). c.à.d. $A \supseteq B$). Reprenons l'exemple précédent. Etant donné les deux concepts formels suivants : ((mercure),(taille :petite, distance au soleil :proche, satellite :non)) et ((mercure, venus),(taille :petite, distance au soleil :proche)). ((mercure),(taille :petite, distance au soleil :proche, satellite :non)) et est un sous concept de ((mercure, venus),(taille :petite, distance au soleil :proche)). Dualement, ((mercure, venus),(taille :petite, distance au soleil :proche)) est un super-concept de ((mercure),(taille :petite, distance au soleil :proche, satellite :non)). L'extension (mercure) du sous-concept est un sous ensemble de l'extension (mercure, venus) du super-concept. De la même manière l'intension (taille :petite, distance au soleil :proche, satellite :non) du sous-concept est un sur-ensemble de l'intension (taille :petite, distance au soleil :proche) du super-concept. Dans la Figure 2.1, nous représentons la hiérarchie de tous les concepts formels du contexte représenté dans la TABLE 2.1.

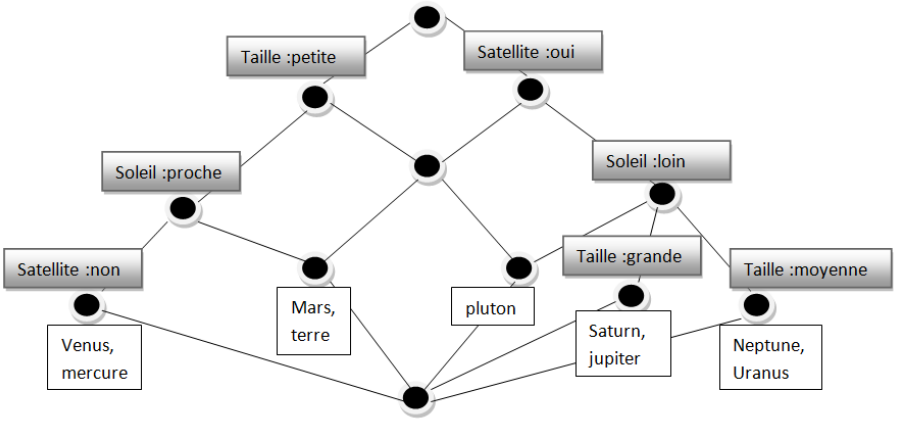


FIGURE 2.1 – Hiérarchie(treillis) de concepts formels

Comment lire la figure 2.1

Cette Figure est constituée de nœuds et de segments. Elle comporte aussi les noms de tous les objets et toutes les propriétés du contexte. Chaque nœud correspond à un concept formel. La Figure peut être lue comme suit : A chaque fois qu'un nœud "n" est étiqueté par une propriété "a", tous les objets descendants de ce nœud "n" héritent la propriété "a". De façons duale, à chaque fois qu'un nœud "n" est étiqueté par un objet "x", "x" est hérité vers le haut et tous les ancêtres du nœud "n" le partagent. Ainsi l'extension "X" d'un concept formel (X,A) correspondant au nœud "n" est obtenue en considérant tous les objets qui apparaissent sur les descendants du nœud "n" dans le treillis et son intension A est obtenue en considérant toutes les propriétés qui apparaissent sur les ancêtres du nœud "n" dans le treillis.
Nous pouvons remarquer que dans cet exemple, l'intension du concept formel sommet correspond à l'ensemble vide, tandis que l'extension du concept formel sommet correspond à l'ensemble de toutes les planètes. Nous pouvons trouver toute fois un concept formel sommet différent de l'ensemble vide.

2.2.1 Rappels mathématiques

Ordre et ordre partiel

Relation binaire

Une relation binaire R entre deux ensembles O et P est un ensemble de couples d'éléments (x,a) tels que x∈O et a∈ P, autrement dit un sous ensemble du produit Cartésien O×P. (x ,a)∈R (aussi noté par x R a) signifie que l'élément x est en relation R avec l'élément a. Si O=P, on parle de relation binaire sur O.
R^{-1}*est la relation inverse de* R, *i.e. la relation entre* P *et* O *telle que* $aR^{-1}x \Leftrightarrow xRa$.

Relation d'ordre (partiel)

Une relation binaire R sur un ensemble E est dite relation d'ordre partiel (ou simplement relation d'ordre) sur E si elle vérifie les conditions suivantes pour tout x,y,z ∈ E:
1. (x,x)∈R (R est réflexive)
2. si (x,y)∈ R et (y,x)∈R alors x=y (R est antisymétrique)
3. si (x,y)∈ R et (y,z)∈R alors (x ,z)∈R (R est transitive)
Une relation d'ordre R est souvent notée par ≤ (R^{-1})*estnotéepar* ≥).

Ensemble ordonné
Un ensemble partiellement ordonné (ou simplement ensemble ordonné) est un couple (E,≤) où E est un ensemble et "≤" est une relation d'ordre sur E. Dans un ensemble ordonné (E,≤), deux éléments x et y de E sont dits comparables lorsque x≤y ou y≤x, autrement ils sont dits incomparables. Pour deux éléments comparables et différents, x ≤y et x≠y, on note x<y. Un sous ensemble de (E,≤) dans lequel tous les éléments sont comparables est appelé chaîne. Un sous ensemble de (E,≤) dans lequel tous les éléments sont incomparables est appelé anti-chaîne.

Successeur, prédécesseur, couverture

Soient (E,≤) un ensemble ordonné et x,y ∈ E. y est dit successeur direct de x lorsque x ≤ y et il n'existe aucun élément z ∈ E telque x ≤z ≤y .tel que z≠x et z≠y. Dans ces cas, x est dit prédécesseur direct de y et on note x ≺y. Lorsque x est un prédécesseur de y on dit que y couvre x (et que x est couvert par y). Tout ensemble ordonné (E,≤) peut être représenté graphiquement par un diagramme appelé "diagramme de Hasse" (ou diagramme de couverture). A partir d'un tel diagramme on peut lire la relation d'ordre comme suit : x ≺ y si et seulement il existe un chemin ascendant qui relie le nœud x au nœud y.

Principe de dualité des ensembles ordonnés

Soit (E,≤) un ensemble ordonné, la relation inverse "≥" de "≤" est aussi une relation d'ordre sur E. "≥" est appelée duale de "≤" et (E,≥) est appelé le dual de l'ensemble ordonné (E,≤). Le diagramme de Hasse de (E,≥) peut être obtenu à partir de celui de (E,≤) par une simple symétrie horizontale. De plus, il est possible de dériver les propriétés duales de (E,≥) à partir des propriétés de (E,≤).

Treillis

Majorant, minorant, supremum, infimum

Soient (E,≤) un ensemble ordonné et S un sous ensemble de E. Un élément a∈E est dit majorant de S lorsque a≥s ∀s ∈ S. De façon duale, a ∈ E est dit minorant de S lorsque a≤s ∀s ∈ S. Le plus petit majorant (respectivement plus grand minorant) de S, s'il existe, est appelé supremum ou borne supérieure (respectivement infimum ou borne inférieure) de S et noté $\bigvee$ S (respectivement $\bigwedge$ S). Dans le cas où S=x,y, $\bigvee$ S et $\bigwedge$ S sont aussi notés par x∨y et x∧y respectivement. Dans tout ensemble ordonné, lorsque le supremum (respectivement l'infimum) existe, il est unique.

treillis, treillis complet

Un treillis est un ensemble partiellement ordonné (E,≤) tel que x∨y et x∧y existent pour tout couple d'éléments x,y ∈ E. Un treillis est dit complet si $\bigvee$ S et $\bigwedge$ S existent pour tout sous ensemble S de E. En particulier, un treillis complet admet un élément maximal (top) noté par ⊤ et un élément minimal (bottom) noté par ⊥.

Semi-treillis

Un ensemble ordonné (E,≤) est un sup-semi-treillis (respectivement inf-semi-treillis) si tout couple d'éléments x,y∈E admet un supremum x∨y (respectivement un infimum x∧y).

Fermeture

Fermeture : est un opérateur de fermeture sur un ensemble ordonné (E,≤) ssi pour tout couple (x,y) d'éléments de E, les propriétés suivantes sont vérifiées :

- x $\leq \lambda(x)(extensivité)$
$- Six \leq yalors\lambda(x) \leq \lambda(y)(monotonie)$
$- \lambda(x) = \lambda(\lambda(x))(idempotence)$
$UnélémentdeEtelquex = \lambda(x)estappeléunfermédeErelativementà\lambda.$

Connexion de Galois

Soit une relation binaire (notée R) complètement définie entre un ensemble d'objet O et un ensemble de propriétés P. Soit l'application ψ *de l'ensemble des parties de O dans l'ensemble des parties de* $P(\psi : 2^O \longrightarrow 2^P)$ *et soit l'application* ϕ *de l'ensemble des parties de P dans l'ensemble des parties de* $O(\phi : 2^P \longrightarrow 2^O)$.
Nous donnons ci après la définition d'une connexion de Galois : La paire d'opérateur (ψ, ϕ) *est une connexion de Galois si la propriété suivante est vérifiée* $\forall X \subseteq O, A \subseteq P$:
$X \subseteq \phi \circ \psi(X) \quad et \quad A \subseteq \psi \circ \phi(A)$.
Un cas particulier de connexion de Galois est obtenu en définissant les fonctions $\psi et\ \phi$ *de la manière suivante.*
La fonction ψ *associe* à *un ensemble d'objets* $X \subseteq O$ *l'ensemble* $\psi(X)$ *des propriétés* $a \in P$ *communes* à *tous les objets* $x \in X$:
$\psi(X) = a \in P \forall x \in X : xRa$
La fonction ϕ *associe* à *un ensemble de propriétés* $A \subseteq P$
l'ensemble $\phi(A)$ *des objets* $x \in O$ *communs* à *toutes les propriétés* $a \in A$:
$\phi(A) = x \in O \forall a \in A : xRa$
Il est aisé de remarquer que le couple d'application (ψ, ϕ) *est une connexion de Galois entre l'ensemble des parties de O et l'ensemble des parties de P Les propriétés suivantes sont vérifiées quelque soient les ensembles* : $X, Y, Z \subseteq O \quad et \quad A, B, C \subseteq P$
$1.X \subseteq Y \Longrightarrow \psi(X) \supseteq \psi(Y) \qquad 1'.A \subseteq B \Longrightarrow \phi(A) \supseteq \phi(B)$
$2.X \subseteq \phi(A) \Longleftrightarrow A \subseteq \psi(X)$.

2.2.2 Fondements de l'analyse de concepts formels

L'ACF fournit un cadre théorique pour l'apprentissage de la hiérarchie de concepts formels. Cet apprentissage s'effectue à partir d'un contexte formel.

Rappelons qu'un **contexte formel**[10] est un triplet K=(O,P,R) où O est un ensemble d'objets, P est un ensemble de propriétés et R est une relation binaire entre O et P appelée relation d'incidence de K et vérifiant R⊆O×P.

Un couple (x,a)∈R (noté aussi x R a) signifie que l'objet x∈O possède la propriété a∈P.

Un contexte formel peut être représenté sous la forme d'un tableau où les lignes correspondent aux objets et les colonnes correspondent aux propriétés. Un exemple de contexte formel reliant un ensemble d'objets ($x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$)à *un ensemble de propriétés* ($a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$) *est illustré* à *travers la TABLE* 2.2. *Les cases de la table sont remplies comme suit* : *si le ième objet x est en relation R avec la jème propriété a alors la case d'intersection de la ligne i et la colonne j contient* ı×ȷ *sinon la case est vide. Par exemple, dans le contexte formel représenté par la TABLE* 2.2, *l'objet* x_2 *possède les propriétés* a_1, a_4, a_6, a_7, *et l'objet* x_3 *possède les propriétés* a_2, a_5, a_7, a_9.*Etc...*

R	a1	a2	a3	a4	a5	a6	a7	a8	a9
x1	×			×	×		×		×
x2	×			×		×	×		
x3		×			×		×		×
x4		×			×		×	×	
x5		×		×			×		×
x6			×	×			×		×
x7			×	×			×		×
x8			×	×		×	×		×

TABLE 2.2 – Exemple d'un contexte formel

Operateurs de dérivation de Galois

Etant donné un objet x et une propriété a, soit R(x) =a∈P | xRa l'ensemble des propriétés satisfaites par l'objet x (xRa signifie que x possède la propriété a) et soit R(a) =(x∈O | xRa) l'ensemble des objets possédant la propriété a. On définit en ACF des correspondances entre les ensembles $2^O et 2^P$.*Ces correspondances sont appelés opérateurs de dérivation de Galois.*

Soit K un contexte formel. Pour tout X⊆O et A⊆P, on définit l'operateur ensembliste de dérivation de Galois, noté $(.)^\Delta$, *commesuit* : X^Δ *est l'ensemble des propriétés communes* à *tous les objets de* X :

$X^\Delta = a \in P \forall x \in O(x \in X \Longrightarrow xRa) = a \in P\ X \subseteq R(a) A^\Delta$ *est l'ensemble des objets possédant toutes les propriétés de* A :

$A^{\Delta} = x \in O \ \forall a \in P(a \in A \Longrightarrow xRa) = x \in O \ A \subseteq R(x)$

Les applications $(.)^{\Delta} : 2^O \longrightarrow 2^P$ *et* $(.)^{\Delta} : 2^P \longrightarrow 2^O$ *sont appelées opérateurs de dérivation entre l'ensemble des objets et l'ensemble des propriétés dans un contexte formel.*
La composition de ces opérateurs produit deux opérateurs $(.)^{\Delta}\Delta : 2^O \longrightarrow 2^O$ *et* $(.)^{\Delta}\Delta : 2^P \longrightarrow 2^P$. *Le premier opérateur permet d'associer* à *un ensemble d'objets X l'ensemble maximal d'objets dans O ayant les propriétés communes aux objets de X. Cet ensemble est noté par* $X^{\Delta}\Delta$. *De façon duale, le second opérateur permet d'associer* à *un ensemble de propriétés A l'ensemble maximal de propriétés dans P communes aux objets ayant les propriétés dans A. Cet ensemble est noté par* $A^{\Delta}\Delta$.

Exemple1

Dans la TABLE 2.2, soient les deux ensembles ordonnés $(2^O, \subseteq) et (2^P, \subseteq)$: *P est un ensemble de propriétés dont un terme A est un sous-ensemble d'un ensemble de propriétés,* $P = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9)$.
Ici $A_1 \subseteq A_2$ *signifie que le terme*
A_1 *est moins spécifique que le terme* A_2 *(par exemple,* $(a_3, a_4) \subseteq (a_3, a_4, a_6)$),
O est un ensemble d'objets $O = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$. *et X un sous ensemble d'un ensemble d'objets*
Les deux opérateurs de dérivation sont définis comme suit :
$(.)^{\Delta} : 2^O \longrightarrow 2^P, (X)^{\Delta} = (a \in P \quad \forall x \in O(x \in X \Longrightarrow xRa)), et$
$(.)^{\Delta} : 2^P \longrightarrow 2^O, (A)^{\Delta} = (x \in O \quad \forall a \in P(a \in A \Longrightarrow xRa))$.
Tel que :
$(X)^{\Delta}$ *représente l'ensemble des propriétés communes* à *tous les objets de X, et*
$(A)^{\Delta}$ *représente l'ensemble des objets qui ont toutes les propriétés de A.*
La paire duale d'opérateurs $((.)^{\Delta}, (.)^{\Delta})$ *constitue ainsi une connexion de Galois sur* 2^O *et* 2^P *qui permet d'induire des concepts formels.*

Concept formel[10]

Concept formel :
Soit K=(O,P,R) un contexte formel. Un concept formel est un couple (X,A) X⊆O, A⊆P, tel que $X^{\Delta} = A$ *et* $A^{\Delta} = X$.
X est l'extension (extent) du concept formel (X, A) *et A son intension (intent).*

L'ensemble des concepts formels associés au contexte formel K=(O,P,R) est noté par B(O,P,R). Dans un contexte formel, un concept formel correspond à un rectangle maximal de la table formée par la relation binaire du contexte : tout objet de l'extension a toutes les propriétés de l'intension.

La figure 2.2 illustre un exemple de concept formel (représenté par le rectangle maximal). Le rectangle en pointillé n'est pas un concept formel car il n'est pas maximal.

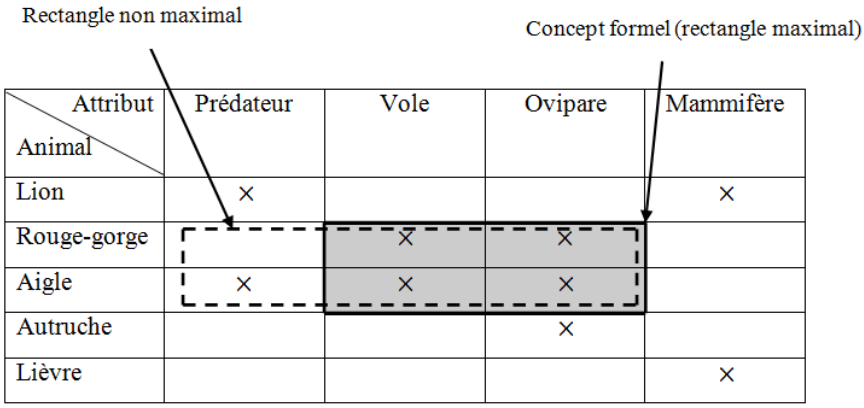


FIGURE 2.2 – Exemple de concept formel

Il est important de noter que cette notion de rectangle maximal est indépendante de l'ordre des lignes et des colonnes. Ces ensembles maximaux d'objets et de propriétés sont à la base de la définition d'un concept formel. Un sous-ensemble A de P est l'intension d'un concept formel dans B(O,P,R) si et seulement si $A^{\Delta\Delta} = A$(*A est l'ensemble maximal de propriétés dans P communes aux objets ayant les propriétés dans A*) *et de façon duale, un sous ensemble X de O est l'extension d'un concept formel dans* $B(O, P, R)$ *si et seulement si* $X^{\Delta\Delta} = X$(*X est l'ensemble maximal d'objets dans O ayant les propriétés communes aux objets de X*).

Les concepts formels de B(O,P,R) sont ordonnés par une relation d'ordre hiérarchique entre concepts formels (appelée aussi relation de subsomption) notée par "$\leq$" et définie comme suit :

Relation de "subsomption" :

Soient (X_1, A_1) *et* (X_2, A_2) *deux concepts formels de* $B(O, P, R)$. $(X_1, A_1) \leq (X_2, A_2)$ *si et seulement si* $X_1 \subseteq X_2$ (*ou de façon duale* $A_2 \subseteq A_1$). (X_2, A_2) *est dit super−concept de* (X_1, A_1) *et* (X_1, A_1) *est dit sous−concept de* (X_2, A_2). *La relation* $\imath \leq \jmath$ *est dite relation de subsom−ption.*

La relation "$\leq$" s'appuie sur deux inclusions duales, entre ensembles d'objets et entre ensembles de propriétés et peut ainsi être interprétée comme une relation de généralisation/spécialisation entre les concepts formels. Un concept formel est plus général qu'un autre concept s'il contient plus d'objets dans son extension. En contre partie, les propriétés partagées par ces objets sont réduites. De façon duale, un concept formel est plus spécifique qu'un autre s'il contient moins d'objets dans son extension. Ces objets ont plus de propriétés en commun.

Treillis de concepts formels

Théorème fondamental[10] :

La relation "$\leq$" permet d'organiser les concepts formels en un treillis complet (B (O,P,R),$\leq$) appelé treillis de concepts formels ou encore treillis de Galois et noté par B(O,P,R) ou B(K). Le supremum et l'infimum dans B(K) sont donnés par :

$$\bigwedge_{j\in J}(X_j, A_j) = (\bigcup_{j\in J} X_j((\bigcup_{j\in J} A_j)^{\Delta})^{\Delta})$$

$$\bigvee_{j\in J}(X_j, A_j) = (((\bigcup_{j\in J} X_j)^{\Delta})^{\Delta}, \bigcap_{j\in J} A_j)$$

Un treillis de concepts formels est une représentation équivalente à un contexte formel qui met en avant les groupements possibles entre objets et propriétés ainsi que les relations d'inclusion entre ces groupements :

L'exemple 2 illustre la relation d'équivalence entre un contexte formel et sa Représentation sous forme d'un treillis de concepts formels

Exemple 2

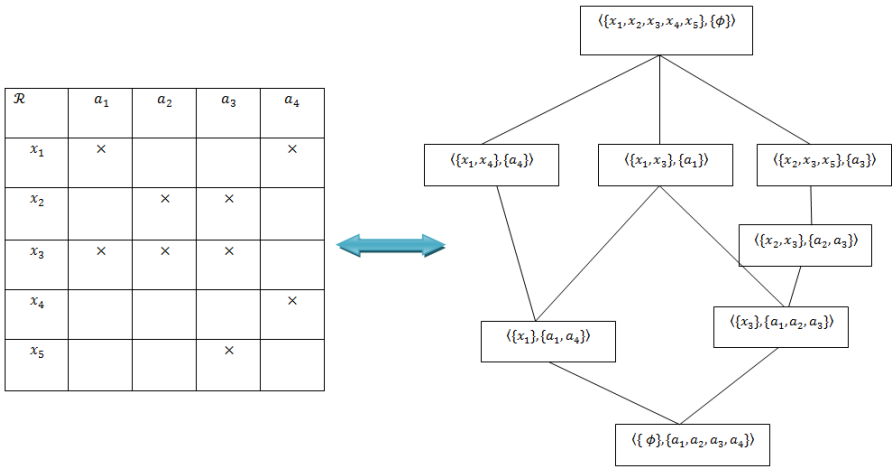


FIGURE 2.3 – contexte formel(k) et treillis de concepts équivalent au contexte formel(k)

De plus, la représentation graphique du treillis de concepts formels, sous la forme d'un diagramme de Hasse, facilite la compréhension et l'interprétation de la relation entre les objets et les propriétés d'une part et entre objets ou propriétés d'autre part. L'avantage de cette représentation est qu'à partir d'un treillis de concepts formels il est toujours possible de retrouver le contexte formel correspondant et inversement.

Algorithmes de construction de treillis de concepts formels

La construction du treillis de concepts formels d'une relation binaire donnée peut être décomposée en trois parties :
1. L'énumération des rectangles maximaux (les concepts formels),
2. La recherche de la relation d'ordre partiel entre ces concepts formels,
3. La construction du diagramme de HASSE correspondant au treillis.

Plusieurs approches ont été proposés pour la construction de treillis de concepts formels : Kuznetsov[11], Ganter[10] et ont fait l'objectif de plusieurs comparaisons. Dans leur article, Kuznetsov et Obiedkov[11] analysent plusieurs algorithmes de construction de treillis de concepts formels. Ils présentent une étude de leurs complexités théoriques et une comparaison expérimentale sur des jeux de données artificiels. Les auteurs font des recommandations en fonction de la nature du contexte. De plus, il existe plusieurs outils qui permettent d'éditer des contextes formels, et de construire le treillis de

concepts formels associé : ConExp et Galicia sont deux outils libres couramment utilisés.

Dans un treillis de concepts formels, un nœud (X,A) est un concept formel, A est l'intension et X est l'extension du concept formel. Les treillis de concepts formels sont intéressant d'un point de vue pratique, dans la mesure ou ils expriment d'une manière rigoureuse les deux facettes d'un concept formels[10]. Le treillis de Galois correspondant au contexte formel de la TABLE 2.2 est représenté par la Figure 2.4.

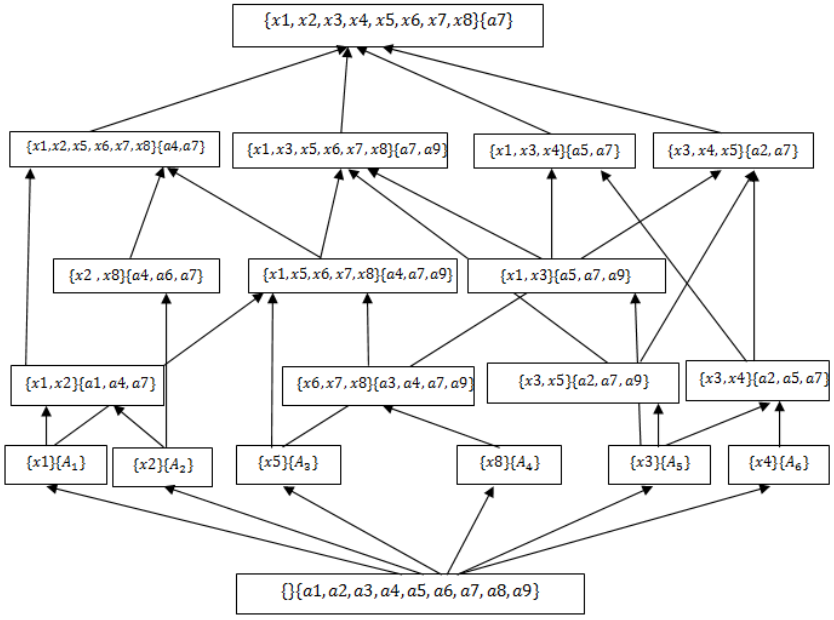


FIGURE 2.4 – Treillis de Galois (G') (correspondant au contexte dans la figure 2.3

Avec :

$A_1 = \{a_1, a_4, a_5, a_7, a_9\}$,
$A_2 = \{a_1, a_4, a_6, a_7\}$,
$A_3 = \{a_2, a_4, a_7, a_9\}$,
$A_4 = \{a_3, a_4, a_6, a_7, a_9\}$,
$A_5 = \{a_2, a_5, a_7, a_9\}$,
$A_6 = \{a_2, a_5, a_7, a_8\}$.

Implications dans un contexte formel

Soient un contexte formel K = (G, M, I) et B1, B2 $\subseteq$ M deux ensembles d'attributs. On dit que B1 implique B2 si et seulement tout objet de G qui

a les attributs de B1 a aussi les attributs de B2 :
B1 $\rightarrow$ B2 ssi B1' $\subseteq$ B2'
Dans le contexte formel des planètes du système solaire donné dans la figure 2.1 , on a l'exemple d'implication suivante :
"Satellite : non" $\rightarrow$ "Taille : petite", "Distance au soleil : proche"
qui se lit : toute planète n'ayant pas de satellite est de petite taille et proche du soleil. Considérons B1 = $\{"Satellite : non"\}$ *et*
$B2 = \{"Taille : petite", "Distance \quad au \quad soleil : proche"\}$
nousavons
$B1' = \{Mercure, Venus\}\, et B2' = \{Mercure, Venus, Terre, Mars, Pluton\}$
qui vérifient ce qu'on a définit au dessus.
Une implication de la forme $B1 \rightarrow B2$ *peut* être *ramenée* à *un ensemble d'implication de la forme* $B1 \rightarrow b$ *pour tout* $b \in B2$. *L'implication donné en exemple plus haut peut* être *ramenée aux deux implications suivantes* :
$"Satellite : non" \rightarrow "Taille : petite"$
Et
$"Satellite : non" \rightarrow "Distance \quad au \quad soleil : proche"$.

L'ensemble d'implication dans un contexte formel K = (G, M, I) peut être déduit directement à partir du treillis de concepts B(K) et on a : P $\rightarrow$ Q est une implication dans K si le concept (A, B) le plus général vérifiant Q $\subseteq$ B vérifie aussi P $\subseteq$ B. Étant donné cette constatation, le diagramme de Hasse d'un treillis de concepts avec étiquetage réduit permet une lecture directe de l'ensemble minimal non redondant de toutes les implications du contexte qui ont un support non nuls (les attributs dans P $\cup$ Q sont possédés par au moins un objet dans G). Cet ensemble est dit minimal non redondant car aucune implication ne peut être déduite en combinant deux ou plusieurs autres règles et à partir de cet ensemble on peut déduire toutes les implications possibles dans le contexte. La figure 2.5 montre l'ensemble d'implications minimales à supports positifs déduites à partir des concepts du treillis correspondant au contexte des planètes du système solaire. Dans le cas où on considère aussi les implications à support nul, l'ensemble d'implications minimales non redondantes du contexte est appelé base d'implications et noté par I(K).

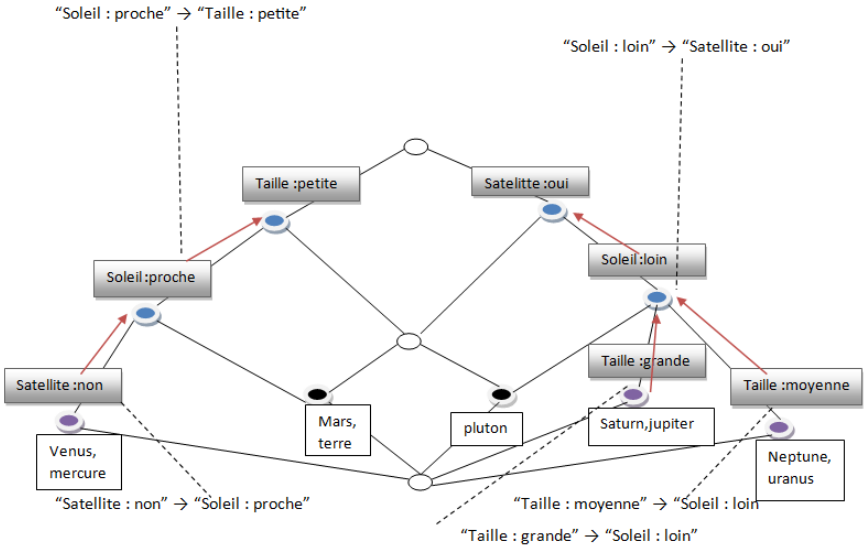


FIGURE 2.5 – Les implications minimales non redondantes à support non nul déduites directement à partir du treillis de concepts.

La base d'implications du contexte des planètes du système solaire est constitué des implications suivantes :
"Satellite : non" → "Soleil : proche"
"Soleil : proche" → "Taille : petite"
"Taille : grande" → "Soleil : loin"
"Taille : moyenne" → "Soleil : loin"
"Soleil : loin" → "Satellite : oui"
"Taille : petite", "Taille : moyenne" → "Soleil : loin", "Satellite : non"
"Taille : petite", "Taille : grande" → "Taille : moyenne", "Satellite : non"
"Taille : moyenne", "Taille : grande" → "Satellite : non"
"Soleil : proche", "Soleil : loin" → "Taille : moyenne", "Taille : grande", "Satellite : non"
"Satellite : oui", "Satellite : non" → "Taille : moyenne", "Taille : grande"

2.2.3 Exemple de domaines d'utilisation de l'ACF

La recherche d'information[9]

La recherche d'information (RI) a été l'une des premières applications qui utilise le treillis de concepts pour la découverte de ressources. Les premiers

travaux ont étudié la possibilité d'utiliser les treillis de concepts comme support pour la recherche documentaire. Des collections de documents sont alors représentées sous la forme de contextes formels. Les objets du contexte sont des documents et les attributs sont les termes d'indexation de ces documents. Chaque concept du treillis correspondant est vu comme un couple formé par une requête, dont les mots clés sont les termes contenus dans l'intension du concept, et l'ensemble de documents pertinents pour cette requête sont les documents contenus dans l'extension du concept. Le critère de pertinence dans ce cas étant celui considéré dans le cas de la recherche booléenne (conjonctive) à savoir la vérification de tous les critères spécifiés dans la requête. Ceci justifie l'interprétation donnée aux concepts du treillis puisque les objets dans l'extension d'un concept partagent tous les attributs dans son intension. Le calcul de la réponse à une requête donnée revient à identifier, dans le treillis, le concept dont l'intension est identique à la requête. Les liens de spécialisation/généralisation entre les concepts permettent d'effectuer une recherche progressive dans le treillis. Cette façon de procéder suppose que la requête existe déjà dans le treillis. Pour assurer cette condition, des algorithmes de construction incrémentale de treillis de concepts sont utilisés pour l'insertion des requêtes dans un treillis déjà construit. De cette manière, un premier mode de recherche par treillis a été défini : la recherche par interrogation. La structure hiérarchique des treillis de concepts permet la définition d'un deuxième mode de recherche : la recherche par navigation. Ces deux modes sont détaillés dans les sections suivantes.

Interogation :

Ce mode de recherche est facilité par la mise en place d'algorithmes performants pour la construction incrémentale des treillis de concepts. La définition de requête consiste à spécifier directement les termes d'indexation qui décrivent le(s) document(s) à trouver. La requête est ensuite insérée dans le treillis. La recherche des documents pertinents revient à localiser le concept le plus général incorporant les termes spécifiés dans la requête.

Navigation

Ce mode de recherche exploite la structure hiérarchique des treillis. Il consiste à explorer librement les concepts en s'appuyant sur la visualisation des treillis par des diagrammes de Hasse. Cette forme d'interaction tire profit d'une caractéristique importante de la cognition humaine : "il est plus facile de reconnaître quelque chose d'intéressant que de le décrire". Le diagramme de Hasse est utilisé comme structure de base pour la recherche. Il

offre une interface de navigation permettant de suivre les liens de spécialisation/généralisation entre concepts pour spécialiser ou élargir graduellement l'espace de recherche. Dans le cas général, le scenario de recherche peut être présenté comme suit : partant du concept le plus général du treillis qui représente la classe de tous les documents avec un ensemble de termes communs souvent vide, on effectue une spécialisation graduelle en suivant les liens descendants dans le treillis. Chaque pas dans le treillis est equivalent à l'ajout d'un nombre minimal de nouveaux termes qui spécifient la description des documents à trouver. Ceci entraîne la restriction de l'espace de recherche à un sous ensemble de documents. Cette opération est répétée jusqu'à l'identification du sous ensemble minimal de documents recherchés. Dans d'autres cas, partant d'une description très précise de documents, on peut relâcher progressivement cette description en suivant les lien ascendants du treillis. Chaque étape correspond à la suppression d'un ensemble minimal de termes ce qui entraîne l'augmentation du nombre de documents qui satisfont la description allégée. De manière générale, on peut effectuer une navigation libre dans le treillis en suivant les liens descendants et/ou ascendants en fonction du besoin en précision lors du passage d'un concept à l'autre dans le treillis.

Recherche dans des domaines spécifiques

En plus des systèmes de recherche d'information sur le web cités dans la section précédente, d'autres approches de découvertes de ressources par treillis de concepts ont aussi été définies pour la découverte de ressources dans des domaines particuliers ou encore pour la découverte de types particuliers de ressources. Parmi ces approches nous pouvons mentionner les systèmes de gestion de messagerie électronique HireMail et MailSleuth, les systèmes de recherche d'images Image-Sleuth, Camelis et de séquences vidéo et le système de recherche de bugs dans des fichiers de code source JAVA. Le principe général de ces système est le même que ceux de la recherche d'information sur le web. Cependant, la spécificité de chaque système provient de sa définition de la pertinence qui peut dépendre de la particularité du domaine étudié.

2.3 Analyse de concepts formels triadique

L'analyse de concepts formels triadique a été introduite par Lehmann et Wille[12] comme une extension à l'analyse de concepts formels. Le but d'une telle analyse est d'identifier des concepts et des implications triadiques(conditionnelles).
L'ACF Triadique traite les relations ternaires et mit en lien trois dimensions : l'ensemble des Objets, l'ensemble des Propriétés et les conditions.
contrairement a l'ACF qui mit en lien deux dimensions : les objets et les propriétés.

2.3.1 définitions

Contexte triadique[13]

On part d'un contexte triadique K :=(O ; P ; C ; R) avec O, P et C représentant respectivement un ensemble d'objets, un ensemble d'attributs et un ensemble de conditions, et R $\subseteq$ O$\times$P$\times$C une relation ternaire entre les trois ensembles. Le triplet (a1 ; a2 ; a3) dans R signifie que l'objet a1 possède l'attribut a2 sous la condition a3. Le but d'une telle analyse est d'identifier des concepts et des implications triadiques.

K	P	N	R	K	S
1	abd	abd	ac	ab	a
2	ad	bcd	abd	ad	d
3	abd	d	ab	ab	a
4	abd	bd	ab	ab	d
5	ad	ad	abd	abc	a

TABLE 2.3 – Un contexte triadique K := (O ; P ; C ; R)

L'exemple du table 2.3 représente un cube de données à trois dimensions (Client, Fournisseur, et Produit). Il y a cinq clients du groupe O notés de 1 à 5 qui achètent auprès de cinq fournisseurs (Pierre, Nelson, Richard, Kevin et Simon) du groupe P les produits de l'ensemble C. Ces produits sont des accessoires, livres (books), ordinateurs (computers) et caméras (digital cameras).

Concept triadique[13]

Un concept triadique d'un contexte K := (O ; P ; C ; R) est un triplet (A1 ; A2 ; A3) avec A1 ⊆ O, A2 ⊆ P, A3 ⊆ Cet A1×A2×A3 ⊆ R. Il représente un cuboïde plein de 1. Les sous-ensembles A1, A2 et A3 sont appelés respectivement l'extension, l'intention et le mode (modus) du concept.
Du TABLE 2.3, on peut extraire plusieurs concepts triadiques comme (12345 ; PRK ; a) et (14 ; PN ; bd). Par contre, le triplet (135 ; PN ; d) n'est pas un concept car il est non maximal puisque son extension peut être augmentée pour aboutir au concept (12345 ; PN ; d) tout en respectant la relation ternaire.

Contexte dyadique

Voici un exemple d'un contexte dyadique K(1) (TABLE 2.4) obtenu a partir du contexte triadique représenté dans la TABLE 2.3 :

	P				N				R				K				S			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
1	1	1		1	1	1		1	1		1		1	1			1			
2	1			1		1	1	1	1	1		1	1			1				1
3	1	1		1				1	1	1			1	1			1			
4	1	1		1		1		1	1	1			1	1						1
5	1			1	1			1	1	1		1	1	1	1		1			

TABLE 2.4 – Contexte dyadique K(1) := (O ; P × C ; R(1)) extrait de K

2.3.2 Extraction des règles d'associations triadiques et implications

Pour les implications triadiques, c'est d'abord Biedermann[14] qui a proposé la définition suivante :

Définition 1

Une implication triadique de la forme (A ⟶ D)C est vraie si chaque fois que A se produit pour l'ensemble des conditions dans C, alors D se produit également dans les mêmes conditions.

L'implication (N $\longrightarrow$ P)abd est vraie puisque chaque fois que le fournisseur N vend l'ensemble des produits (a ; b ; d), le fournisseur P fait autant.
Plus tard, Ganter et Obiedkov [15] ont proposé trois nouveaux types d'implications :
les implications de type attribut×condition, les implications d'attributs conditionnels (conditional attribute implications), et les implications de type conditions attributionnelles (attributional condition implications).

Définition 2

Une implication attribut×condition (AxCI) a la forme A $\longrightarrow$ D, avec A ; D des sous-ensembles de P×C. De telles implications sont extraites du contexte binaire K(1) := (O ; P×C ; R (1)) où (ai ; (aj ; ak)) $\in$ Y (1) :, (ai ; aj ; ak) $\in$ Y .
De telles implications sout plutôt dyadiques puisqu'elles sont produites à partir du contexte K(1) issu de l'aplatissement du contexte triadique (voir la TABLE 2.4).
Par exemple, l'implication (N − b $\longrightarrow$ K − a ; N − d ; P − a ; P − d ; R − a) [60% ; 100%] signifie que tous les clients ayant acheté le produit b auprès du fournisseur N ont également acheté le produit a auprès des fournisseurs K ; P et R et le produit d auprès des fournisseurs N et P . Le support de cette implication vaut 60%.

Définition 3

Une implication d'attributs conditionnels (CAI) est de la forme A$\xrightarrow{c}$ $D, avec\ A\ et\ D\ des\ sous-ensembles\ de\ P,\ et\ c\ un\ sous-ensemble\ de\ C.\ L'implication\ signifie\ que\ A\ implique\ D\ pour\ la\ totalité\ ou\ tout\ sous-ensemble\ des\ conditions\ de\ c.\ Une\ telle\ implication\ est\ alors\ liée$ à $la\ définition\ de\ l'implication\ triadique\ de\ Biedermann\ comme\ suit$ [12] : $A \xrightarrow{c} D \iff (A \xrightarrow{c} D)_{C1}\ pour\ tout\ C1 \subseteq C,\ y\ compris\ pour\ tout\ élément\ atomique\ de\ C.$
$Par\ exemple,\ N \xrightarrow{ad} P\ veut\ dire\ que\ quand\ Nelson\ fournit\ des\ accessoires\ et\ des\ caméras\ (digital\ cameras),\ alors\ Pierre\ fait\ autant.\ Bien\ que\ l'implication$ à $la\ Biedermann\ (N \longrightarrow P)_{abd}\ soit\ vraie,\ l'implication\ CAI\ N \xrightarrow{abd} P\ ne\ l'est\ pas\ car\ (N \longrightarrow P)_{C1}\ n'est\ pas\ vérifiée\ pour\ tous\ les\ C1 \subseteq (a;b;d)\ et\ en\ particulier\ pour\ C1 \in (b;bd).$

Définition 4

Une implication de type conditions attributionnels (ACI) est de la forme $\mathrm{A} \xrightarrow{c} D$, *où* A *et* D *sont des sous−ensembles de* C, *et* c *est un sous−ensemble de* P.
La relation $b \xrightarrow{PN} d$ *signifie que chaque fois que les livres*(*books*) *sont fournis par Pierre et Nelson*(*ou l'un d'eux*), *alors les caméras* (*digital cameras*) *sont aussi fournies par ces deux fournisseurs. En étendant la notion d'implications d'attributs conditionnels aux règles d'association, on obtient la définition suivante* :

Définition 5

Une règle d'association d'attributs conditionnels à la Biedermann (BCAAR) $(\mathrm{A} \longrightarrow \mathrm{D})_c$ $[s; c]$ *est vraie si chaque fois que* $A \subseteq P$ *est vraie sous l'ensemble de conditions* (*et non nécessairement un sous−ensemble de*) $c \subseteq C$, *alors* $D \subseteq P$ *est vraie pour* C *avec un support* s *et une confiance* c.
L'avantage des règles d'association triadiques et principalement des implications comme CAI et ACI repose sur le fait qu'elles représentent des motifs(*connaissances*) *de manière plus compacte et significative que les AxCI et plus généralement des règles d'association qui peuvent être extraites du contexte formel* (*dyadique*) $K(1) := (O; P \times C; R(1))$.

2.4 Conclusion

Dans ce chapitre, nous avons parlés d'analyse de concepts formels et l'ACF triadique.
On a mis l'accent sur le fait qu'il y a trois dimension dans l'ACF Triadique contrairement a l'ACF classique qui en a juste deux, tout ça on l'avait expliqué dans l'introduction.
Ensuite on a donné des petits rappels (définitions) sur l'ACF et l'ACF triadique et juste après on a vu les différents types des règles d'associations triadiques et les implications.
L'ACF triadique comme l'ACF classique peut être un outil qui nous permet d'extraire des relations plus détaillées entre propriétés.par exemple dans l'ACF la relation entre les propriétés se formalise comme suit : C1$\longrightarrow$ C2 (si la propriété appartient à C1 alors elle appartient aussi à C2) Dans l'ACF triadique elle se formalise comme suit : C1$\xrightarrow{a1,a2}$ $C2$ (*si la propriété appartient a* $C1$ *sous les conditions* $a1$ *et* $a2$ *alors elle appartient aussi a* $C2$ *sous les même conditions*.

Chapitre 3

Approche proposée

3.1 Introduction

La classification multi-label[1] est une extension de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, chaque individu pouvant appartenir à plusieurs classes simultanément. Ce type de classification est requis par un grand nombre d applications actuelles telles que la classification d images et l annotation de vidéos. Le principal objectif de ce projet est la proposition de nouvelles méthodes pour répondre au problème de classification multi-labels.
Plusieurs travaux ont porté sur la représentation des relations existantes entre les classes[1,2]. Une de ces approches est basée sur l'extraction et l'exploitation de règles d'association pour comprendre et analyser ce type de relations et pour améliorer les résultats des classifieurs . les règles multi-labels sont générées à partir des règles d'associations et sont utilisées pour modéliser les relations entre les labels afin d'améliorer les résultats d'une classification. les règles d'association sont utilisées pour la catégorisation hiérarchique des documents, les relations hiérarchiques entre les différentes classes sont basées sur la similarité des documents qui leur appartiennent.
Ainsi, nous présentons dans ce chapitre une nouvelle approche pour extraire une variante des règles d'associations dans le contexte de la classification multi-labels, appelée règles d'associations conditionnelles, basée sur l'analyse triadique des concepts formels qui représente une technique de tri-clustering basée sur la théorie des treillis[10]. Les principales contributions de ce chapitre sont les suivantes :
1. L'utilisation de l'analyse des concepts formels pour extraire toutes les règles d'association capturant les relations entre les classes.
2. L'introduction de règles d'associations conditionnelles dans le contexte de

la classification multi-labels. L'approche proposée représente non seulement les relations entre les labels, mais prend également en compte les caractéristiques des règles conditionnelles en tant que paramètres.

3.2 Problématique

Dans la classification multi-labels, chaque objet peut appartenir à plusieurs classes simultanément. Très souvent, dans le cas d'applications réelles, les classes ne sont pas indépendantes les unes des autres. L'exploitation des relations entre les classes peut enrichir et faciliter le processus d'apprentissage de la classification multi-labels. Une des approches existantes[2,10] dans ce contexte est basée sur l'extraction et l'exploitation des règles d'associations. Cependant, les travaux[1,10] utilisant des règles d'associations dans la classification multi-labels sont basés sur l'exploitation de règles d'associations simples de la forme
(Y1 $\longrightarrow$ Y2), où Y1 et Y2 sont des sous-ensembles de classes, sans aucune autre information.
Bien que ces règles d'association présentent un outil efficace pour découvrir d'éventuelles relations entre les classes, il présente certaines limites, notamment
- La génération d'un très grand nombre de règles d'association,
- Certaines règles sont inutiles et ne fournissent pas de nouvelles informations,
- Les règles d'associations générées ne sont pas riches sur le plan sémantique.

Il serait intéressant d'étudier les relations de classes du point de vue des attributs que les objets partagent et de les inclure ensuite dans les règles d'associations, ce qui donnera plus de sens aux règles d'associations produites et les enrichira sémantiquement.
Ainsi, nous proposons dans ce chapitre une nouvelle approche pour extraire une variante des règles d'association dans le contexte de la classification multi labels, appelée règles d'association conditionnelles, basée sur l'analyse de concepts formels. Ces règles d'association sont de la forme (Premisse $\longrightarrow$ Conclusion)condition où la prémisse et la conclusion sont des sous-ensembles d'étiquettes et conditionnent un sous-ensemble d'attributs. Ce type de règles d'association exploite non seulement les relations entre les étiquettes, mais prend également en compte les caractéristiques des instances en tant que paramètres d'une règle conditionnelle, ce qui donne plus de sens à la relation entre les labels existants.

Par exemple, dans le domaine médical, lors du traitement des pathologies, l'implication (maladie1 $\longrightarrow$ maladie 2)C1,C2, qui signifie que lorsque la maladie 1 est présente sous les symptômes C1 et C2 alors la maladie 2 est également présente sous les mêmes symptômes, est plus expressive que lors de l'utilisation de la simple implication du formulaire (maladie 1 $\longrightarrow$ maladie 2) sans inclure les attributs dans la relation des étiquettes. De plus, les règles obtenues sont basées sur l'extraction d'ensembles d'items fréquents fermés au lieu d'ensembles d'items fréquents. Cette approche permet de :

- D'améliorer les temps de calcul, puisque dans la plupart des cas, le nombre d'items fermés fréquents est bien inférieur à celui des items fréquents, surtout pour les contextes denses
- Générer uniquement des règles associatives qui ne sont pas redondantes (un ensemble minimal de règles).
- l'extraction d'items fermés fréquents a donné lieu à une sélection de sous-ensembles de règles sans perte d'information. Cette sélection est basée sur l'extraction d'un sous-ensemble de toutes les règles d'association, appelé base générique, à partir duquel le reste des règles pourrait être déduit. Ces bases génériques présentent un ensemble minimal de règles, à partir duquel toutes les règles valides peuvent être trouvées en appliquant des axiomes.

3.3 Les étapes générales de l'approche proposée

Pour illustrer les différentes propositions, nous utilisons l'exemple des données multi-labels données par la TABLE 3.1 , avec 9 instances décrites par 4 attributs (a1, a2,a3, a4) et répartis en 3 classes (y1, y2, y3).

	Attributs				Classes		
	a1	a2	a3	a4	y1	y2	y3
1	1	1	1	1	1	1	1
2	0	0	1	0	1	0	0
3	0	1	1	1	1	1	0
4	1	0	1	1	1	1	1
5	0	1	1	1	1	1	0
6	1	0	0	1	0	1	1
7	0	1	1	1	1	1	0
8	1	0	1	1	1	1	1
9	0	0	1	1	1	0	0

TABLE 3.1 – Ensembles de données multi-labels

Lecture du tableau 3.1
si on prend l'exemple d'objet 3 est décrit par les attributs a2, a3 et a4 ET il appartient a la classe y1 et y2.

Le format arff des données multi-labels

Pratiquement les données multi-labels(MLD) sont présentées sous format **arff** comme le montre l'exemple suivant :
Voici le format d'un fichier arff. Chaque fichier a la structure suivante :
@relation : Name of the data set
@attribute : Description of an attribute (one for each attribute)
@inputs : List with the names of the input attributes
@output : List with the names of the output attributes
@data : Starting tag of the data

Dans l'exemple illustratif, ci dessous, on a choisi un fichier avec 3 classes (classe1, classe2, classe3) et 4 attributs (attribut1, attribut2, attribut3, attribut4). Dans la partie inputs on introduit les attribut et en outputs en aura les classes. La première ligne de la partie @data signifie que l'objet 1 possède les attributs 2 et 4 et ne possède pas les attributs 1 et 3 ET il possède les classes 2 et 3 mais pas la classe 1.

Remarque le nombre de classes, d'attributs et d'objets est connu d'avance.

```
@relation exemple : -C -3     // C est le nombre de classes et dans cet exemple il est égale à 3

@attribute attribut1{0,1}
@attribute attribut2{0,1}
@attribute attribut3{0,1}
@attribute attribut4{0,1}
@attribute classe1{0,1}
@attribute classe2{0,1}
@attribute classe3{0,1}

@inputs attribut1, attribut2, attribut3, attribut4
@outputs classe1, classe2, classe3

@data
0, 1, 0, 1, 0, 1, 1  // les quatre premiers champs représentent les attributs et les trois derniers
1, 0, 0, 1, 1, 0, 0      représentent les classes. 1 veut dire que la classe ou l'attribut est présent
1, 1, 0, 1, 1, 0, 1      et 0 veut dire qu'il est absent
1, 0, 1, 0, 1, 1, 0
```

3.3.1 Représentation des données multilabels sous la forme d'un contexte formel triadique.

La première étape consiste à transformer les données multi-labels afin d'obtenir un contexte formel triadique K = (O, P, C, R) où O représente l'ensemble des objets, P l'ensemble de toutes les classes et C l'ensemble de tous les attributs (qui représente l'ensemble des conditions). Dans le cadre de la transformation, les données multi-labels de la TABLE 3.1 sont représentées par le contexte triadique formel K (TABLE 3.2)

K	y1	y2	y3
1	a1,a2,a3,a4	a1,a2,a3,a4	a1,a2,a3,a4
2	a3		
3	a2,a3,a4	a2,a3,a4	
4	a1,a3,a4	a1,a3,a4	a1,a3,a4
5	a2,a3,a4	a2,a3,a4	
6		a1,a4	a1,a4
7	a2,a1,a4	a1,a4	
8	a1,a3,a4	a1,a3,a4	a1,a3,a4
9	a3		

TABLE 3.2 – Le contexte formel triadique obtenu après la transformation

3.3.2 Transformation du contexte formel triadique en un contexte formel dyadique[4].

La deuxième étape consiste à Transformer le contexte formel triadique (relation tridimensionnelle) obtenu à l'étape 1(TABLE 3.2) en un contexte formel dyadique (relation bidimensionnelle), présenté dans la TABLE 3.3.

	y1				y2				y3			
	a1	a2	a3	a4	a1	a2	a3	a4	a1	a2	a3	a4
1	1	1	1	1	1	1	1	1	1	1	1	1
2			1									
3		1	1	1		1	1	1				
4	1		1	1	1		1	1	1		1	1
5		1	1	1		1	1	1				
6					1			1	1			1
7		1	1	1	1			1				
8	1		1	1	1		1	1	1		1	1
9			1									

TABLE 3.3 – Le contexte formel dyadique obtenu aprés la transformation

3.3.3 Lattice Miner et Le format des données qu'il utilise

Lattice Miner[13]

Lattice Miner 2.0[13,16] est un prototype d'exploration de données développé par le laboratoire de recherche LARIM de l'Université du Québec en Outaouais sous la direction du professeur Rokia Missaoui[13]. Il permet la génération de clusters (appelés concepts formels) et de règles d'association (y compris les implications logiques) étant donné une relation binaire entre une collection d'objets (ou d'individus) et un ensemble d'attributs (ou de propriétés). Lattice Miner se concentre sur la visualisation, l'exploration et l'approximation de la découverte de motifs (connaissances) à travers une représentation en treillis d'une structure plate ou imbriquée. Lattice Miner est une plate-forme Java du domaine public dont les principales fonctions incluent toutes les opérations et structures de bas niveau pour la représentation et la manipulation des données d'entrée, des treillis et des règles d'association. L'interface propose un éditeur de contexte et un manipulateur de réseau

de concept pour assister l'utilisateur dans un ensemble de tâches d'exploration de données interactives et ciblées. L'architecture de Lattice Miner est suffisamment ouverte et modulaire pour permettre l'intégration de nouvelles fonctionnalités et installations. La dernière version appelée Lattice Miner 2.0 d'avril 2017 inclut le calcul des implications avec négation, ainsi que le calcul d'implication triadique (tridimensionnel).

Le format des données utilisées par Lattice Miner

Le logiciel Lattice Miner n'accepte pas d'autres format des fichier à part le format .JSON voici un exemple qui illustre ce type de format :

```
{
 "name": "exemple",
 "objects": ["1", "2", "3", "4"],
 "attributes": ["classe1", "classe2", "classe3"],
 "conditions": ["attribut1", "attribut2", "attribut3", "attribut4"],
 "relations": [
  [
   [], ["attribut2", "attribut4"],  ["attribut2", "attribut4"]
  ],
  [
   ["attribut1", "attribut4"], [], []
  ],
  [
   ["attribut1","attribut2","attribut4"], [], ["attribut1","attribut2","attribut4"]
  ],
  [
   ["attribut1", "attribut3"], [], ["attribut1", "attribut3"]
  ]
 ]
}
```

Exemple : la première ligne de la première illustration(format arff) sera traduite avec le contenu de ce qui est entre deux crochets dans la première ligne de "relations", comme suit .

```
0, 1, 0, 1, 0, 1, 1
  [
   [], ["attribut2", "attribut4"], ["attribut2", "attribut4"]
  ],
```

la première classe est à 0 donc il y a rien à l'interieur des crochets la classe 2 est à 1 et donc elle apparait quand les attribut 2 et 4 apparaissent idem pour la classe 3.

3.3.4 Extraction des règles d'association conditionnelle de la forme (Premisse⟶Conclusion)Condition

La définition de Bidermann[14] peut être étendue au contexte de la classification multi-labels et peut être interprétée comme suit :

Soit MLD= (O, P, C) un ensemble de données multi-labels où O est un ensemble d'objets, P un ensemble de classes et C un ensemble d'attributs. Soit K= (O, P, C, R) le contexte formel triadique obtenu à partir des données multi-labels. Une implication conditionnelle des classes entre deux sous-ensembles de classes Y1 = (y1, y2, ... yn) et Y2 = ($y1'$, $y2'$, ... , yn') du formulaire (Y1 ⟶ Y2)C signifie que lorsque Y1 apparaît sous la condition c alors Y2 apparaît également sous la même condition, avec Y1 ∩ Y2 = ∅ et c est un sous-ensemble des attributs c = (a1, a2, ... , am).

La figure 3.4 présente l'ensemble de toutes les implications conditionnelles entre les classes obtenues à partir du contexte K (figure 3.2) en utilisant le lattice miner :

({y2} -> {y3})a1
({y3} -> {y2})a1
({y1} -> {y2, y3})a1
({y1} -> {y2})a2
({y2} -> {y1})a2
({y3} -> {y1, y2})a2
({y2} -> {y1})a3
({y3} -> {y1, y2})a3

({y1} -> {y2})a4
({y3} -> {y2})a4
({y1, y3} -> {y2})a4
({y2} -> {y3})a1
({y2} -> {y1})a2
({y2} -> {y1})a3
({y3} -> {y2})a1
({y3} -> {y1, y2})a2

({y3} -> {y1, y2})a3
({y3} -> {y2})a4
({y1} -> {y2, y3})a1
({y1} -> {y2})a2
({y1} -> {y2})a4
({y1, y3} -> {y2})a4

FIGURE 3.1 – Ensemble des implications conditionnelles entre les classes

Nous remarquons que ces implications ajoutent beaucoup de clarté dans la description des relations existantes entre les classes ; ceci en incluant les attributs des objets comme conditions. Par exemple, l'implication ((y1, y2) -> (y4))a5 signifie que chaque fois que les classes y1 et y2 apparaissent lorsque l'attribut a5 est présent, la classe y4 apparaît également lorsque le même attribut est présent (c'est-à-dire a5).
De même, l'implication ((y3) -> (y1, y4))a4 signifie que si un objet avec l'attribut a4 est dans la classe y3, alors il est également dans les classes y1 et y4.
Bien que ce type d'implication améliore la relation d'attrait des classes et lui donne plus de sens, le problème du grand nombre d'implications demeure.
Pour cela, nous présentons, dans ce qui suit, une autre base d'implications sémantiquement riche mais avec un nombre d'implications réduit.

3.3.5 Base condensée des implications des classes conditionnelles

Dans la classification multi-label, l'implication de Bidermann[14] (Y1 $\longrightarrow$ Y2)C est interprété comme : Si un objet possède toutes les classes de Y1 dans toutes les conditions de C, alors il possède également toutes les classes de Y2 dans toutes les conditions de C.
Dans ce qui suit, nous considérons un autre type d'implication conditionnelle qui est une extension de la définition de Bidermann :
Soit MLD= (O, P, C) un ensemble de données multi-labels où O est un ensemble d'objets, P un ensemble de classes et C un ensemble d'attributs. Soit K = (O, P, C, R) le contexte formel triadique obtenu à partir des données multi-labels.
Une implication conditionnelle des classes entre deux sous-ensembles de classes Y1 = y1, y2, ... yn et Y2 = y1′, y2′, ... yn′ du formulaire

$(\text{Y1} \xrightarrow{C} Y2)$ *et se lit comme suit* :

Y1 implique Y2 dans toutes les conditions de C ou de tout sous-ensemble de celui-ci et signifie que chaque fois que Y1 apparaît sous la condition C ou tout sous-ensemble dans C alors Y2 apparaît également sous la même condition, avec Y1 ∩ Y2 = ∅ et C est un sous-ensemble des attributs C = a1, a2, ..., am.

La figure 3.5 présente la base de toutes les classes d'implications conditionnelles obtenues à partir de contexte K (figure 3.2) en utilisant le treillis minier :

{y2}-(a1)-->{y3}
{y2}-(a23)-->{y1}
{y3}-(a1234)-->{y2}
{y3}-(a23)-->{y1}
{y1}-(a1)-->{y3}
{y1}-(a124)-->{y2}
{y1, y3}-(a4)-->{y2}

FIGURE 3.2 – Ensemble de la base condensée des implications conditionnelles.

Par exemple, l'implication (y1, y3)-(a4)>(y2) signifie que chaque fois que les classes y1 et y3 apparaissent sous la condition a4 (lorsque l'attribut a4 est présent), la classe y2 apparaît également sous la même condition.
De même, (y3)-(a1234)>(y2) signifie que chaque fois que la classe y3 apparaît sous les conditions a1, a2, a3 et a4 (ou tout sous-ensemble de ces attributs), la classe y2 apparaît également sous les mêmes conditions.

3.4 Conclusion

Dans ce chapitre, nous avons proposé une approche pour extraire les implications entre les classes dans le contexte de la classification multi-labels, basée sur la théorie de l'analyse formelle des concepts. Cette approche nous permet, non seulement de définir les relations entre les classes, mais aussi d'inclure les attributs des objets comme paramètres. Cela permet une meilleure compréhension et une meilleure interprétation de ces relations. En perspective, nous avons l'intention d'utiliser les implications conditionnelles dans la réduction du nombre de classes dans une phase de prétraitement.

Conclusion générale

Dans le cadre de ce mémoire, nous avons introduit la classification multi-labels avec ses différentes méthodes, puis on a présenté l'analyse de concept formel et l'ACF triadique ou on a expliqué comment extraire des implications conditionnelles a partir d'un contexte formel triadique.

Le présent mémoire parle d'une approche qui consiste a étudier la corrélations entre les classes dans le cadre de la classification multi-labels tout en utilisant l'analyse de concept formels triadique pour justement extraire des implications conditionnels de forme (Y1 $\longrightarrow$ Y2)C qui est interprété comme : Si un objet possède toutes les classes de Y1 dans toutes les conditions de C, alors il possède également toutes les classes de Y2 dans toutes les conditions de C.

Il permet, non seulement de définir les relations entre les classes, mais aussi d'inclure les attributs des objets comme paramètres. Cela permet une meilleure compréhension et une meilleure interprétation de ces relations entre les classes.

Dans un premier temps, on a transformé les connaissances multi-labels sous la forme d'un contexte triadique, ensuite nous avons transformé le contexte formel triadique obtenu en un contexte formel dyadique, pour après extraire les règles d'associations conditionnelles.

L'avantage d'une telle approche, en plus de l'obtention du treillis de Galois (des concepts triadique), est que les règles d'associations qu'on a extrait sont conditionnelle. Par exemple, (classe1 $\longrightarrow$ classe3)C, Cette implication est vraie si chaque fois que la classe1 se produit pour l'ensemble des conditions dans C, alors la classe3 se produit également dans les mêmes conditions.
Par exemple, (classe1 $\rightarrow$classe3)$_{A1,A3}$ *est vraie puisqu'*à *chaque fois que la classe*1 *se produit pour les attributs A*1 *et A*2 *alors la classe*3 *se produit aussi pour les même attributs*(*la même conditions*).

Bibliographie

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

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

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

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

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

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

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

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

[9] Yoav Freund et Robert Schapire, « A decision-theoretic generalization of on-line learning and an application to boosting », Journal of Computer and System Sciences, vol. 55, no 1, 1997, p. 119-139

[10] Ganter, B., Wille, R. : Formal Concept Analysis. Springer, Heidelberg (1999).

[11] Cheng, W., K. Dembczynski, et E. Hüllermeier. (2010). Graded multilabel classification : The ordinal case. In M. Atzmüller, D. Benz, A. Hotho, et G. Stumme (Eds.),

[12] Lehmann, F., and Wille, R. A Triadic Approach to Formal Concept Analysis. In Proceedings of the Third International Conference on Conceptual Structures : Applications, Implementation and Theory (1995), pp. 32–43.

[13] Missaoui, R., and Kwuida, L. Mining Triadic Association Rules from Ternary Relations. In 9th International Conference ICFCA (May 2011), pp. 204–218.

[14] Biedermann, K. How Triadic Diagrams Represent Conceptual Structures. In ICCS (1997), pp. 304–317.

[15] Ganter, B., and Obiedkov, S. A. Implications in Triadic Formal Contexts. In ICCS (2004), pp. 186–195.

[16] Roberge, G. Visualisation des résultats d'une fouille de données dans les treillis de concepts. Master's thesis, Université du Québec en Outaouais, 2007.