



HAL
open science

Apprentissage par Regroupement d'Attributs dans les Systèmes d'Inférence Floue

Ilef Ben Slima

► **To cite this version:**

Ilef Ben Slima. Apprentissage par Regroupement d'Attributs dans les Systèmes d'Inférence Floue. Apprentissage [cs.LG]. Université de Tunis El Manar, 2019. Français. NNT: . tel-03181242

HAL Id: tel-03181242

<https://hal.science/tel-03181242>

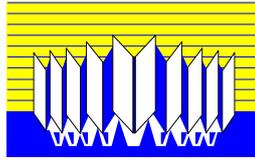
Submitted on 20 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



École Doctorale de Mathématiques,
Informatique, Science
et Technologie de la Matière



Faculté des Sciences Mathématiques,
Physiques
et Naturelles de Tunis



Université Tunis El Manar

Laboratoire d'Informatique en Programmation Algorithmique et Heuristique (LIPAH)

Thèse

Apprentissage par Regroupement d'Attributs dans les Systèmes d'Inférence Floue

Présentée pour obtenir le grade de

DOCTEUR EN INFORMATIQUE

Par

Ilef BEN SLIMA

Soutenue le 25 Avril 2019

Devant le jury composé de

Mr. Faouzi Moussa	Professeur, Faculté des Sciences de Tunis, Université de Tunis El Manar	Président
Mr. Mounir Ben Ayed	Maître de Conférences, Faculté des Sciences de Sfax, Université de Sfax	Rapporteur
Mr. Zied Elouedi	Professeur, Institut Supérieur de Gestion de Tunis, Université de Tunis	Rapporteur
Mr. Sami Faïz	Professeur, Institut Supérieur des Arts Multimédia de la Manouba, Université La Manouba	Examineur
Mme Amel Borgi	Maîtresse de Conférences, Institut Supérieur d'Informatique, Université Tunis El Manar	Directrice de thèse

Résumé

Les systèmes de classification à base de règles floues possèdent l'avantage de produire des modèles de classification intelligibles et compréhensibles par l'utilisateur. Ces méthodes permettent de construire un modèle sous forme d'un ensemble de règles floues. L'inconvénient majeur de ces systèmes est leur complexité lorsqu'ils sont appliqués à des bases de données de grandes dimensions. En effet, le nombre de règles floues générées par ces systèmes augmente exponentiellement avec l'augmentation du nombre d'attributs descriptifs. Ce problème peut toutefois être résolu par des méthodes d'ensemble. Ces méthodes consistent à combiner plusieurs classifieurs, en fournissant à chaque classifieur une projection différente des données de départ. En utilisant cette méthode, nous bénéficions de deux avantages principaux. D'une part, la complexité du système global est réduite par la décomposition de ce problème en des sous-problèmes de complexité inférieure. D'autre part, la performance du système de classification global peut être améliorée par la participation de plusieurs classifieurs.

C'est dans ce cadre que s'inscrit notre recherche, celui des systèmes de classification à base de règles floues utilisant l'approche des méthodes d'ensemble. Dans ce contexte, nous nous intéressons plus précisément aux méthodes d'ensemble qui visent à créer un regroupement des attributs descriptifs ; les attributs qui sont liés sont regroupés et traités par un même classifieur.

Dans un premier temps, nous cherchons à analyser les différentes manières de regrouper les attributs. Des méthodes et mesures d'association et de corrélation sont ainsi étudiées.

Dans un deuxième temps, nous proposons de nouvelles méthodes pour le regroupement d'attributs qui sont basées sur le concept des règles d'association et de la génération des itemsets fréquents. Ces méthodes permettent de détecter des associations intéressantes, de différents types et formes, entre les attributs. Ces méthodes sont aussi applicables sur différents types de données, notamment les variables quantitatives et qualitatives. Finalement, les méthodes de regroupement d'attributs, proposées dans ce travail, pourraient être utilisées dans d'autres contextes d'apprentissage que ceux des systèmes de classification à base de règles floues.

Mots clés : Regroupement d'attributs, Apprentissage supervisé, Systèmes de classification à base de règles floues, Méthodes d'ensemble, Génération d'itemsets fréquents.

Abstract

Fuzzy Rule-Based Classification Systems are very powerful since they provide an easily interpretable model consisting of linguistic if-then rules. The big challenge of these systems is how to deal with high dimensional databases. Indeed, when the number of attributes is high, an exponential increase of the generated rules number is expected. In this context, we focus on an interesting solution to cope with that problem which is the use of ensemble methods. In these methods, the learning problem which involves a big number of attributes is decomposed into sub-problems of lower complexity. Different classifiers are thus constructed with different projections of the features set and the decisions of the different classifiers are combined in order to form the final classification model. Using an ensemble method, we can take advantage of the following benefits. On the one hand, the complexity of the classification system can notably be reduced. On the other hand, the performance and precision of the classification task can be improved since we take in consideration the opinions of different classifiers rather than one single opinion.

Our research lies with the scope of the ensemble methods which are used in the context of fuzzy rule-based classification systems. We are interested in particular in the methods which can be used to regroup the set of attributes into sub-groups of dependent ones.

First, we analyze the different methods proposed in the literature to detect associations and correlations between the features.

Then, we propose new methods of attributes regrouping which are based on the association rules concept and the frequent itemsets mining. The proposed methods are able to detect interesting associations between the attributes ; these associations can be of different types and shapes. These methods can be applied on numerical as well as categorical attributes. In addition, the proposed methods of attributes regrouping can be applied not only on the fuzzy rule based classification systems, but also on different other learning approaches.

Key words : Attributes regrouping, Supervised learning, Fuzzy rule-base classification systems, Ensemble methods, Frequent itemsets mining.

Table des matières

Table de matières	4
Liste des figures	8
Liste des tableaux	11
Notations	13
Introduction Générale	15
1 Apprentissage à base de règles	21
Introduction	21
1.1 Apprentissage automatique	21
1.1.1 Apprentissage supervisé	23
1.1.2 Apprentissage non supervisé	26
1.2 Partie 1 : Apprentissage par Règles d'Association	27
1.2.1 Définitions et notions de base	27
1.2.2 Les algorithmes d'extraction d'itemsets fréquents	29
1.2.3 Les algorithmes d'extraction d'itemsets fermés et maximaux	35
1.3 Partie 2 : Apprentissage à base de règles floues	37
1.3.1 Phase d'apprentissage	38
1.3.2 Phase de classification	53
1.3.3 SCBRF et les bases à haute dimension	56
Conclusion	59
2 Méthodes d'ensemble et Regroupement d'attributs	61
Introduction	61
2.1 Ensembles d'apprentissage artificiel	62

2.2	Regroupement d'attributs dans les SCBRF	66
2.3	Dépendance entre les attributs	68
2.3.1	Mesures de corrélation entre des variables quantitatives	68
2.3.2	Mesures d'association entre des variables qualitatives	70
2.4	Méthode SIFCO	73
	Conclusion	75
3	Regroupement d'attributs par RA dans les SCBRF : la méthode SIFRA	77
	Introduction	77
3.1	Les Règles d'Association pour le <i>Subspace Clustering</i>	79
3.2	Méthode d'apprentissage supervisée par regroupement d'attributs : la méthode SIFRA	81
3.2.1	Phase de regroupement d'attributs	82
3.2.2	Phase d'apprentissage	84
3.2.3	Phase de classification	85
3.3	Une nouvelle méthode de regroupement d'attributs basée sur les Règles d'Association	85
3.3.1	Étape 1 : Génération des itemsets fréquents	86
3.3.2	Étape 2 : Détermination des attributs liés	87
3.3.3	Étape 3 : Sélection des groupes d'attributs finaux	89
3.3.4	Exemple illustratif	91
3.4	Étude des méthodes de discrétisation	95
3.5	Analyse des critères de sélection des groupes d'attributs finaux	97
3.6	Analyse du coût de la méthode de regroupement d'attributs	102
	Conclusion	109
4	Extensions de la méthode SIFRA	110
	Introduction	110
4.1	SIFRA avec discrétisation supervisée multi-variée	110
4.1.1	Méthodes de discrétisation	111
4.1.2	Les méthodes de discrétisation dans le contexte de SIFRA	114
4.2	Regroupement d'Attributs Supervisé : les méthodes SIFRA-SAR	116
4.2.1	Méthode 1 : SAR avec l'attribut Classe	118
4.2.2	Méthode 2 : SAR avec facteur d'homogénéité	121
	Conclusion	127

5 Etudes Expérimentales	129
Introduction	129
5.1 Implémentation	130
5.2 Description des bases d'apprentissage	133
5.3 Analyse des associations	134
5.4 Analyse du coût de la méthode SIFRA	142
5.5 Analyse des effets des paramètres sur la méthode SIFRA	145
5.6 Analyse de la performance des méthodes proposées	151
5.6.1 Comparaison entre SIFRA et SIFRA-SAR	151
5.6.2 Comparaison des méthodes proposées avec d'autres méthodes d'en- semble	154
5.6.3 Comparaison des méthodes proposées avec d'autres SCBRFs	155
Conclusion	162
Conclusion et perspectives	164
A La logique floue	186
B Présentation des plateformes WEKA, SPMF et SIFCO	190
C Architecture de SIFRA	193
D Étude des critères de sélection des groupes d'attributs finaux	195
E Étude des types d'associations	201
F Réduction du coût de la méthode SIFRA	205
G Effet des seuils $minsupp$ et β_{min} sur le coût de la méthode SIFRA	214
H Effet des paramètres d'entrée sur la méthode SIFRA	218
I Effet des méthodes de discrétisation sur SIFRA et SIFRA-SAR	224

Liste des figures

1	Illustration des domaines scientifiques apparentés à l'apprentissage artificiel	
	Feuilloy, 2009	23
2	Phase d'apprentissage	24
3	Phase de classification	24
4	Exemple d'arbre de décision	25
5	Exemple de FP-Tree	35
6	Itemsets maximaux dans un jeu de données Pasquier, 2000a	37
7	Structure d'un SCBRE	38
8	Apprentissage à base de grille floue	39
9	Exemple de partition floue	40
10	Exemple de grille floue simple	48
11	Grille floue multiple	49
12	Grille floue avec partition efficace	50
13	Grille floue rectangulaire	50
14	Représentation d'une fonction d'appartenance trapézoïdale	52
15	Construction d'une partition floue supervisée	53
16	Grille floue obtenue par la méthode de la partition floue supervisée	53
17	Corrélation et dépendance entre variables numériques Saporta, 2006	68
18	Sensibilité du coefficient de Pearson aux individus extrêmes	69
19	Différentes formes de corrélation avec le même coefficient r	69
20	Corrélation monotone non-linéaire	70
21	Étapes de la méthode SIFCO	73
22	Graphe des attributs corrélés	75
23	Relations non linéaires : Curvilinéaire, Cyclique et Liaisons croisées	
	Kenny, 1979	78

24	Exemple de base de données avec quatres <i>clusters</i> qui chevauchent entre eux [Parsons et al., 2004]	79
25	Les données projetées sur trois sous-espaces différents [Parsons et al., 2004]	80
26	Identification de <i>clusters</i> [Agrawal et al., 1999]	81
27	Étapes de la méthode SIFRA	83
28	<i>Subspace Clustering</i> : deux <i>clusters</i> détectés dans le sous-espace constitué par les attributs X et Y, aucun cluster détecté dans le sous-espace constitué par les attributs Y et Z	84
29	Étapes de la méthode de regroupement d'attributs de SIFRA	86
30	Exemples de grille d'association	88
31	Étapes de sélection des groupes d'attributs finaux.	90
32	Grille d'association du groupe d'attributs {SL, PW}	93
33	Regroupement d'attributs récursif	100
34	Sélection des groupes d'attributs finaux dans le cas de la méthode SIFRA avec itemsets maximaux	108
35	Graphe de voisins et clusters pour un exemple à deux classes (classe \circ et classe \bullet) [Muhlenbach and Rakotomalala, 2002]	114
36	Discrétisation des attributs en rapport avec la distribution des données	116
37	Sous-espaces détectés avec regroupement d'attributs non-supervisé (a) ou supervisé (b)	118
38	Diagramme décrivant les étapes de la méthode "SAR avec attribut Classe"	119
39	Étapes de sélection des groupes d'attributs finaux contenant l'attribut <i>Classe</i>	121
40	Étapes de sélection des groupes d'attributs finaux qui ne contiennent pas l'attribut <i>Classe</i>	122
41	Distribution des données étiquetées dans la grille d'association	124
42	Interface graphique de la plateforme SIFRA	131
43	Interface graphique : Choix de la méthode de regroupement d'attributs	132
44	Interface graphique : Choix de l'algorithme de génération des itemsets	133
45	Interface graphique : Choix de la méthode de discrétisation	133
46	Matrices de corrélation des bases de données	137
47	Quelques types d'associations détectées par SIFRA	141
48	Effet de la partition floue sur la méthode SIFRA	147
49	Valeurs de $minsupp$ et β_{min} donnant le meilleur taux de classification	148
50	Effet des méthodes de discrétisation sur la performance de la méthode SIFRA	149

51	Sous-ensemble classique	187
52	Sous-ensemble flou	187
53	Exemples d'intersection, d'union et de complément de sous-ensembles flous	188
54	En-tête d'un fichier .arff	191
55	Données d'un fichier .arff	191
56	Architecture générale du système SIFRA	194
57	Associations détectées par SIFCO	202
58	Associations détectées par SIFCO et SIFRA	203
59	Associations détectées par SIFRA	204
60	Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Iris, Balance-scale, Ecoli, Glass, Wine et Heart-statlog	208
61	Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Diabetes et Vehicle	209
62	Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour la base Unbalanced	211
63	Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Ionosphere, mfeat-fourrier et Sonar	211
64	Effet du seuil $minsupp$ sur le temps de réponse de SIFRA (avec un Zoom sur l'axe de temps)	216
65	Effet du seuil β_{min} sur le temps de réponse de SIFRA	217
66	Effet des méthodes de discrétisation sur la performance de la méthode SIFRA	226
67	Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans SIFRA	226
68	Effet des méthodes de discrétisation sur la performance de la méthode "SAR avec l'attribut Classe"	229
69	Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans "SAR avec l'attribut Classe"	229
70	Effet des méthodes de discrétisation sur la performance de la méthode "SAR avec facteur d'homogénéité"	230
71	Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans "SAR avec facteur d'homogénéité"	230

Liste des tableaux

1	Extrait d'une base de données transactionnelle	29
2	Base de données relationnelle	33
3	Mappage de la base de données relationnelle à une base de données booléenne	33
4	Principales T-normes	43
5	Table de contingence	71
6	Itemsets fréquents générés par Apriori	92
7	Description des bases de données	134
8	Comparaison de la performance de SIFRA avec d'autres méthodes de regroupement d'attributs	139
9	Comparaison des groupes d'attributs, obtenus par SIFRA et SIFCO	141
10	Comparaison entre les méthodes proposées : SIFRA et SAR	153
11	Comparaison des méthodes proposées avec d'autres méthodes d'ensemble	155
12	Comparaison de SIFRA et SIFRA-SAR avec la méthode d'Ishibuchi	157
13	Comparaison de SIFRA et SIFRA-SAR avec des méthodes de sélection de règles et de sélection d'attributs	158
14	Comparaison de SIFRA et SIFRA-SAR avec des méthodes basées sur le concept des RA	161
15	Principales T-normes et T-conormes	189
16	Méthodes de sélection des groupes d'attributs finaux : comparaison des taux de bonne classification	197
17	Méthodes de sélection des groupes d'attributs finaux : comparaison des groupes d'attributs obtenus	200
18	Groupes d'attributs obtenus par les différentes versions de SIFRA	213

19	Effet de la partition floue sur la méthode SIFRA	220
20	Effet des seuils <i>minsupp</i> et β_{min} sur la méthode SIFRA	222
21	Taux de classification obtenus par les différentes méthodes de discrétisation	225
22	Groupes d'attributs obtenus par les différentes méthodes de discrétisation .	225

Notations

Abréviations et acronymes

Abréviation	Signification
AG	Algorithme Génétique
ANFIS	Adaptive Neuro-Fuzzy Inference System
CBD	Clustering-Based Discretization
CMAR	Classification based on Multiple Association Rules
EFI	Equal Frequency Interval
EWI	Equal Width Interval
FA	Fonction d'Appartenance
MDLP	Minimum Description Length Principle
RA	Règles d'Association
SAR	Supervised Attributes Regrouping
SCBRF	Systèmes de Classification à Base de Règles Floues
SIF	Système d'Inférence Floue
SIFCO	Système d'Inférence Floue avec COrrélation
SIFRA	Système d'Inférence Floue par Règles d'Association
SIFRA-SAR	SIFRA with Supervised Attributes Regrouping
SUCRAGE	SUpervised Classification by Rules Automatic GEneration
SVM	Support Vector Machine

Symboles

Symbole	Signification
N	nombre d'attributs
m	nombre d'observations
C	nombre de classes
y_1, y_2, \dots, y_C	les C classes du problème d'apprentissage
X_i	l'attribut N° i

k	nombre de sous-ensembles flous
k_1, k_2, \dots, k_N	nombre de sous-ensembles flous des attributs X_1, X_2, \dots, X_N respectivement
E_p	une observation (exemple d'apprentissage); $p \in \{1, \dots, m\}$
$Y(E_p)$	la classe associée à l'exemple E_p
$A_1^k, A_2^k, \dots, A_k^k$	les sous-ensembles flous d'une partition de taille k
μ_i^k	fonction d'appartenance du sous-ensemble A_i^k ; $i \in \{1, \dots, k\}$
R_{ij}^k	règle floue du sous-espace $A_i^k \times A_j^k$
y_{ij}^k	conclusion de la règle floue R_{ij}^k
CF_{ij}^k	degré de certitude de la règle floue R_{ij}^k
S_R	l'ensemble des règles floues
T	la T-norme
$Int_{X_i}^1, Int_{X_i}^2, \dots, Int_{X_i}^l$	les l intervalles issus d'une discrétisation régulière de taille l de l'attribut X_i
$Int_{X_1}^j$	l'intervalle $N^{\circ}j$ de l'attribut X_1
$U_{X_1 X_2}^{ij}$	la région formée par les intervalles $Int_{X_1}^i$ et $Int_{X_2}^j$
$w_{X_1 X_2}^{ij}$	le coefficient de pondération de la région $U_{X_1 X_2}^{ij}$
$\beta_{X_1 X_2}$	le degré d'association du groupe d'attributs $\{X_1, X_2\}$
β_{min}	le seuil d'association
$minsupp$	le seuil de la mesure support
H	la mesure d'entropie
$Hom(U)$	l'homogénéité de la region U

Introduction Générale

L'apprentissage chez l'être humain est un phénomène à plusieurs facettes. Il inclut l'acquisition de nouvelles connaissances, le développement de nouvelles compétences, l'extraction d'informations, la découverte de nouveaux faits et de nouveaux théorèmes, etc. Face à la complexité et à l'évolution des problèmes du monde réel, plusieurs domaines ont nécessité que les machines accomplissent de telles tâches complexes. Nous citons par exemple les domaines du diagnostic médical, de la reconnaissance des formes, de la prévision météo, de la détection de pannes, ... où l'être humain n'est parfois pas capable de prendre des décisions bonnes et rapides. De ce fait, l'apprentissage automatique, appelé encore apprentissage artificiel ou apprentissage machine, désigne un domaine très vaste et très fascinant depuis les années 90.

Les systèmes d'apprentissage artificiels génèrent des connaissances qui peuvent prendre des formes différentes. Une classe particulière de ces méthodes utilise la construction de règles floues de type Si-Alors. Cette catégorie de méthodes d'apprentissage, connues sous le nom des systèmes d'apprentissage à base de règles floues, représentent un outil très utilisé en apprentissage automatique. La spécificité de ces systèmes réside d'une part en la compréhensibilité et l'intelligibilité du modèle de prédiction qu'ils génèrent, et d'autre part en leur capacité de traiter des données imprécises et imparfaites.

L'objectif des systèmes d'apprentissage à base de règles floues consiste à générer un modèle de prédiction formé d'un ensemble de règles linguistiques floues ; ce modèle sera utilisé pour prédire la classe de nouvelles données. Dans ces systèmes, la base de règles floues peut être construite à l'aide d'un expert humain. Cependant, la phase d'acquisition des connaissances est délicate et lourde vu que l'expert exprime ses connaissances de manière spontanée en utilisant des termes linguistiques qui nécessitent d'être codée en une forme comprise par la machine. De plus, un être humain peut trouver des difficultés pour exprimer ses idées, ce qui explique la lourdeur de cette tâche d'acquisition. L'acquisition des connaissances pour la construction de la base de règles représente un handicap pour les systèmes à base de règles floues. C'est la raison pour laquelle plusieurs travaux se sont consacrés à l'automatisation de la construction de la base de règles [D'Andrea and Lazzerini, 2013], [Dehzangi et al., 2007], [Ishibuchi et al., 1992], [Ishibuchi and Yamamoto, 2005], [Fakhrahmad et al., 2007].

Dans ces systèmes, la génération des règles floues dépend principalement des attributs

descriptifs ; une base de règles énorme peut être générée lorsque le nombre d'attributs traités est important. De ce fait, le défi majeur de ces systèmes consiste à garder un compromis entre une base de règles compacte et intelligible avec un nombre restreint de règles floues, d'une part, et une haute performance de classification, d'autre part. En effet, la génération d'une base de règles de taille importante a plusieurs effets sur le système d'apprentissage. D'un côté, le coût de la phase d'apprentissage et de génération de règles devient important en termes de temps d'exécution et d'espace mémoire. D'un autre côté, l'explosion du nombre de règles peut dégrader la compréhensibilité et l'intelligibilité du modèle prédictif. Dans ce cas, l'utilisation des règles linguistiques qui peuvent être interprétées par l'être humain n'a plus de sens car il n'est pas possible d'interpréter un nombre très important de règles même linguistiques ! Enfin, le traitement d'un très grand nombre d'attributs en même temps peut dégrader la précision de l'apprentissage (problème de malédiction de la dimension [Bellman, 1961]). L'objectif de ce travail est alors de permettre la génération d'un nombre réduit de règles floues même avec un nombre élevé d'attributs.

Dans ce contexte, plusieurs travaux se sont consacrés à l'optimisation du nombre de règles générées. Parmi ces travaux, la sélection des règles pertinentes présente l'une des solutions les plus utilisées dans les Systèmes de Classification à Base de Règles Floues (SCBRF) [Antonelli et al., 2014, Ishibuchi and Yamamoto, 2004, Alcalá et al., 2007]. Cependant, le problème de la complexité de la génération des règles demeure car la sélection des règles pertinentes intervient en aval de la phase de génération de règles. La sélection d'attributs paraît comme une autre solution aussi très utilisée dans les SCBRF [Chandrashekar and Sahin, 2014, Yu and Liu, 2003, Ferchichi et al., 2009, Ladha and Deepa, 2011, Mansoori and Shafiee, 2016]. Elle consiste à sélectionner un sous-ensemble d'attributs les plus significatifs et éliminer les attributs redondants.

Toutefois, nous nous intéressons dans ce travail à un autre aspect de réduction du nombre de règles qui utilise les ensembles d'apprentissage artificiel. Le concept des ensembles d'apprentissage repose sur la combinaison des décisions de plusieurs apprenants pour améliorer l'exécution d'un système global [Sagi and Rokach, 2018, Valentini and Masulli, 2002, Rokach, 2009]. L'idée de cette approche consiste à décomposer le problème complexe (qui implique un grand nombre d'attributs) en des sous-problèmes de complexité inférieure en créant des sous-groupes d'attributs et en attribuant chaque sous-groupe à un classifieur. Plusieurs classifieurs sont alors considérés, chacun avec une projection différente des données. Cette solution permet d'une part de diminuer la complexité de la génération des règles et de produire un nombre réduit de règles, et d'autre part de résoudre le problème de la haute dimension en traitant indépendamment chaque groupe d'attributs seul. Par ailleurs, en plus de la décomposition du problème en des sous-problèmes moins complexes, nous bénéficions de l'avantage des méthodes d'ensemble qui peuvent permettre d'améliorer la précision du système final en considérant des opinions individuelles des différents classifieurs.

Les méthodes d'ensemble ont été traitées et utilisées dans plusieurs travaux de

la littérature, en particulier dans le cadre des SCBRF [Amouzadi and Mirzaei, 2010], [Del Jesus et al., 2004], [Hoffmann, 2001], [Cao et al., 2011], [Soua et al., 2013]. Combinées avec des algorithmes d'apprentissage flou, les méthodes d'ensemble ont prouvé une haute performance et une amélioration de la précision des systèmes de classification. C'est par exemple le cas de l'algorithme des forêts d'arbre (Random Forest) qui consiste à combiner plusieurs arbres de décision pour produire un classifieur plus précis [Breiman, 2001], [Cutler et al., 2007].

Notre travail s'inspire principalement d'une méthode d'ensemble proposée dans [Soua et al., 2013], appelée la méthode SIFCO. Cette méthode adopte l'approche de regroupement d'attributs pour résoudre le problème d'explosion du nombre de règles. En se basant sur une recherche de corrélation linéaire, la méthode SIFCO rassemble les attributs qui sont corrélés et traite chaque groupe d'attributs indépendamment. Une base de règles locale est construite pour chaque groupe d'attributs liés et les différentes bases locales sont par la suite combinées pour former le modèle de classification final. L'handicap majeur de cette méthode est qu'elle ne détecte que les corrélations linéaires entre les attributs ; d'autres types de liaisons peuvent toutefois exister entre les attributs.

Dans le cadre de cette thèse, nous étudions différentes méthodes de recherche de corrélation et d'association entre les attributs qui peuvent être utilisées pour le regroupement d'attributs dans le cadre des SCBRF. L'objectif de ce travail est de développer une méthode de regroupement d'attributs qui est capable de traiter différents types de données (quantitatives ou qualitatives) et de détecter plusieurs types d'associations (autre que la corrélation linéaire). Nous souhaitons également que la méthode développée puisse être utilisée dans différentes approches d'apprentissage, et pas uniquement avec les systèmes de classification à base de règles floues.

Dans ce contexte, nous avons proposé une nouvelle méthode de regroupement d'attributs qui se base sur le concept des règles d'association et plus précisément sur la génération d'itemsets fréquents. Les algorithmes de génération d'itemsets fréquents déterminent des associations intéressantes entre les attributs en analysant leurs apparitions simultanées dans les enregistrements d'une base de données. Contrairement à la méthode SIFCO, qui utilise une recherche de corrélation linéaire pour détecter les attributs liés, notre méthode est capable de détecter différents types d'associations entre les attributs sans chercher une forme spécifique de liaison (telle qu'une liaison linéaire ou parabolique,...). De plus, la méthode de regroupement d'attributs que nous proposons dans cette thèse est capable de traiter des attributs aussi bien numériques que catégoriques. Cette méthode correspond à notre première contribution qui a donné lieu à un système nommé SIFRA (Système d'Inférence Floue par Règles d'Association). Notre système SIFRA consiste donc en une méthode d'apprentissage supervisée par génération de règles floues qui utilise l'approche de regroupement d'attributs comme une phase de pré-traitement de données.

La phase de pré-traitement de données possède une importance particulière dans notre système SIFRA. En effet, il ne suffit pas d'utiliser le regroupement d'attributs pour décomposer le problème d'apprentissage complexe en des sous-problèmes moins complexes, mais

il faut évidemment bien choisir les groupes d'attributs appropriés ; les attributs qui sont liés seront par la suite regroupés et traités ensemble pour générer une base de règles locale. Donc, la qualité des règles floues générées dépend principalement du choix des groupes d'attributs liés. À ce niveau, nous attirons l'attention sur le fait que la méthode de regroupement d'attributs que nous proposons dans SIFRA détecte des liaisons intéressantes entre les attributs, mais ne considère pas le pouvoir prédictif des groupes d'attributs formés. Pour cette raison, nous proposons une deuxième contribution qui consiste à prendre en compte l'information liée à l'attribut *Classe* lors de la formation des groupes d'attributs liés. Ainsi, des méthodes de regroupement d'attributs supervisées, appelées SIFRA-SAR (SIFRA with Supervised Attributes Regrouping), ont été proposées dans cette thèse pour choisir les groupes d'attributs liés tout en analysant leur pouvoir prédictif. Cette approche de regroupement d'attributs supervisé a été proposée en deux versions différentes ("SAR avec attribut Classe" et "SAR avec facteur d'homogénéité"); chaque version prend en compte l'information liée à l'attribut *Classe* de manière différente. Une étude expérimentale sur différents jeux de données a été menée afin d'évaluer nos approches et les comparer à d'autres méthodes d'ensemble et d'autres méthodes de classification.

D'autres extensions de la méthode SIFRA ont également été proposées afin d'améliorer la performance globale de cette méthode. Ces extensions portent sur les points suivants :

- La méthode de regroupement d'attributs proposée utilise un algorithme de génération d'itemsets fréquents pour détecter les associations intéressantes entre les attributs. Dans ce contexte, plusieurs algorithmes de génération d'itemsets fréquents peuvent être utilisés. Le choix de l'algorithme approprié est discuté dans ce rapport et des tests expérimentaux sont fournis.
- Les algorithmes de génération d'itemsets fréquents sont généralement applicables sur des données booléennes. Afin de traiter des variables numériques, une phase de discrétisation est requise. Dans ce projet de thèse, nous avons étudié différentes méthodes de discrétisation possibles et avons discuté l'effet de chaque type de discrétisation sur le résultat de la méthode SIFRA. Cette étude nous a menés à sélectionner le type des discrétisations supervisées et multi-variées comme étant le type le plus adapté dans notre contexte.

Plan et guide de lecture

- **Chapitre 1**

Dans le premier chapitre, intitulé "Apprentissage à base de règles", nous commençons par une brève étude bibliographique sur les systèmes d'apprentissage artificiel qui génèrent des connaissances sous forme de règles. Nous focalisons ensuite la présentation de l'état de l'art sur deux axes différents de l'apprentissage à base de règles : nous étudions d'abord une méthode d'apprentissage non-supervisé qui consiste en la découverte des règles d'association, et présentons ensuite une méthode d'apprentissage

supervisé qui correspond aux systèmes de classification à base de règles floues. Enfin, nous présentons l'utilisation des systèmes de classification à base de règles floues sur des données de grande dimension et nous exposons différentes méthodes de réduction de la complexité de ces systèmes.

- **Chapitre 2**

Le chapitre 2 est essentiellement consacré à la description des méthodes d'ensemble en général et de l'approche de regroupement d'attributs en particulier. Cette dernière est un type particulier de méthodes d'ensemble qui décompose les attributs descriptifs en des sous-groupes d'attributs liés. Plusieurs méthodes de recherche de dépendances entre les attributs sont ensuite présentées, à savoir la recherche de corrélation linéaire, l'information mutuelle, etc. Finalement, nous consacrons le reste de ce chapitre à présenter une méthode d'ensemble qui utilise l'approche de regroupement d'attributs dans le contexte des systèmes de classification flous. Cette méthode, nommée SIFCO, utilise la recherche de corrélation linéaire pour détecter les attributs liés.

- **Chapitre 3**

Le chapitre 3 est dédié à la présentation de notre première contribution qui consiste en la proposition d'une méthode de regroupement d'attributs basée sur le concept des règles d'association. Nous montrons que cette méthode, appelée SIFRA, est capable de détecter différents types d'association entre des attributs numériques et/ou qualitatifs. Nous présentons les principales étapes de cette approche, ainsi qu'une discussion des différents choix adoptés dans la construction de la méthode. À la fin de ce chapitre, nous fournissons une analyse du coût global de la méthode proposée et nous suggérons quelques pistes pour réduire ce coût.

- **Chapitre 4**

Le chapitre 4 porte sur la proposition des extensions possibles de l'approche SIFRA. La première extension concerne la méthode de regroupement d'attributs supervisé (SIFRA-SAR) qui vise à prendre en compte l'information liée à la classe dans la phase de regroupement d'attributs. Ainsi, à part l'analyse des associations entre les attributs, le pouvoir prédictif des groupes d'attributs est aussi étudié et pris en compte lors de la formation des groupes d'attributs liés. La deuxième extension présentée dans ce chapitre consiste en l'utilisation d'une méthode de discrétisation supervisée et multi-variée dans la phase de regroupement d'attributs. Comme la procédure de discrétisation a un effet considérable sur les groupes d'attributs formés et sur la précision des classificateurs construits, le choix de la méthode de discrétisation appropriée a été analysé dans ce chapitre. Ainsi, une discrétisation supervisée et multi-variée qui prend en compte les inter-dépendances entre les attributs, d'une part, et l'information de l'attribut *Classe*, d'autre part, peut être intéressante dans notre contexte.

- **Chapitre 5**

Le dernier chapitre est consacré à l'implémentation et l'évaluation de notre système SIFRA ainsi que ses différentes versions et extensions. Nous présentons les différentes études expérimentales et comparatives menées dans le cadre de cette thèse afin de valoriser les apports de nos contributions.

- **Conclusion et Perspectives**

Nous concluons ce rapport par une synthèse des différentes contributions de ce travail. Nous présentons enfin une discussion sur les limites de nos solutions, ainsi qu'une exploration des diverses perspectives possibles de ce travail.

Apprentissage à base de règles

Introduction

Dès l'apparition des ordinateurs, les chercheurs se sont efforcés de construire des machines capables d'apprendre et d'accomplir des tâches qui requièrent de l'intelligence. L'apprentissage automatique représente l'une des facettes les plus importantes et ambitieuses de l'intelligence artificielle. Nous consacrons ce chapitre à présenter des méthodes d'apprentissage qui reposent sur la génération de règles. L'apprentissage à base de règles consiste à produire des modèles contenant des règles, descriptions compréhensibles par l'être humain.

Nous présentons dans la première partie de ce chapitre l'apprentissage automatique et ses deux grandes catégories : l'apprentissage supervisé et l'apprentissage non-supervisé. Ensuite, nous nous focalisons dans la deuxième partie sur un type spécifique d'apprentissage non supervisé, qui est celui des règles d'association. Ce type d'apprentissage, sur lequel se base notre travail, vise à extraire des motifs fréquents ou des associations intéressantes entre les éléments d'une base de données. Enfin, nous nous intéressons dans la dernière partie du chapitre à un type d'apprentissage, supervisé cette fois-ci, qui correspond aux systèmes de classification à base de règles floues. Ces systèmes, qui sont au cœur de notre travail, génèrent à partir des exemples d'apprentissage un modèle de prédiction compréhensible composé de règles de classification de type SI-Alors. Par ailleurs, l'avantage de ces systèmes est qu'ils utilisent des notions de la logique floue pour traiter les incertitudes et imperfections des données.

1.1 Apprentissage automatique

Plusieurs définitions de l'apprentissage artificiel (ou apprentissage automatique) ont été proposées dans la littérature. Parmi ces définitions, Antoine Mazières présente l'apprentissage artificiel, ou " machine learning ", comme étant un ensemble de méthodes permettant d'établir, à partir de données, des modèles de prise de décision, de prédiction ou de

classification [Mazieres, 2016]. De son côté, Tom Mitchell définit l'apprentissage automatique comme étant l'étude des algorithmes qui permettent aux programmes de s'améliorer automatiquement par expérience [Mitchell, 1997]. Selon Cornuéjols et al., l'apprentissage artificiel s'intéresse à l'écriture des programmes d'ordinateurs capables de s'améliorer automatiquement au fil du temps, soit sur la base de leur propre expérience, soit à partir de données antérieures fournies par d'autres programmes [Cornuéjols et al., 2003]. En résumé, l'apprentissage artificiel représente le champ d'étude où l'on essaie de reproduire la capacité de l'homme à apprendre. L'apprentissage artificiel peut avoir plusieurs objectifs tels que : l'acquisition des connaissances, la reconnaissance des formes, la prédiction, l'extraction et compression d'informations, etc. Les algorithmes d'apprentissage artificiel ont montré leur efficacité dans divers domaines tels que le domaine médical pour l'aide au diagnostic médical [Cruz and Wishart, 2006] [Mena et al., 2012] ; la reconnaissance des formes pour reconnaître par exemple les caractères manuscrits [Lorigo and Govindaraju, 2006] ; la robotique [Nguyen-Tuong and Peters, 2011], etc. La figure 1, reprise du site internet de l'équipe de recherche "Équipe Apprentissage Machine", du laboratoire LEIBNIZ2, montre la diversité des domaines d'étude de l'apprentissage artificiel [Feuilloy, 2009].

Dans [Carbonell et al., 1983], les auteurs ont proposé plusieurs manières de classer les systèmes d'apprentissage automatique selon différents critères : la stratégie utilisée, la forme des connaissances acquises ou les domaines d'application. Les systèmes d'apprentissage artificiels génèrent des connaissances. Ces connaissances peuvent prendre différentes formes telles que des règles de classification, des expressions logiques, des heuristiques de résolution de problèmes, etc. On peut alors classer les approches de l'apprentissage artificiel selon le type des connaissances acquises ; on distingue notamment : les arbres de décision, les règles de production, les grammaires formelles, les graphes et réseaux,... Dans le cadre de cette thèse, on s'intéresse aux systèmes d'apprentissage qui fournissent des règles de production. Ce type de système fournit un modèle facile à interpréter et à comprendre par l'utilisateur final. Cette caractéristique d'intelligibilité des modèles d'apprentissage est très utile dans divers domaines tels que le domaine médical ou le domaine commercial où l'utilisateur final n'est pas un informaticien et où l'intelligibilité des modèles est indispensable.

Une autre classification de l'apprentissage artificielle se base sur la nature des données, si elles sont étiquetées ou pas ; cette classification distingue quatre types d'apprentissage :

- Apprentissage supervisé : en apprentissage supervisé, les classes des données sont connues a priori ; on parle de données étiquetées. Le rôle de l'apprentissage supervisé est de créer un modèle, à partir des données étiquetées, ce modèle sera capable de prédire la classe de nouvelles instances. [Berka and Rauch, 2010]
- Apprentissage non supervisé : contrairement à l'apprentissage supervisé, les données dans ce type d'apprentissage sont non étiquetées et le système ne reçoit aucune information sur la sortie désirée de son apprentissage. L'objectif de l'apprentissage est alors de dégager des motifs ou des représentations des données qui peuvent être

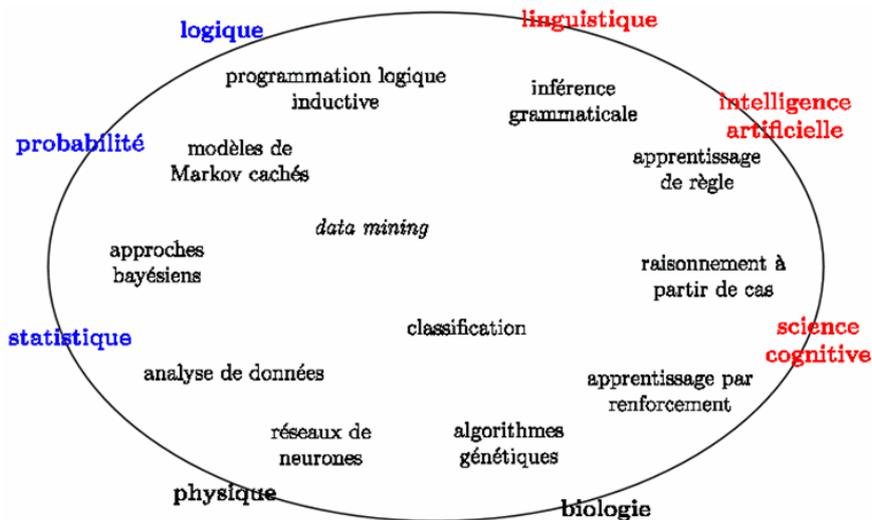


Figure 1 — Illustration des domaines scientifiques apparentés à l'apprentissage artificiel [Feuilloy, 2009]

plus tard utiles pour d'autres systèmes d'apprentissage (pour prendre des décisions, prédire d'autres données,...). [Ghahramani, 2004]

- Apprentissage semi supervisé : l'apprentissage semi-supervisé utilise en même temps les données étiquetées et non étiquetées. L'objectif est le même que l'apprentissage supervisé mais avec un nombre plus réduit de données labellisées. [Zhu and Goldberg, 2009]
- Apprentissage par renforcement : dans ce type d'apprentissage, le système interagit avec son environnement en utilisant le principe de rétroaction : à chaque fois que le système effectue une action, il reçoit une récompense ou une sanction pour juger la qualité de son action et son impact sur l'environnement [Mitchell, 1997]. L'objectif est d'apprendre à réagir de façon à maximiser les récompenses et minimiser les sanctions dans ses futures actions. [Ghahramani, 2004]

Plus de détails sur ces types d'apprentissage sont fournis dans [Russel and Norvig, 2002] et [Cornuéjols et al., 2003]. On s'intéresse dans le cadre de cette thèse aux deux types d'apprentissage supervisé et non supervisé, que nous présentons ci-dessous.

1.1.1 Apprentissage supervisé

L'objectif de l'apprentissage supervisé ou des problèmes de classification est de construire, à partir d'un échantillon d'apprentissage (des observations connues déjà étiquetées), un modèle qui permet d'associer, à une nouvelle observation, une classe Y . La sortie Y peut être quantitative ou bien qualitative. Dans le cas où la sortie Y est quantitative, on parle généralement d'un problème de régression [Cornillon and Matzner-Lober, 2007].

Dans le cas contraire, on parle de discrimination [Saporta, 2006], de classement ou de reconnaissance des formes [Besse, 2009]. Notre travail se place dans le cas des problèmes de classification où la sortie Y est qualitative. Dans ce type d'apprentissage, on distingue deux phases :

- La phase d'apprentissage : l'objectif de cette phase est de définir une fonction de classement f , à partir de la base d'apprentissage ; cette fonction permet de prédire la classe d'un exemple inconnu.



Figure 2 — Phase d'apprentissage

- La phase de reconnaissance : cette phase consiste à associer à un nouvel exemple une classe Y_i parmi les C classes du problème, et ce en utilisant la fonction f trouvée dans la phase précédente.

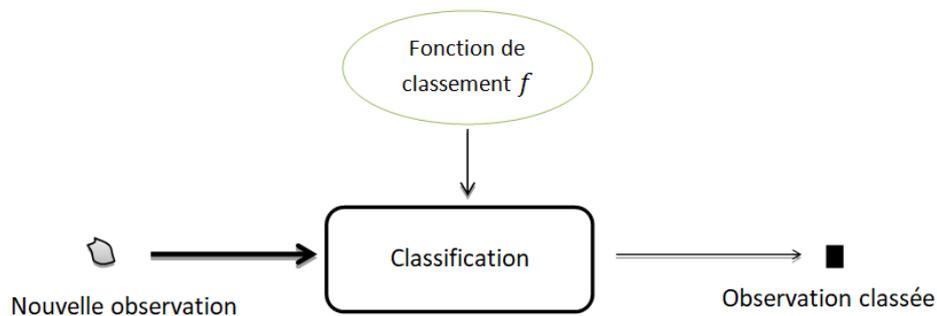


Figure 3 — Phase de classification

Parmi les méthodes d'apprentissage supervisé, on trouve les techniques de régression linéaire et non linéaire [Cornillon and Matzner-Lober, 2007], les méthodes à base d'inférence bayésienne [Naim et al., 2011], les arbres de décision [Kohavi and Quinlan, 2002, Rokach and Maimon, 2015, Wang et al., 2015], etc. Dans le cadre de cette thèse, nous ne nous intéressons qu'aux approches par génération de règles, elles sont de la forme :

Si [PRÉMISSE] Alors [ACTION]

Où la prémisse est une conjonction de conditions (variable = valeur). L'interprétation de cette forme de représentation est : si une situation satisfait la [PRÉMISSE], alors on peut inférer la [ACTION].

Nous présentons ici quelques méthodes de classification à base de règles, notamment les arbres de décision, la méthode SUCRAGE et les systèmes d'inférence floue.

1.1.1.1 Les arbres de décision

Les arbres de décision présentent l'une des fameuses techniques d'apprentissage à base de règles; on trouve par exemple les méthodes ID3 [Quinlan, 1986], CART [Breiman et al., 1984] et C4.5 [Quinlan, 1993].

L'idée de cette méthode est de réaliser la classification d'un objet par une suite de tests sur les attributs qui le décrivent. Ces tests sont organisés en utilisant la structure d'un arbre où chaque noeud dénote un test sur un attribut et chaque branche représente une sortie possible de ce test. Une feuille de cet arbre désigne une des C classes (la classe à prédire) [Cornuéjols et al., 2003]. Un exemple d'arbre de décision est présenté dans la figure 4 où on traite la possibilité que le patient soit malade ou pas selon sa température et son état de gorge.

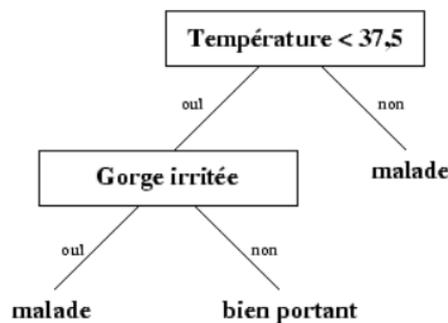


Figure 4 — Exemple d'arbre de décision

Les algorithmes d'arbre de décision choisissent à chaque noeud l'attribut le plus discriminant, qui sépare le mieux les différentes classes. Idéalement, les observations sont partitionnées en des sous-groupes purement classifiés (c'est-à-dire les données de chaque partition appartiennent à la même classe) [Han et al., 2012]. Ainsi, des mesures de sélection d'attributs sont utilisées pour choisir à chaque étape le meilleur attribut discriminant. Parmi les mesures de sélection d'attributs utilisées, ID3 utilise la mesure gain d'information [Quinlan, 1986], C4.5 utilise le gain ratio [Quinlan, 1993] et CART utilise l'index de Gini [Breiman et al., 1984].

Les branches d'un arbre de décision peuvent facilement se transformer en une base de règles de type Si-Alors. Un exemple de règle obtenue à partir de l'arbre de la figure 4 est :

SI Température < 37.5 **ET** Gorge irritée **ALORS** patient malade.

Les méthodes d'arbre de décision sont parmi les méthodes d'apprentissage les plus utilisées dans la littérature. Ainsi, elles fournissent des résultats de classification très intéressants et construisent des règles de classification intelligibles et compréhensibles par l'être humain. Néanmoins, le désagrément de ces méthodes est qu'elles ne prennent pas en considération la dépendance entre les attributs : elles ne considèrent qu'un seul attribut à la fois. On appelle ce type de méthodes mono-attributs ou monothétiques. D'autres méthodes, dites méthodes polythétiques ou multi-attributs, prennent en compte

plusieurs attributs simultanément (les attributs qui sont dépendants) et bénéficient de cette dépendance dans l'opération de classification. Parmi ces méthodes, on présente dans le paragraphe suivant la méthode SUCRAGE [Borgi, 1999, Borgi and Akdag, 2001].

1.1.1.2 La méthode SUCRAGE

SUCRAGE est une méthode d'apprentissage supervisé multi-attributs qui prend en compte la dépendance entre les attributs afin de construire les règles de classification. La sélection en un bloc d'attributs liés est réalisée par une recherche de corrélation linéaire entre les attributs. Puis, les attributs corrélés sont regroupés et apparaissent ensemble dans la prémisse d'une même règle [Borgi, 1999, Borgi and Akdag, 2001].

SUCRAGE a été initialement proposé avec un moteur d'inférence exacte : seules les règles dont les prémisses sont vérifiées par la nouvelle observation à classer sont déclenchées. Ensuite, ce moteur a été étendu à une inférence approximative qui permet de déclencher les règles pas trop éloignées de la nouvelle observation [Borgi, 2008]. Le raisonnement approximatif permet de générer de nouvelles règles plus générales et de prendre en compte les imprécisions des observations.

Une autre extension de SUCRAGE a été aussi proposée dans [Soua et al., 2013]. Celle-ci utilise la théorie de la logique floue afin de prendre en compte les imprécisions et incertitudes des données. Une description détaillée de ce travail dont nous nous inspirons est fournie dans le chapitre 2.

1.1.1.3 Systèmes d'inférence floue

Les systèmes d'inférence floue (SIF) sont des méthodes d'apprentissage supervisé par génération de règles. Elles reposent sur la logique floue [Zadeh, 1965] et vise à s'approcher du raisonnement humain à travers l'utilisation de la théorie des sous-ensembles flous. Les travaux portant sur les SIF ont commencé dans le domaine du contrôle flou où les données de sorties sont continues [Chaudhari et al., 2014, Walia et al., 2015]. Les SIF ont été également utilisés dans le domaine de la classification où la sortie est une valeur discrète qui correspond à une classe parmi un ensemble de classes prédéfinies [Antonelli et al., 2014, Dehzangi et al., 2007, Mansoori and Shafiee, 2016]. Ce type de systèmes d'inférence floue est connu sous le nom de Système de Classification à Base de Règles Floues (SCBRF). Nous nous intéressons dans ce travail à ce type de systèmes flous liés au domaine de la classification. Nous consacrons, plus loin, une partie de ce chapitre à une étude bibliographique approfondie sur ces systèmes (SCBRF).

1.1.2 Apprentissage non supervisé

Comme nous l'avons mentionné, ce type d'apprentissage utilise des données non étiquetées et ne dispose d'aucune supervision ou récompense d'un expert pour lui indiquer

comment les données doivent être traitées ou représentées. Un des problèmes les plus connus en apprentissage non-supervisé est le regroupement en classes ("Clustering" en anglais). L'objectif du Clustering est de regrouper les données en classes (des "clusters") en maximisant la similarité intra-classes et minimisant la similarité inter-classes [Cornuéjols et al., 2003]. Plusieurs algorithmes de Clustering ont été proposés dans la littérature comme l'algorithme k-means [MacQueen, 1967], TGCA [He and Tan, 2012] et fuzzy c-means [Havens et al., 2012].

L'extraction des Règles d'Association (RA) représente un autre type d'apprentissage non supervisé. Kotsiantis et Kanellopoulos [Kotsiantis and Kanellopoulos, 2006] l'ont défini comme une technique qui vise à extraire des corrélations intéressantes, des motifs fréquents, ou des associations entre les ensembles d'éléments dans des bases de données transactionnelles ou autres dépôts de données. Ce type d'apprentissage peut être par exemple appliqué dans le cas de supermarché où les tickets de caisse des clients sont analysés afin d'associer les articles qui sont souvent achetés ensemble par les clients et découvrir par la suite leurs habitudes d'achat. Les relations découvertes aideront à mieux agencer les rayons du supermarché et à proposer les promotions de vente aux clients. Plus de détails sur les règles d'association sont fournis dans la section suivante.

1.2 Partie 1 : Apprentissage par Règles d'Association

La découverte des Règles d'Association (RA) est une approche très utilisée d'apprentissage non supervisé. Le problème de l'extraction des règles d'association est assez ancien dans le domaine de l'apprentissage et la fouille de données. Il a été introduit dans les années 90 dans les travaux de [Agrawal et al., 1993] [Agrawal and Srikant, 1994] dans le but d'analyser les transactions de ventes dans les supermarchés afin de découvrir les habitudes d'achat des clients. Ce problème est connu sous le nom de panier de la ménagère. Dans une telle base de données transactionnelle, chaque exemple (observation) correspond à une transaction alors que les différents attributs binaires correspondent aux différents articles vendus dans le supermarché. Un exemple d'une règle d'association qui peut être découvert est le suivant : "Si un client achète du pain et du beurre, il est très probable qu'il achète aussi du lait". L'intérêt des règles d'association découvertes dans ce cas consiste à aider les gérants du supermarché à bien agencer les articles dans les rayons du supermarché et à proposer les promotions de vente aux clients.

1.2.1 Définitions et notions de base

On se situe dans le cadre d'une base de données binaires D , munie d'un ensemble de N attributs $I = I_1, I_2, \dots, I_N$, appelés aussi items, et d'un ensemble de m transactions, $T = t_1, t_2, \dots, t_m$. Chaque transaction t_i possède un identifiant TID et désigne un sous-ensemble de I (le sous-ensemble des attributs I_j qui valent 1 pour t_i). Un sous-ensemble d'items $X \subseteq I$ est appelé itemset ou motif. Appelons le nombre d'attributs dans

un itemset sa taille; un itemset de taille k est un k -itemset. On dit qu'une transaction t vérifie l'itemset X si pour chaque attribut I_j appartenant à X , I_j apparaît dans t . [Agrawal and Srikant, 1994](#)

Une Règle d'Association RA est une implication $X \rightarrow Y$ où X et Y sont deux itemsets et $X \cap Y = \emptyset$. X désigne la prémisse de la règle (appelé aussi son antécédent) tandis que Y représente sa conséquence. Cette implication signifie que les transactions qui vérifient le motif X ont tendance à vérifier également le motif Y . Afin d'évaluer la qualité d'une RA, deux mesures sont généralement utilisées : le support et la confiance. Ces deux mesures, proposées dans [Agrawal and Srikant, 1994](#), sont définies comme suit :

- Le support d'une règle d'association $X \rightarrow Y$ est la proportion des transactions qui vérifient à la fois X et Y

$$Supp(X \rightarrow Y) = P(X \cup Y) \simeq \frac{n(XY)}{m} \quad (1)$$

- La confiance de la règle $X \rightarrow Y$ est la proportion des transactions qui vérifient Y parmi celles qui vérifient X

$$Conf(X \rightarrow Y) = \frac{P(X \cup Y)}{P(X)} \simeq \frac{n(XY)}{n(X)} \quad (2)$$

Notons que $n(XY)$ est le nombre de transactions qui vérifient à la fois X et Y ; $n(X)$ est le nombre de transactions qui vérifient les items de X et m est le nombre total de transactions.

Le support permet de mettre en évidence la fréquence d'apparition des motifs. Il sert à vérifier que les motifs se produisent dans suffisamment de transactions; en d'autres termes, si les motifs sont rares dans la base de transactions, alors il serait abusif de tirer des conclusions utiles [Brin et al., 1997](#). D'un autre côté, la confiance permet de souligner la force de l'implication d'une règle d'association. Supposons que la RA " $pain \wedge beurre \rightarrow lait$ " ait une confiance de 0.67, cela veut dire que 67% des transactions qui contiennent pain et beurre contiennent aussi lait.

Nous considérons comme exemple la base de données transactionnelle illustrée dans le tableau [1](#). Elle est composée de 5 transactions (5 paniers d'achat) décrites par 5 attributs binaires (Pain, Lait, Couche, Beurre et Oeuf). Considérons la règle suivante : $Pain, Beurre \rightarrow Lait$. Le support de la règle est $\frac{2}{5} = 0.4$ puisque le nombre de transactions qui contiennent simultanément les attributs Beurre, Lait et Pain est égal à 2 et le nombre total de transactions est égal à 5. D'autre part, la confiance de la règle est $\frac{2}{3} = 0.67$. Ce nombre est obtenu en divisant le nombre de transactions contenant les items Beurre, Lait et Pain par le nombre de transactions contenant les items Pain et Beurre.

TID	Pain	Lait	Couche	Beurre	Oeuf
1	1	1	0	0	0
2	1	0	1	1	1
3	0	1	1	1	0
4	1	1	1	1	0
5	1	1	0	1	0

Tableau 1 — Extrait d'une base de données transactionnelle

Le modèle support-confiance, fondé par Agrawal et al. [Agrawal et al., 1993], consiste à extraire les règles d'association qui vérifient un certain seuil des valeurs support et confiance, c'est-à-dire dont le support et la confiance sont supérieurs ou égaux à des seuils fixés au préalable par l'utilisateur, notés *minsupp* et *minconf*.

Le problème d'extraction des règles d'association se décompose en deux sous-problèmes :

1. Recherche des itemsets fréquents : un itemset fréquent est un itemset qui possède un support supérieur ou égal au seuil *minsupp*. Cette étape consiste à utiliser un algorithme qui recherche et détermine les itemsets fréquents parmi tous les itemsets possibles. Il est à noter que pour un ensemble de N items, le nombre d'itemsets fréquents qui peuvent être générés est 2^N . Nous présentons dans le paragraphe suivant un état de l'art sur quelques algorithmes d'extraction d'itemsets fréquents proposés dans la littérature.
2. Génération des règles d'association : la deuxième étape consiste à générer toutes les règles d'association à partir des itemsets fréquents générés précédemment. Pour chaque itemset fréquent X , il s'agit de déterminer toutes les combinaisons de règles possibles $Y \rightarrow (X \setminus Y)$ où Y est un sous-ensemble de X ($Y \subset X$). Afin de limiter le nombre de règles extraites, seules les règles d'association qui possèdent une confiance supérieure ou égale au seuil *minconf* sont valides et sont extraites.

1.2.2 Les algorithmes d'extraction d'itemsets fréquents

Plusieurs travaux se sont concentrés sur la recherche d'itemsets fréquents depuis plusieurs années; les algorithmes AIS [Agrawal et al., 1993] et Apriori [Agrawal and Srikant, 1994] sont les premiers algorithmes proposés dans la littérature. AIS est le premier algorithme proposé par Agrawal pour extraire les itemset fréquents. Néanmoins, son temps d'exécution énorme et sa complexité élevée ont favorisé l'apparition de l'algorithme Apriori, le plus connu et utilisé dans la littérature. La fameuse propriété d'anti-monotonie d'Apriori l'a distingué de tous les autres algorithmes. Cependant, l'apparition de grandes masses de données a mis en cause l'utilisation de l'algorithme Apriori. En effet, bien que la propriété d'anti-monotonie puisse réduire significativement le nombre d'itemsets fréquents générés, il nécessite cependant des accès multiples à la base de données, ce qui alourdit son traitement. Plusieurs autres travaux se

sont concentrés sur le développement d'autres méthodes d'extraction d'itemsets fréquents plus efficaces et moins complexes tels que les algorithmes FP-Growth [Han et al., 2000], Eclat [Zaki et al., 1997], FIN [Deng and Lv, 2014], Prepost [Deng et al., 2012] et Prepost+ [Deng and Lv, 2015],...

Nous fournissons dans cette section une description détaillée de l'algorithme Apriori, comme étant le plus connu des algorithmes d'itemsets fréquents, et nous présentons ensuite un aperçu sur les autres algorithmes proposés dans la littérature et sur les différentes structures de données utilisées.

1.2.2.1 Algorithme Apriori

L'algorithme Apriori [Agrawal and Srikant, 1994] consiste en deux étapes : d'abord il génère les itemsets candidats (qui sont susceptibles d'être des motifs fréquents), puis il détermine parmi ceux-ci les itemsets fréquents en analysant la base de données et en comptant le support de chacun ; l'itemset candidat ayant un support supérieur ou égal au seuil *minsupp* est considéré comme itemset fréquent. L'algorithme Apriori est le premier algorithme de recherche d'itemsets fréquents qui inclut la propriété d'anti-monotonie. Cette propriété dit que :

"Si un itemset X est fréquent alors tout itemset $X_1 \subseteq X$ est aussi fréquent. Autrement dit, chaque sous-ensemble d'un motif fréquent est aussi nécessairement fréquent."

Une conséquence directe de cette règle est que les sur-ensembles des itemsets non-fréquents sont aussi non-fréquents. Du coup, si un motif non-fréquent se présente, alors tous les motifs situés au dessus (tous les sur-ensembles de ce motif) sont aussi non-fréquents et par conséquent il est inutile de les générer [Agrawal and Srikant, 1994]. Cette propriété est utilisée lors de l'étape de génération des itemsets candidats. D'abord, les itemsets candidats de taille k (k -itemsets) sont générés par une jointure des itemsets fréquents de taille $k - 1$. Ensuite, la propriété d'anti-monotonie va permettre d'élaguer les candidats non fréquents de l'ensemble généré. Cette propriété permet de générer un nombre beaucoup plus restreint de motifs candidats que celui généré par l'algorithme AIS.

La description détaillée de l'algorithme Apriori se résume dans les étapes de l'algorithme 1 suivant [Agrawal and Srikant, 1994] :

Les étapes de l'algorithme Apriori se résument comme suit :

- D'abord, on génère les motifs fréquents de taille 1 (ligne 1) en calculant le support de chaque attribut dans la base de transactions et on garde seulement les 1-itemsets qui possèdent un support supérieur ou égal à *minsupp*
- Puis, les k -itemsets candidats sont générés à partir des $(k-1)$ -itemsets fréquents (en appelant la fonction Apriori-Gen)

- Ensuite le support de chaque k-itemset candidat est calculé comme suit (ligne 4 à 9) : pour chaque transaction, on détermine l'ensemble C_t des k-itemsets candidats qui sont vérifiés par cette transaction t (à travers la fonction sous-ensemble(C_k, t)), puis on incrémente de 1 le support de chaque itemset de C_t .
- Enfin, seulement les k-itemsets candidats dont le support dépasse le seuil prédéfini $minsupp$ sont gardés (ligne 10). Ces itemsets gardés forment l'ensemble de k-itemsets fréquents.

Algorithme 1 : Apriori

Entrée : Base de transactions T , $minsupp$

Sortie : F : ensemble des motifs fréquents et de leurs supports

```

1)  $F_1 \leftarrow \{1\text{-itemsets fréquents}\}$ 
2) for ( $k \leftarrow 2$ ;  $F_{k-1} \neq \emptyset$ ;  $k++$ ) do
3)    $C_k \leftarrow \text{Apriori-Gen}(F_{k-1})$ 
4)   for chaque transaction  $t \in T$  do
5)      $C_t \leftarrow \text{sous-ensemble}(C_k, t)$ 
6)     for chaque candidat  $c \in C_t$  do
7)        $Supp(c).count++$ 
8)     end for
9)   end for
10)   $F_k \leftarrow \{c \in C_k / Supp(c) \geq minsupp\}$ 
11) end for
12) retourner  $F \leftarrow \bigcup_k F_k$ 

```

Concernant l'étape de génération des itemsets candidats, la procédure Apriori-Gen est utilisée ; elle se résume en deux étapes : une étape de jointure et une étape d'élagage.

- Les k-itemset candidats, noté C_k , sont générés en effectuant une jointure des (k-1)-itemsets fréquents, noté F_{k-1} . Deux itemsets p et q de taille $k-1$ forment un itemset c de taille k s'ils possèdent $k-2$ attributs en commun.
- L'étape d'élagage permet de s'assurer, pour chaque k-itemset candidat généré, que tous ses sous-ensembles de taille $k-1$ sont fréquents ; c'est le rôle de la propriété d'anti-monotonie d'Apriori.

1.2.2.2 Application sur des données numériques

Le problème d'extraction des RA défini dans [Agrawal et al., 1993, Agrawal and Srikant, 1994] est un problème de Règles d'Association Booléennes (BAR) qui considère seulement la présence des attributs binaires dans les bases de données transactionnelles. Cependant, la plupart des bases de données réelles contiennent généralement des attributs quantitatifs (âge, salaire,...) et/ou qualitatifs (couleur, sexe,

état civil,...). Les variables de type booléen sont un type particulier des variables qualitatives où le domaine de variation est limité à $\{0,1\}$. Il existe deux possibilités pour traiter les données quantitatives. La première solution consiste à utiliser un algorithme spécifique comme Min-Apriori [Han et al., 1997] et C-Apriori [Zhou et al., 2013]. Min-Apriori et C-Apriori étendent l'algorithme Apriori sur des données quantitatives. Pour ce faire, de nouvelles définitions du calcul de la mesure support sont proposées afin de prendre en compte les valeurs continues des attributs numériques. La deuxième solution consiste à partitionner les variables quantitatives en des intervalles consécutifs et considérer chaque intervalle comme une valeur catégorique. Cette idée a été proposée dans [Srikant and Agrawal, 1996] afin de pouvoir générer des Règles d'Association en considérant aussi bien les variables booléennes que celles quantitatives et qualitatives. De plus, en utilisant cette méthode, tout algorithme de génération d'itemsets fréquents peut être appliqué [Srikant and Agrawal, 1996].

L'approche utilisée dans [Srikant and Agrawal, 1996] consiste à mapper le problème de RAs quantitatives au problème classique de RAs booléennes. Ce mappage fait appel au codage disjonctif complet et se fait de la manière suivante :

- Pour chaque valeur d'un attribut catégorique, la paire $\langle \text{attribut}, \text{valeur} \rangle$ est mappée à un attribut booléen.
- Pour un attribut quantitatif, on discrétise son domaine de variation en des intervalles et chaque paire $\langle \text{attribut}, \text{intervalle} \rangle$ est mappée à un attribut booléen.
- Si l'attribut quantitatif ne possède que quelques valeurs, alors il peut être traité comme un attribut catégorique ; c.à.d. chaque valeur de l'attribut constitue une paire $\langle \text{attribut}, \text{valeur} \rangle$ et est mappée à un attribut booléen.

Autrement dit, le couple $\langle \text{attribut } i, \text{valeur } j \rangle$ vaut "1" si l'attribut X_i possède la valeur j dans l'enregistrement de la base, et vaut "0" sinon. Ensuite, l'algorithme Apriori est utilisé pour déterminer les associations entre ces nouveaux attributs booléens formés (ces couples).

Afin d'illustrer cette approche, nous enchaînons par un exemple cité dans [Srikant and Agrawal, 1996]. Le tableau 2 décrit les enregistrements d'une la base relationnelles qui possède 3 attributs : 2 attribut quantitatifs (âge et nombre de voitures) et un attribut qualitatif (marié). L'attribut quantitatif "Âge" est partitionné en 2 intervalles : $[20..29]$ et $[30..39]$. Chaque intervalle est mappé à un attribut booléen. L'attribut qualitatif "Marié" possède deux valeurs : Oui et Non. Chaque valeur forme un attribut booléen. Enfin, l'attribut quantitatif "Nombre de voitures" possède seulement 3 valeurs, il n'est donc pas partitionné en des intervalles mais chaque valeur est mappée en un attribut booléen. Le tableau 3 représente le mappage du problème décrit dans Tab. 2 en un problème de RAs booléennes.

Un exemple de RA quantitative est alors le suivant :

Enregistrement	Âge	Marié	Nombre de voitures
e_1	23	Non	1
e_2	25	Oui	1
e_3	29	Non	0
e_4	34	Oui	2
e_5	38	Oui	2

Tableau 2 — Base de données relationnelle

Enregistrement	Âge : [20..29]	Âge : [30..39]	Marié : Oui	Marié : Non	Nombre de voitures : 0	Nombre de voitures : 1	Nombre de voitures : 2
e_1	1	0	0	1	0	1	0
e_2	1	0	1	0	0	1	0
e_3	1	0	0	1	1	0	0
e_4	0	1	1	0	0	0	1
e_5	0	1	1	0	0	0	1

Tableau 3 — Mappage de la base de données relationnelle à une base de données booléenne

$\langle \text{Âge}, [30..39] \rangle \wedge \langle \text{Marié}, \text{Oui} \rangle \rightarrow \langle \text{Nombre de voitures}, 2 \rangle$ avec 100% de confiance

Cette RA affirme que 100% des personnes âgées entre 30 et 39 ans et mariées, possèdent deux voitures.

Il existe d'autres travaux dans la littérature qui ont cherché à appliquer le concept des RAs sur des variables quantitatives en utilisant la logique floue. Les auteurs dans [Hong et al., 2001] ont proposé un algorithme qui intègre le concept des sous-ensembles flous avec l'algorithme Apriori afin de permettre la découverte des Règles d'Association sur des données quantitatives. Comme il a été proposé dans [Srikant and Agrawal, 1996], le domaine de variation de chaque attribut est partitionné mais cette fois-ci en des sous-ensembles flous et non pas en des intervalles. Ces sous-ensembles flous issus de l'étape de partitionnement sont appelés des termes linguistiques. Ainsi, les valeurs des attributs, pour chaque observation, sont transformées en des degrés d'appartenance aux différents termes linguistiques. La méthode de [Hong et al., 2001] a prouvé son efficacité en améliorant la précision de classification grâce à l'utilisation des fonctions d'appartenance floues. Cependant, elle ne peut pas être utilisée sur des bases de données contenant des variables aussi bien qualitatives que quantitatives, alors que la méthode proposée dans [Srikant and Agrawal, 1996] est capable de traiter les différents types de données.

1.2.2.3 Autres algorithmes

Une présentation détaillée d'Apriori (dans le paragraphe précédent) nous a montré que ses points forts résident d'une part dans sa simplicité et d'autre part dans l'utilisation de la propriété d'anti-monotonie. Cette propriété permet de réduire le nombre de

génération d'itemsets candidats. Cependant, il a été démontré dans la littérature que l'algorithme Apriori possède un coût important lorsqu'il est appliqué sur des bases à haute dimension [Jiang and Wang, 2011] [Li and Shang, 2013] [Busarov et al., 2016]. En effet, Apriori nécessite plusieurs accès à la base de données pour générer les itemsets candidats et pour compter leurs supports. De plus, le nombre d'itemsets candidats générés augmente exponentiellement en fonction du nombre d'attributs dans la base de données traitée. Pour une base de données de N attributs, 2^N itemsets candidats peuvent être générés [Tan et al., 2005]. Afin de contourner ces problèmes, plusieurs travaux se sont concentrés sur l'amélioration de la performance de l'algorithme Apriori (comme AprioriTid [Agrawal and Srikant, 1994] et Partition [Savesere et al., 1995]) ou ont proposé d'autres algorithmes plus efficaces et moins complexes qu'Apriori. Concernant ce dernier cas, de nouvelles structures de données qui consomment moins d'espace mémoire et moins de temps de passage et de lecture, et qui génèrent le même ensemble d'itemsets fréquents, ont été proposées.

La structure de données FP-Tree (Frequent Pattern Tree) est l'une des structures de données les plus connues qui ont été proposées pour contourner la complexité de Apriori. L'algorithme FP-Growth [Han et al., 2000] utilise cette structure de données afin de générer tous les itemsets fréquents (le même ensemble d'itemsets généré par Apriori) sans générer les itemsets candidats et en faisant seulement deux accès à la base de données. Les enregistrements d'une base de données sont traversés seulement deux fois afin de construire la FP-Tree ; cette structure de données contient les items fréquents (1-itemsets fréquents) ainsi que leurs supports. Ensuite, l'algorithme FP-Growth utilise la FP-Tree afin de générer tous les itemsets fréquents sans avoir recours une autre fois à la base de données. Un exemple de FP-Tree est présenté dans la figure 5 [Busarov et al., 2016]. Dans cette figure, les éléments c, b, d, f et e représentent des items (des attributs) fréquents et le chiffre indiqué après chaque item représente son support.

Il a été prouvé, via plusieurs travaux expérimentaux, que l'algorithme FP-Growth est plus performant que l'algorithme Apriori pour les grandes bases en termes de temps d'exécution [Gyorodi et al., 2004] [Alhamzi et al., 2014] [Garg and Gulia, 2015]. De plus, les travaux dans [Gyorodi et al., 2004] ont montré que la performance de FP-Growth n'est pas trop influencée par l'augmentation du seuil *minsupp* tandis que la performance d'Apriori diminue avec l'augmentation de la valeur de *minsupp*.

Malgré la performance de FP-Growth, démontrée dans plusieurs travaux, les auteurs dans [Pei et al., 2007] ont affirmé que l'utilisation de cet algorithme n'est pas appropriée pour les bases de données vastes et dispersées parce que la FP-Tree devient très large et complexe. Pour cette raison, ils ont proposé un autre algorithme, H-mine, qui utilise une autre structure de données nommée H-struct [Pei et al., 2007]. H-struct stocke les éléments fréquents avec leurs références aux transactions correspondantes dans la base de données au lieu de stocker explicitement les transactions comme dans FP-Growth. FIN [Deng and Lv, 2014] est un autre algorithme qui utilise une autre structure de données "NodeSets" pour générer les itemsets fréquents. La structure NodeSets ne consomme pas

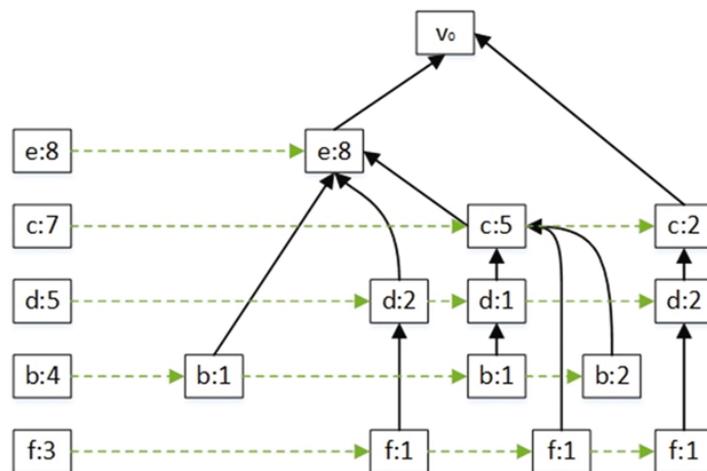


Figure 5 — Exemple de FP-Tree

[Busarov et al., 2016]

beaucoup de mémoire et est facile à construire [Bharati and Kanchan, 2016]. Les travaux dans [Bharati and Kanchan, 2016] et [Busarov et al., 2016] ont montré que cette structure de données est très efficace et que l'algorithme FIN est plus rapide que FP-Growth et H-mine. Récemment, un algorithme PréPost+ a été proposé dans [Deng and Lv, 2015]. Cet algorithme utilise en même temps trois structures de données différentes : N-list, PPC-tree et Set-enumeration-Tree. L'utilisation de trois structures de données requiert la consommation de plus d'espace mémoire que FP-Growth. Cependant, il a été démontré que cet algorithme est beaucoup plus efficace en termes de temps d'exécution comparé à plusieurs autres algorithmes de génération d'itemsets fréquents tels que FIN, H-mine et FP-Growth [Bharati and Kanchan, 2016], [Busarov et al., 2016]. Plus de détails sur ces différents algorithmes et structures de données peuvent être trouvés dans [Bharati and Kanchan, 2016], [Busarov et al., 2016] et [Alhamzi et al., 2014].

1.2.3 Les algorithmes d'extraction d'itemsets fermés et maximaux

Dans les bases de données qui possèdent un nombre important d'attributs, un grand nombre d'itemsets fréquents pourrait être généré. Un itemset de longueur l implique la présence de $2^l - 2$ autres itemsets fréquents (qui sont des sous-ensembles du premier). Étant donné que l pourrait être très grand dans ces bases, la méthode de génération des itemsets fréquents devient coûteuse en terme de temps de réponse et d'espace mémoire. Bien que plusieurs travaux se sont focalisés sur la réduction du coût des algorithmes d'extraction d'itemsets fréquents (comme les algorithmes FP-Growth [Han et al., 2000], FIN [Deng and Lv, 2014] et PréPost+ [Deng and Lv, 2015]), d'autres travaux ont proposé de diminuer le nombre des itemsets générés en gardant un sous-ensemble compact et en éliminant les informations redondantes dans l'ensemble des itemsets. Deux

types de méthodes ont été proposées dans ce contexte : les algorithmes d'extraction des itemsets fermés [Pasquier et al., 1999, Grahne and Zhu, 2005, Li et al., 2017] et les algorithmes d'extraction des itemsets maximaux [Gouda and Zaki, 2001, Burdick et al., 2001, Grahne and Zhu, 2003].

1.2.3.1 Algorithmes d'extraction d'itemsets fermés

La génération des itemsets fermés consiste à générer seulement les itemsets fréquents qui ne possèdent aucun sur-ensemble avec le même support. Les itemsets fermés constituent donc un ensemble non redondant minimal pour tous les itemsets fréquents et leurs supports [Pasquier, 2000b]. Tous les itemsets fréquents et leurs supports peuvent donc être déduits, à partir de l'ensemble des itemsets fermés, sans accéder de nouveau à la base de données. Cette propriété découle du fait que le support d'un itemset fréquent qui n'apparaît pas dans la liste des itemsets fermés est égal au support de son sur-ensemble fréquent. [Pasquier, 2000b]

Ces algorithmes présentent alors l'avantage de générer un ensemble réduit d'itemsets sans perdre aucune information. De plus, cette procédure est moins coûteuse que les méthodes de génération de tous les itemsets fréquents et cela en terme de temps de réponse et d'espace mémoire.

Parmi les algorithmes de génération des itemsets fermés, on trouve Charm [Zaki and Hsiao, 2002], Closet [Pei et al., 2000] et FPclose [Grahne and Zhu, 2005]. Les résultats expérimentaux présentés dans [Grahne and Zhu, 2005] et [Goethals and Zaki, 2003] montrent que l'algorithme FPclose est robuste et plus performant que les autres algorithmes de génération des itemsets fermés.

1.2.3.2 Algorithmes d'extraction d'itemsets maximaux

Un itemset fréquent est dit maximal si tous ses sur-ensembles sont non fréquents [Pasquier, 2000a]. Les algorithmes de génération des itemsets maximaux génèrent donc seulement les itemsets qui ne possèdent aucun sur-ensemble fréquent. L'ensemble généré par ces algorithmes est moins réduit que celui généré par les méthodes d'extraction d'itemsets fermés : seuls les itemsets les plus longs sont gardés.

Les algorithmes d'extraction des itemsets maximaux ont été développés dans le but de réduire le temps d'extraction des itemsets fréquents à partir des bases de données pour lesquels la taille des plus longs itemsets fréquents est élevée. Les itemsets maximaux peuvent être vus comme une bordure de l'ensemble des itemsets fréquents : tous les itemsets qui sont au-dessus de la bordure sont non-fréquents, et tous les itemsets qui sont au-dessous de la bordure sont fréquents [Grahne and Zhu, 2003]. La figure 6 présente un exemple d'itemsets ; seuls les itemsets $\{ABCE\}$ et $\{ACD\}$ sont maximaux. Prenons le cas de l'itemset $\{ACD\}$, la figure montre que tous ces sous-ensembles sont fréquents (comme $\{AC\}$, $\{AD\}$ et $\{CD\}$) et tous ces sur-ensembles sont non-fréquents (comme $\{ABCD\}$

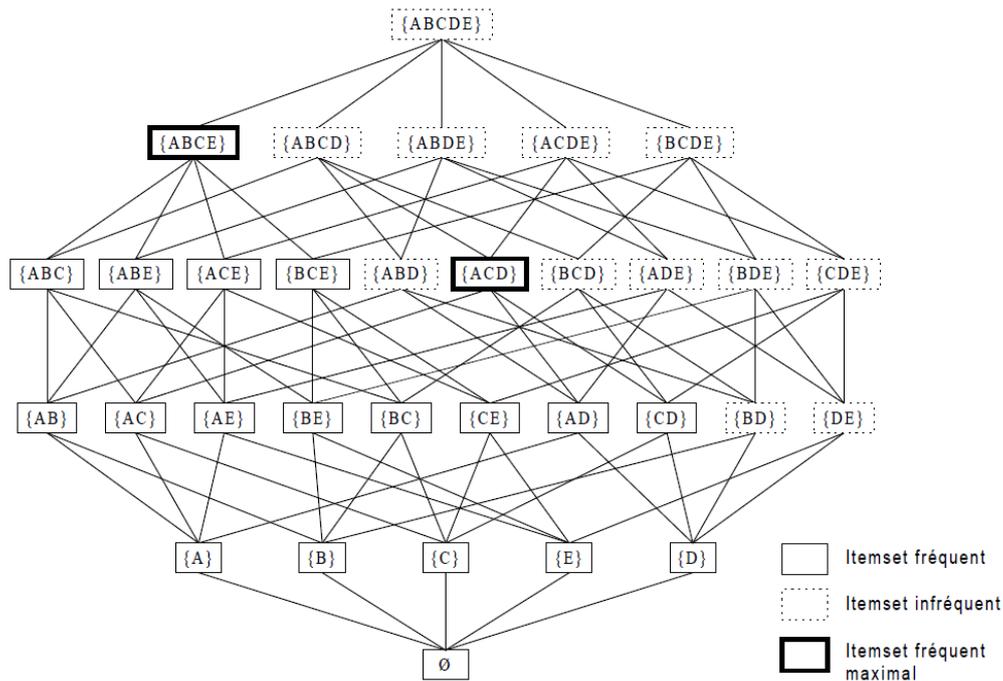


Figure 6 — Itemsets maximaux dans un jeu de données [Pasquier, 2000a]

et $\{ACDE\}$).

L'inconvénient de cette méthode est qu'elle ne permet pas de garder toute l'information fournie par l'ensemble des itemsets fréquents, comme c'était le cas des algorithmes de génération des itemsets fermés. En effet, à partir des itemsets maximaux, on peut déduire tous les itemsets fréquents, mais on ne peut pas déduire la valeur exacte de leur support.

Plusieurs travaux se sont focalisés sur les algorithmes de génération des itemsets maximaux. Mafia [Burdick et al., 2001], GenMax [Gouda and Zaki, 2001] et FPMax [Grahne and Zhu, 2003] présentent l'un de ces algorithmes. FPmax utilise une structure d'arbre, similaire à FP-Tree ; l'utilisation de cette structure de données permet de fournir des résultats efficaces et compétitifs par FPmax. Les résultats expérimentaux dans [Grahne and Zhu, 2003] montrent que FPmax est plus performant que GenMax et MA-FIA dans la plupart des cas testés.

1.3 Partie 2 : Apprentissage à base de règles floues

Les systèmes d'apprentissage à base de règles floues utilisent essentiellement la théorie des sous-ensembles flous. Un sous-ensemble flou est une généralisation du sous-ensemble classique dans le sens où on introduit un aspect graduel (nuancé) de l'appartenance d'un élément au sous-ensemble. Généralement, dans le monde réel, les sous-ensembles n'ont pas de limites précises ; par exemple on utilise les termes : petit, jeune, beau, ... qui ne peuvent pas être modélisés par des intervalles de limites précises. Cette notion de sous-ensemble flou vise à éviter le passage brusque d'une classe à une autre et permet l'appartenance

partielle à chacune avec un degré spécifique [Bouchon-Meunier, 1995]. Plus de détails sur la logique floue et les sous-ensembles flous sont fournis dans l'annexe A.

Le processus d'apprentissage et d'extraction de connaissances peut être appliqué sur différents types de données : discrètes (ordonnées ou non ordonnées), continues, textuelles, etc. Dans le cadre de ce travail, nous nous intéressons aux données continues, et nous étudions de près l'une des méthodes d'apprentissage qui traitent ce type de données : les Systèmes de Classification à Base de Règles Floues (SCBRF).

Comme toute méthode d'apprentissage, les SCBRF contiennent deux phases principales : la phase d'apprentissage et la phase de classification. Durant la phase d'apprentissage, les règles floues sont générées automatiquement à partir des données numériques. Dans la phase de classification, les règles ainsi générées seront utilisées par une méthode d'inférence floue afin d'associer une classe à une nouvelle observation. La figure 7 illustre les deux phases des SCBRF avec leurs différentes étapes.

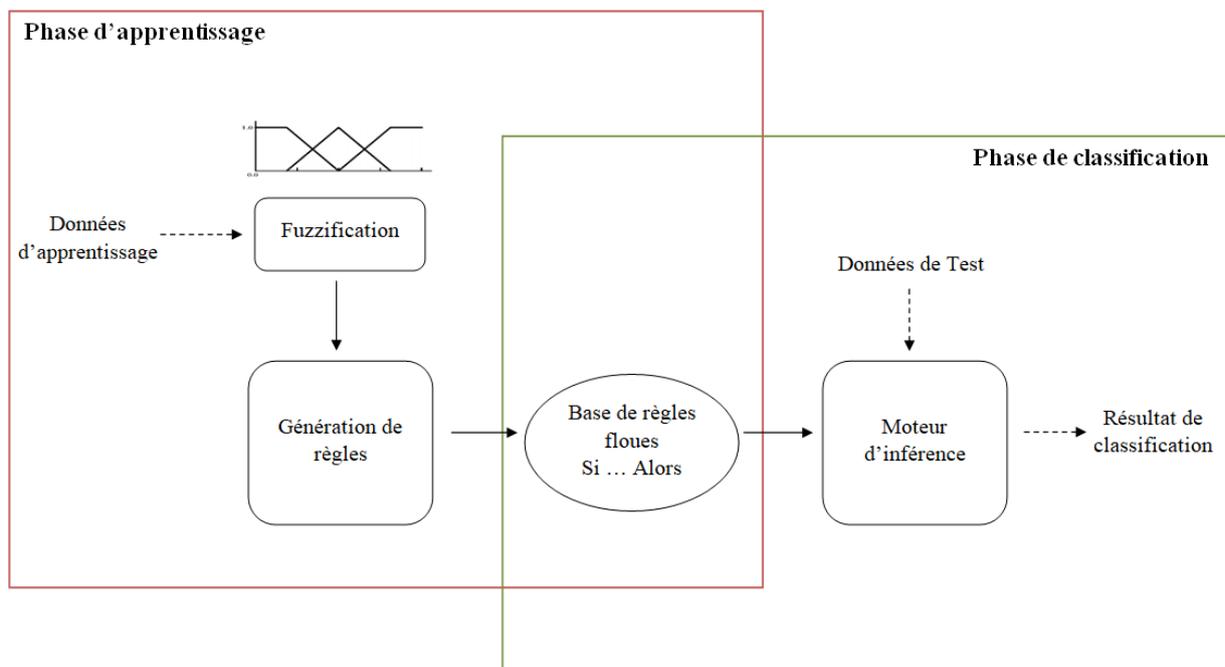


Figure 7 — Structure d'un SCBRF

1.3.1 Phase d'apprentissage

Plusieurs méthodes de génération de règles floues dans les SCBRF ont été proposées dans la littérature. L'une d'elles consiste à partitionner l'espace des entrées en des sous-espaces flous puis de construire une règle floue pour chaque sous-espace. Cette méthode, appelée méthode à base de grille floue, a été adoptée dans plusieurs travaux en utilisant différents types de grille floue [Ishibuchi et al., 1992] [Ishibuchi and Yamamoto, 2005] [Ishibuchi et al., 1994] [Nozaki et al., 1996]. Les arbres de décision ont été aussi utilisés

pour construire les règles floues à partir des données numériques [Wang et al., 2015], [Pal and Chakraborty, 2001], [Hefny et al., 2010]. Les réseaux de neurones présentent une autre alternative pour construire les règles floues [Mitra and Hayashi, 2000]. L'idée est d'utiliser la notion de neurone artificiel qui reçoit un ensemble de signaux à son entrée, qui vont être pondérés au sein du réseau de neurones [Mazieres, 2016]. Les entrées à ces réseaux correspondent aux domaines des attributs numériques et aux fonctions d'appartenance utilisées. Une méthode adaptative, appelée ANFIS, a été proposée dans le but d'adapter les fonctions d'appartenance au fur et à mesure et de construire les règles floues avec les fonctions d'appartenance les plus appropriées [Jang et al., 1997]. Nous nous intéressons dans ce travail seulement aux méthodes d'apprentissage à base de grille floue. Dans ces méthodes, deux étapes principales sont à distinguer : le partitionnement flou de l'espace des entrées en des sous-espaces flous et la construction d'une règle pour chaque sous-espace (voir Figure 8). La partition floue de tous les attributs numériques conduit à une grille floue. Cette grille représente les différents sous-espaces flous issus de l'opération de partitionnement. Chaque axe de la grille concerne un attribut X_j et est divisé en k sous-ensembles flous $A_1^k, A_2^k, \dots, A_k^k$. Dans le cas d'une partition de même taille k pour tous les attributs et dans le cas d'un problème à N attributs, le nombre de sous-espaces flous dans la grille est égal à k^N . Comme une règle floue est construite pour chaque sous-espace flou de la grille, alors le nombre de règles floues générées dans ce cas est k^N . D'un autre côté, si une partition de taille différente k_i est adoptée pour chaque attribut X_i , alors le nombre de règles floues générées est égale à $k_1 \times k_2 \times \dots \times k_N$.

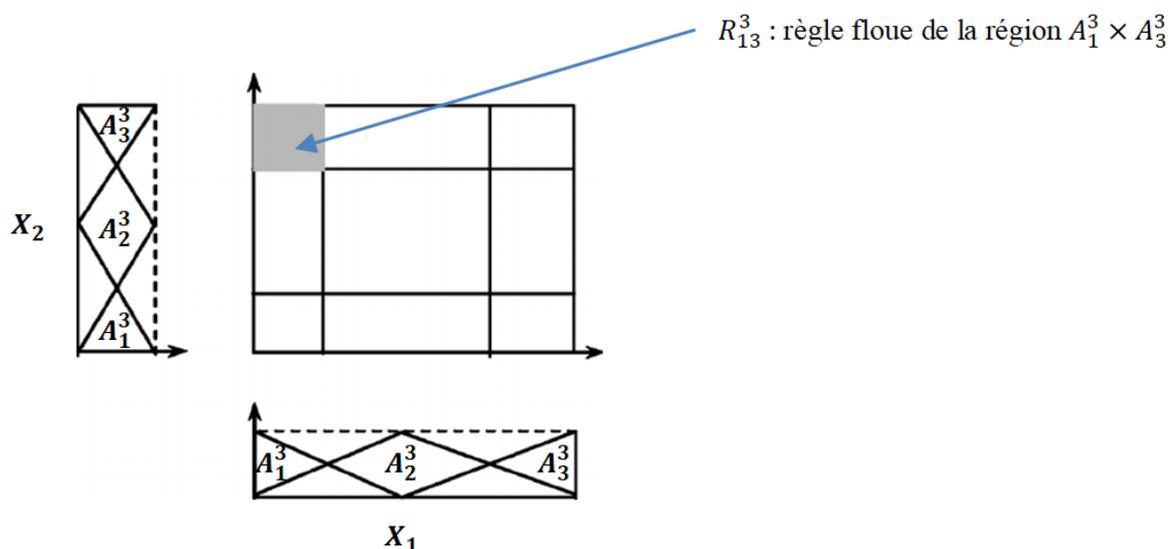


Figure 8 — Apprentissage à base de grille floue

Nous fournissons dans ce qui suit une description détaillée de l'étape de partitionnement flou puis des principales approches de construction des règles floues [Ishibuchi and Yamamoto, 2005], [Ishibuchi et al., 1992]. Ensuite, nous présentons

quelques méthodes de partitionnement flou proposées dans la littérature.

1.3.1.1 Partition floue

Afin de simplifier les notations, nous nous situons dans le cas d'un problème à deux attributs X_1 et X_2 . chacun des deux attributs X_1 et X_2 est partitionné en k sous-ensembles flous notés $A_1^k, A_2^k, \dots, A_k^k$. Un sous-ensemble flou est caractérisé par une fonction d'appartenance μ_i^k qui traduit le degré d'appartenance d'un exemple E_p au sous-ensemble A_i^k . La figure 9 illustre un exemple de partition floue de l'attribut X_1 avec $k = 5$ (k : nombre de sous-ensembles flous).

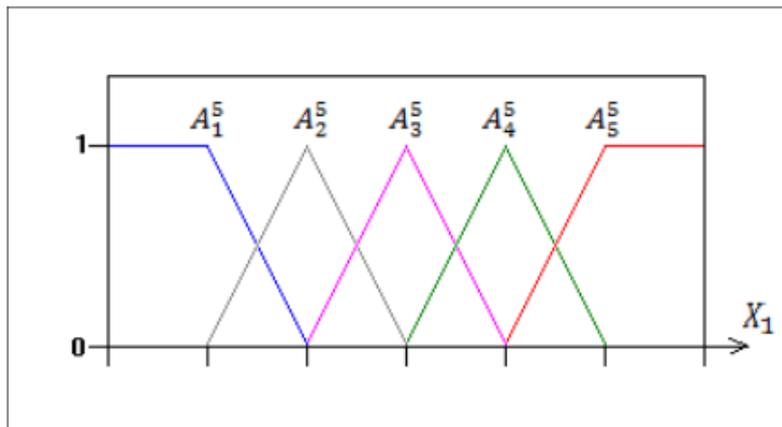


Figure 9 — Exemple de partition floue

Le partitionnement flou consiste à déterminer, pour chaque attribut, le nombre de sous-ensembles flous ainsi que la fonction d'appartenance de chaque sous-ensemble flou. Comme les fonctions d'appartenance peuvent avoir différentes formes, le nombre de sous-ensembles flous et leurs points de coupure peuvent être déterminés de plusieurs manières. Les domaines de variation des attributs continus peuvent être divisés en un nombre quelconque d'intervalles en utilisant une méthode de discrétisation spécifique. Nous distinguons différents types de discrétisation à savoir la discrétisation régulière et la discrétisation supervisée. La discrétisation supervisée prend en compte l'information relative à la classe pour trouver les intervalles appropriés alors que la discrétisation non-supervisée n'utilise pas cette information durant la phase de discrétisation. Parmi les méthodes de discrétisation non supervisée, nous trouvons la méthode d'intervalles d'amplitudes égales (EWI : Equal Width Interval). Elle divise le domaine de variation de l'attribut en un nombre prédéfini d'intervalles de largeur égale. Nous mentionnons aussi la méthode d'Intervalles de fréquences égales (EFI : Equal Frequency Interval) ; celle-ci divise le domaine de variation de l'attribut en un nombre prédéfini d'intervalles de telle sorte que chaque intervalle contienne le même nombre de valeurs. Plusieurs méthodes de discrétisation supervisée ont été aussi proposées dans la littérature comme par exemple la méthode ChiMerge [Chimerge, 1992] et MDLP [Fayyad and Irani, 1993].

Une caractéristique majeure de la partition floue est le chevauchement entre les

sous-ensembles adjacents [Ishibuchi and Yamamoto, 2002]. Une fois le domaine de variation de la variable continue est discrétisé, nous obtenons des points de coupure. Ils servent à déterminer les fonctions d'appartenance floues. La fonction d'appartenance (FA) peut avoir différentes formes telles que la forme triangulaire (Figure 9), trapézoïdale, gaussienne, etc. La forme triangulaire, utilisée dans plusieurs travaux de la littérature ([Ishibuchi et al., 1992], [Ishibuchi et al., 1999], [Dehzangi et al., 2007]), est définie comme suit :

$$\mu_i^k(x) = \max\left\{\frac{1 - |x - a_i^k|}{b^k}, 0\right\}; i = 1, 2, \dots, k \quad (3)$$

Où

$i = 1, 2, \dots, k$ représentent les points de coupure issus de l'étape de discrétisation ;

$$a_i^k = \frac{i - 1}{k - 1};$$

$$b^k = \frac{1}{k - 1}.$$

D'autres méthodes ont proposé d'ajuster les fonctions d'appartenance, avec application d'algorithmes génétiques, afin d'améliorer la performance de classification et diminuer au maximum le nombre de règles [Alcalá-Fdez et al., 2009] [Nguyen et al., 2013]. D'autre part, des travaux ont cherché à construire automatiquement les fonctions d'appartenance appropriées pour les attributs quantitatifs à partir des données d'apprentissage [Hong and Lee, 1996] [MutaherBa-Alwi, 2013]. La fonction triangulaire non symétrique est utilisée dans ces travaux ; le nombre de sous-ensembles flous ainsi que le chevauchement entre ces sous-ensembles sont déterminés automatiquement et dérivés des exemples d'apprentissage. Enfin, une méthode de partition floue supervisée a été également proposée dans [Soua et al., 2013] qui utilise la forme trapézoïdale pour définir les fonctions d'appartenance.

1.3.1.2 Construction des règles floues

Le partitionnement flou de l'espace des entrées conduit à une grille floue. Quelque soit le type de la partition, la grille floue contient les différents sous-espaces flous issus de la phase de partitionnement ; dans chacun de ces sous-espaces (case de la grille) une règle floue sera construite. La prémisse de la règle floue correspond à son domaine d'activation ; elle est formée par les sous-ensembles flous des attributs en question. La conséquence de la règle est une classe y_i parmi les C classes du problème (y_1, y_2, \dots, y_C) , accompagné d'un degré de certitude CF qui traduit la validité de la règle. Le CF indique à quel degré on est certain qu'un exemple appartenant au sous-espace flou de la règle (délimité par l'antécédent de la règle) est associé à la classe indiquée dans sa conclusion.

Prenons le cas d'un problème bi-dimensionnel où les données sont décrites par deux attributs continus X_1 et X_2 . Chaque attribut est partitionné en k sous-ensembles flous notés $A_1^k, A_2^k, \dots, A_k^k$. La règle floue correspondant à la région floue $A_i^k \times A_j^k$ et labellisée

R_{ij}^k s'écrit sous la forme suivante :

$$R_{ij}^k : \text{ si } X_1 \text{ est } A_i^k \text{ et } X_2 \text{ est } A_j^k \text{ alors } E = (X_1, X_2) \text{ appartient à } y_{ij}^k \text{ avec } CF = CF_{ij}^k \quad (4)$$

Avec :

- A_i^k et A_j^k sont les sous-ensembles flous respectifs de X_1 et X_2
- y_{ij}^k correspond à une classe parmi les C classes
- CF_{ij}^k est le degré de certitude de la règle R_{ij}^k

Notons que la notation X_1 est A_i^k signifie que la valeur de l'attribut X_1 appartient au sous-ensemble flou A_i^k .

La conclusion y_{ij}^k et le degré de certitude CF_{ij}^k sont déterminés à partir des données d'apprentissage. Dans [Ishibuchi and Yamamoto, 2005], [Ishibuchi et al., 1999], [Ishibuchi et al., 1992], la conclusion et le degré de certitude sont déterminés comme suit :

1. Calculer la somme des compatibilités des différents exemples d'apprentissage par rapport à la prémisse, pour chaque classe y_t :

$$\beta_{y_t} = \sum_{Y(E_p)=y_t} \mu_i^k(X_{1p}) \times \mu_j^k(X_{2p}); t = 1, 2, \dots, C \quad (5)$$

Avec :

- y_t : une parmi les C classes (y_1, y_2, \dots, y_C)
 - $E_p = (X_{1p}, X_{2p})$: un exemple d'apprentissage
 - $Y(E_p)$: la classe associée à l'exemple E_p
 - μ_i^k, μ_j^k : fonctions d'appartenance respectives de A_i^k et A_j^k
2. Trouver la classe y_a qui a la plus grande valeur de compatibilité avec la prémisse de la règle

$$\beta_{y_a} = \max\{\beta_{y_1}, \beta_{y_2}, \dots, \beta_{y_c}\} \quad (6)$$

Si deux ou plusieurs classes prennent la valeur maximale dans (6), alors la règle floue correspondante au sous espace flou $A_i^k \times A_j^k$ ne peut pas être générée. Sinon, la conclusion y_{ij}^k est déterminée par la formule (6) et vaut y_a .

3. Déterminer le degré de certitude CF_{ij}^k :

$$CF_{ij}^k = \frac{|\beta_{y_a} - \beta|}{\sum_{t=1}^C \beta_{y_t}} \quad (7)$$

Avec

$$\beta = \sum_{y_i \neq y_a} \frac{\beta_{y_i}}{C - 1} \quad (8)$$

Il faut noter que cette procédure ne permet pas de générer une règle floue dans le cas où aucun exemple n'existe dans le sous-espace $A_i^k \times A_j^k$. Autrement dit, si $\mu_i^k(X_{1p}) \times \mu_j^k(X_{2p}) = 0$ pour tout exemple E_p , alors la règle floue correspondant à $A_i^k \times A_j^k$ ne peut pas être construite à partir des données d'apprentissage. Ce cas apparaît surtout lorsque la partition floue est très fine (c.à.d. k est très grande).

Il est à noter que le degré de compatibilité a été calculé dans l'équation (5) en utilisant l'opérateur T-norme Produit. Cependant, d'autres T-normes peuvent être utilisées. De plus, le degré de certitude CF (calculé dans l'équation (7)) peut être déterminé de plusieurs autres manières [Ishibuchi and Yamamoto, 2005], [Fakhrmahad et al., 2007], [Nozaki et al., 1996]. Nous évoquons, dans ce qui suit, quelques autres propositions pour la génération des règles floues.

a) Autres alternatives pour le calcul du degré de compatibilité

Un sous-espace flou de la grille correspond à l'intersection de plusieurs sous-ensembles flous (un sous-ensemble flou pour chaque attribut). Dans la théorie des sous-ensembles flous, les opérateurs T-norme (introduits par [Menger, 1942], [Schweizer and Sklar, 1960]) peuvent être utilisés pour modéliser l'intersection des sous-ensembles flous (c.à.d. le connecteur logique ET entre les sous-ensembles flous). Un opérateur T-norme, noté T , représente une agrégation floue des degrés d'appartenance aux différents sous-ensembles flous (9). Si nous notons A l'antécédent d'une règle floue qui correspond à l'intersection des sous-ensembles A_1^k, \dots, A_N^k , alors le degré de compatibilité d'un exemple d'apprentissage avec la prémisse A est défini comme suit :

$$\mu_A = T(\mu_{A_1}^k, \mu_{A_2}^k, \dots, \mu_{A_N}^k) \quad (9)$$

Il existe de nombreuses définitions de l'opérateur T-norme ; les principales sont présentées dans le tableau 4.

T-norme	Notation	Définition
Minimum	T_M	$\min(x, y)$
Produit	T_P	$x \times y$
Lukasiewicz	T_L	$\max(x + y - 1, 0)$
Drastique	T_D	0 si $(x, y) \in [0, 1]^2$ et $\min(x, y)$ sinon

Tableau 4 — Principales T-normes

Dans la génération des règles floues, les auteurs dans [Ishibuchi et al., 1992] ont utilisé la T-norme Produit pour déterminer l'intersection entre les sous-ensembles flous et calculer par la suite la compatibilité des exemples d'apprentissage par rapport à la prémisse.

L'opérateur Minimum a aussi été utilisé dans [Ishibuchi et al., 1992]. Une comparaison entre les opérateurs T-norme Produit et T-norme Minimum a été faite. Cette comparaison a montré la supériorité de l'opérateur Produit par rapport à l'opérateur Minimum en analysant la performance de classification. Gonzalez et al. utilisent aussi la T-norme Minimum dans [Gonzalez and Perez, 1998]. D'autres définitions de T-normes peuvent être explorées dans [Farahbod, 2012] où une comparaison entre plusieurs mesures T-norme a été faite dans le cadre des problèmes de classification.

b) Heuristiques de calcul du degré de certitude

Le degré de certitude CF a un effet considérable sur le résultat de classification des systèmes à base de règles floues [Ishibuchi and Nakashima, 2001]. Ce degré a été déterminé selon les formules (7) et (8) dans [Ishibuchi et al., 1992]. Dans [Ishibuchi and Yamamoto, 2005], les auteurs présentent différentes heuristiques du calcul du degré de certitude basées sur les mesures Support et Confiance. De plus, dans [Nozaki et al., 1996], une méthode d'ajustement du degré de certitude a été proposée. Nous présentons ici une description de ces travaux.

- Méthodes basées sur la mesure Confiance

Nous nous situons dans le cas d'un problème à N dimensions, contenant m exemples d'apprentissage ($E_p = (X_{p1}, X_{p2}, \dots, X_{pN}); p = 1, 2, \dots, m$). Chaque attribut est partitionné en k sous-ensembles flous $A_1^k, A_2^k, \dots, A_k^k$.

Les auteurs dans [Ishibuchi and Yamamoto, 2005] ont proposé des méthodes heuristiques pour le calcul du degré de certitude, basées sur les notions de support et confiance issues du Data Mining [Agrawal and Srikant, 1994]. Comme les mesures support et confiance sont généralement utilisés pour évaluer les Règles d'Association, les auteurs ont considéré qu'une règle floue R_{ij}^k (de la forme (4)) peut être vue comme une règle d'association $A_q \rightarrow y_q$ où A_q est la prémisse de la règle floue qui correspond à l'intersection des sous-ensembles flous ($A_{q1}^k, A_{q2}^k, \dots, A_{qN}^k$) et y_q est sa conclusion. La version floue des deux mesures support et confiance [Hong et al., 2001] a alors été utilisée.

- La confiance d'une règle floue $A_q \rightarrow y_q$ représente la validité de la règle et est définie comme suit :

$$c(A_q \rightarrow y_q) = \frac{\sum_{Y(E_p)=y_q} \mu_{A_q}(E_p)}{\sum_{p=1}^m \mu_{A_q}(E_p)} \quad (10)$$

- Le support d'une règle floue est défini comme suit :

$$s(A_q \rightarrow y_q) = \frac{\sum_{Y(E_p)=y_q} \mu_{A_q}(E_p)}{m} \quad (11)$$

où $\mu_{A_q}(E_p)$ représente le degré de compatibilité de l'exemple E_p avec l'antécédent A_q ; il est calculé en utilisant l'opérateur Produit tel que défini dans [Ishibuchi et al., 1992] : $\mu_{A_q}(X_p) = \mu_{A_{q1}^k}(X_{p1}) \times \mu_{A_{q2}^k}(X_{p2}) \times \dots \times \mu_{A_{qN}^k}(X_{pN})$.

Quatre heuristiques pour la définition du degré de certitude ont été présentées dans [Ishibuchi and Yamamoto, 2005] :

1. Premièrement, la confiance de la règle floue $A_q \rightarrow y_q$ est utilisée directement en tant que degré de certitude CF de la règle.

$$CF = c(A_q \rightarrow y_q) \quad (12)$$

Cette définition a été aussi utilisée dans [Cordón et al., 1999].

2. Une autre définition a été utilisée dans [Ishibuchi and Yamamoto, 2005] et [Ishibuchi et al., 2001b]

$$CF = c(A_q \rightarrow y_q) - c_{ave} \quad (13)$$

Où c_{ave} est la confiance moyenne des autres règles floues avec le même antécédent A_q mais des conséquences différentes de y_q (classes différentes de y_q)

$$c_{ave} = \frac{1}{C-1} \sum_{h=1 \text{ et } h \neq y_q}^C c(A_q \rightarrow y_h) \quad (14)$$

Des simulations expérimentales dans [Ishibuchi et al., 2001b] ont démontré que l'utilisation des définitions (13) et (14) pour le calcul du degré de certitude, donne de meilleurs résultats que la définition (12), lorsqu'on considère la performance de classification comme critère d'évaluation.

3. Une nouvelle définition a été proposée dans [Ishibuchi and Yamamoto, 2005]. Elle est basée sur la différence entre la plus grande et la deuxième plus grande confiance :

$$CF = c(A_q \rightarrow y_q) - c_{2nd} \quad (15)$$

Avec c_{2nd} est la deuxième plus grande confiance parmi les confiances des règles ayant l'antécédent A_q et une classe différente de la classe y_q .

$$c_{2nd} = \max\{c(A_q \rightarrow y_h) / h = 1, 2, \dots, C; y_h \neq y_q\} \quad (16)$$

4. Enfin, [Ishibuchi and Yamamoto, 2005] propose une dernière définition :

$$CF = c(A_q \rightarrow y_q) - c_{sum} \quad (17)$$

Avec c_{sum} la somme des confiances des règles floues ayant le même antécédent A_q mais des conséquences différentes de y_q .

$$c_{sum} = \sum_{h=1 \text{ et } h \neq y_q}^C c(A_q \rightarrow y_h) \quad (18)$$

Il est à noter que la troisième et la quatrième définitions (15) et (17) sont les mêmes que la deuxième définition (13) lorsque le problème de classification implique seulement deux classes ($C=2$). Ceci est dû au fait que $c_{ave} = c_{2nd} = c_{sum}$ dans le cas où $C = 2$ dans les formules (14), (16) et (18). Notons également que la valeur du CF est toujours positive dans les trois premières définitions (12), (13) et (15) ; par contre, elle peut être négative lorsqu'on utilise la dernière définition (17). Les règles floues avec un degré de certitude négatif ne sont pas prises en compte dans le système de classification.

Pour comparer l'effet des quatre heuristiques sur la performance de classification, les auteurs dans [Ishibuchi and Yamamoto, 2005] utilisent deux bases de données de l'UCI Repository¹. Ces deux bases, Wine et Glass, sont de caractéristiques différentes. La première implique un petit nombre de classes (3 classes) alors que la deuxième contient un nombre important de classes (7 classes). Les expérimentations montrent que, dans le cas de la première base où le nombre de classes est petit, le choix de la spécification du degré de certitude n'a pas grand effet sur la performance de classification : la différence, dans le taux de classification, entre les quatre définitions n'est pas grande. Par contre, dans le cas de la deuxième base qui implique plusieurs classes, la quatrième définition (17) améliore notablement la performance de classification. Les auteurs dans [Ishibuchi and Yamamoto, 2005] ont expliqué cette supériorité de la quatrième définition par le fait que les règles avec un CF négatif sont éliminées par cette définition. Une règle avec un degré négatif de la quatrième définition (17) coïncide avec une règle avec un degré inférieur à 0.5 de la première définition (12). Lorsque ces règles sont éliminées par la définition (17), elles sont utilisées dans le cas de la définition (12) malgré que le CF est inférieur.

- Méthode de classification adaptative

Les méthodes précédemment citées définissent des heuristiques du calcul du degré de certitude ; la méthode de [Nozaki et al., 1996] quant à elle ajuste le degré de certitude de chaque règle après avoir analysé son degré de classification. Cette méthode consiste en une procédure d'apprentissage basée sur une correction d'erreur : si un exemple est correctement classé par une règle floue, alors le degré de certitude de cette règle est augmenté ; si un exemple est mal-classé par une règle, alors son degré de certitude est diminué. La procédure proposée dans [Nozaki et al., 1996] est décrite comme suit : d'abord, les valeurs initiales des degrés de certitude de chaque règle floue sont déterminées par la procédure heuristique proposée dans [Ishibuchi et al., 1992] (équation (7)). Puis, ces CF sont ajustés en prenant les exemples d'apprentissage un par un. Pour chaque exemple d'apprentissage E_p , si la règle R_{ij}^k a conduit à sa classification, alors son degré de certitude est ajusté de la façon suivante :

1. si E_p est correctement classifié par R_{ij}^k , alors CF est renforcé :

1. <https://archive.ics.uci.edu/ml/datasets.html>

$$CF_{ij}^k = CF_{ij}^k + \eta_1 \times (1 - CF_{ij}^k) \quad (19)$$

Où η_1 est une constante d'apprentissage positive servant à augmenter le degré de certitude.

2. si E_p est mal classé par R_{ij}^k alors CF est affaibli

$$CF_{ij}^k = CF_{ij}^k - \eta_2 \times CF_{ij}^k \quad (20)$$

Où η_2 est une constante d'apprentissage positive qui sert à diminuer le CF .

Les auteurs dans [Nozaki et al., 1996] ont spécifié que la constante η_1 doit être beaucoup plus petite que η_2 : $0 < \eta_1 \ll \eta_2 < 1$. Ceci parce que, généralement, le nombre d'exemples correctement classifiés est beaucoup plus grand que celui des exemples mal classifiés, et donc le degré de certitude de chaque règle tend à être augmenté jusqu'à sa limite supérieure ($CF_{ij}^k \approx 1$) si on attribue la même valeur aux constantes η_1 et η_2 .

L'application de cette méthode sur une base de données réelle de l'UCI (la base de données Iris) a permis d'examiner la capacité d'apprentissage (en analysant le pourcentage de classification des données d'apprentissage) ainsi que la capacité de généralisation (en analysant le pourcentage de classification des données de test) du système construit. La méthode adaptative a été comparée avec la méthode de la grille floue simple [Ishibuchi et al., 1992]. La simulation sur les données d'apprentissage a montré que la capacité d'apprentissage de la méthode adaptative est plus importante que celle de la méthode de la grille floue simple. D'autre part, la simulation sur les données de test a montré que la capacité de généralisation du système de classification a été amélioré par la méthode adaptative dans le cas où la partition floue de l'espace d'entrée est grossière ($k = 2, 3, 4$). Par contre, lorsque la partition floue est fine ($k = 5, 6$), la capacité de généralisation de la méthode adaptative est légèrement moins bonne que celle de la méthode de grille floue simple. Cette détérioration est due à un sur-ajustement aux données d'apprentissage, c.à.d un sur-apprentissage.

1.3.1.3 Choix de la partition floue

Nous avons décrit dans les deux paragraphes précédents les étapes principales de la phase d'apprentissage : la partition floue de l'espace des entrées en des sous-espaces flous et la construction d'une règle floue pour chaque sous-espace. Nous avons mentionné auparavant que plusieurs méthodes de partition floue ont été proposées dans la littérature et peuvent être appliquées dans ce contexte. Nous décrivons dans cette partie quelques méthodes de partition floue parmi celles proposées dans la littérature. Les méthodes de discrétisation, de calcul du chevauchement entre les sous-ensembles flous et de la détermination des fonctions d'appartenance varient d'un travail à un autre.

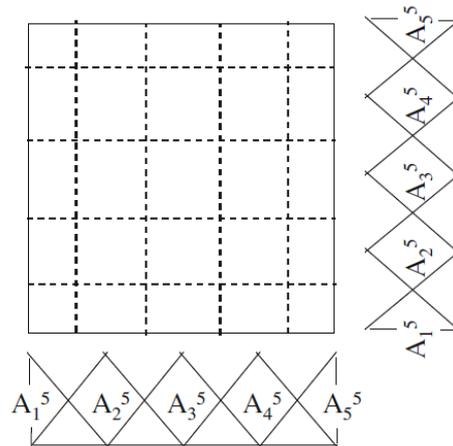


Figure 10 — Exemple de grille floue simple

Partition floue simple

La partition floue simple consiste à partitionner les attributs numériques en un même nombre d'intervalle k ; les intervalles sont de même taille [Ishibuchi et al., 1992]. Cela signifie que la valeur de k est la même pour tous les attributs. La méthode de discrétisation utilisée est donc la discrétisation régulière EW (Equal Width Interval).

Suite à l'opération de partitionnement, une grille floue simple est obtenue. Chaque axe de la grille est divisé en k sous-ensembles flous $A_1^k, A_2^k, \dots, A_k^k$. La fonction d'appartenance triangulaire symétrique est utilisée pour définir les sous-ensembles flous A_i^k . Le nombre de sous-espaces flous dans une grille à deux dimensions est égal à k^2 . Dans le cas général, pour un problème à N attributs et k partitions pour chaque attribut, on obtient une grille floue de k^N sous-espaces flous. Un exemple de grille floue simple est illustré par la figure 10. Cette grille concerne un problème à deux attributs ($N = 2$) où chaque attribut est partitionné en 5 sous-ensembles flous ($k = 5$). Le nombre de sous-espaces flous de cette grille est égale à $5^2 = 25$. Une règle floue sera ensuite construite pour chaque sous-espace flou.

Partition floue multiple

La performance de la méthode de la grille floue simple dépend du choix de la valeur de k : si cette valeur est assez petite (la partition floue est assez grossière), on risque de générer des règles de faible capacité de classification ; par contre, dans le cas où k est assez grande (partition floue très fine), certaines règles ne peuvent pas être générées à cause du manque d'exemples dans le sous-espace correspondant. C'est pour contourner ce problème qu'Ishibuchi et al. ont proposé de considérer des partitions fines et grossières simultanément [Ishibuchi et al., 1992]. La méthode consiste à superposer plusieurs grilles floues où chaque grille correspond à une partition floue (Figure 11). Les règles générées par cette méthode sont appelées "Règles floues distribuées". Dans le cas de N attributs et k sous-ensembles flous pour chaque attribut, cette méthode de grille floue multiple considère $(k - 1)$ partitions floues en même temps. Par exemple, si $k = 3$, les deux grilles

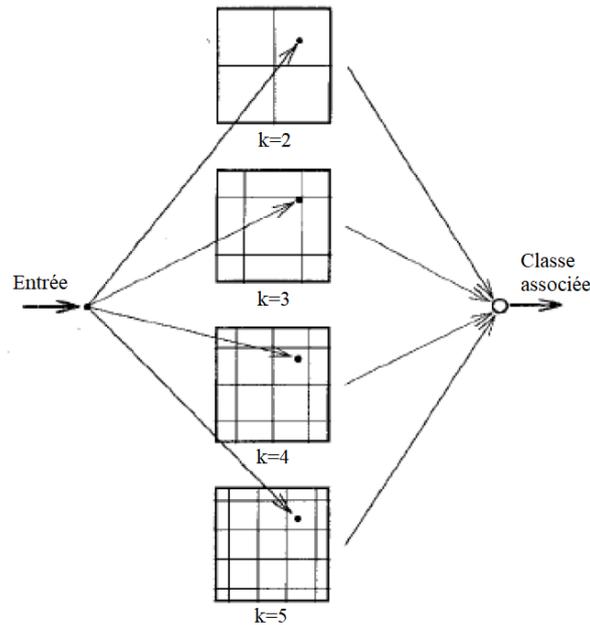


Figure 11 — Grille floue multiple

floues qui correspondent à $k = 2$ et $k = 3$ sont considérées pour générer la base de règles. Le nombre de règles floues obtenues dans ce cas est $2^N + 3^N$.

Partition floue efficace

L'utilisation de la grille floue multiple [Ishibuchi et al., 1992] résout le problème des partitions fines et grossières, mais en contrepartie, le nombre de règles générées ($2^N + 3^N + \dots + k^N$) augmente considérablement dans le cas d'un espace de dimension élevée (lorsque la valeur de N est grande). Dans [Ishibuchi et al., 1993], une nouvelle méthode de partitionnement floue a été proposée ; cette méthode consiste à partitionner l'espace des entrées en des sous-espaces de tailles différentes : des zones seront divisées en de fins sous-espaces et d'autres en de larges sous-espaces flous (voir Figure 12). La taille de chaque sous-espace dépend de la configuration des données d'apprentissage.

La procédure de partitionnement consiste à diviser l'espace des entrées en 2^N sous-espaces flous ; puis de sélectionner un de ces sous-espaces pour le subdiviser encore en 2^N sous-espaces. Cette procédure est répétée jusqu'à ce qu'une condition d'arrêt soit satisfaite. Pour sélectionner les sous-espaces flous à subdiviser, trois critères ont été proposés :

- Le premier critère consiste au degré de certitude CF de la règle correspondante au sous-espace flou. En effet, un CF inférieur à 1 signifie que le sous-espace flou inclut des exemples qui appartiennent à des classes différentes. Donc, le degré de certitude peut être considéré comme le degré d'homogénéité des données par rapport à la classe. En utilisant ce critère, durant chaque itération, le sous-espace flou qui possède le CF le plus petit est sélectionné pour être subdiviser.
- Le deuxième critère correspond au nombre d'observations qui ont été malclassifiées

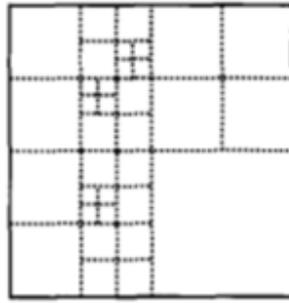


Figure 12 — Grille floue avec partition efficace

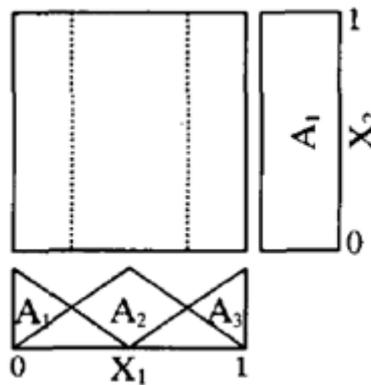


Figure 13 — Grille floue rectangulaire

par la règle floue. L'objectif de ce choix est de pouvoir classer correctement toutes les observations. Donc, le sous-espace dont la règle correspondante se trompe sur la classe du plus grand nombre d'observations est à subdiviser.

- Le dernier critère est similaire au deuxième. On cherche aussi à réduire le nombre de mauvaises classifications en choisissant comme critère le pourcentage des exemples malclassifiés.

L'inconvénient de cette méthode est qu'on risque d'avoir un problème de sur-apprentissage : les règles de classification peuvent trop coller aux données d'apprentissage et perdre par la suite leur pouvoir de prédiction sur de nouvelles observations.

Partition floue rectangulaire

Dans les méthodes précédentes, toutes les règles floues ont été générées à partir de sous-espaces flous carrés. Dans ces méthodes, la même partition floue (la même valeur de k) est utilisée pour chaque axe (chaque attribut), ce qui conduit à avoir la forme carrée des sous-espaces de la grille. Une autre méthode a été proposée dans [Ishibuchi et al., 1994], elle utilise des partitions floues différentes pour chaque axe (valeurs différentes de k) et construit par la suite des sous-espaces flous de forme rectangulaire. La figure 13 montre un exemple d'une grille construite par cette méthode.

Partition floue supervisée

Dans les méthodes citées auparavant, le nombre de sous-ensembles flous ainsi que les fonctions d'appartenance sont fixés a priori. Une discrétisation régulière a permis de partitionner le domaine de chaque attribut. Cette discrétisation, bien que simple et facile à mettre en oeuvre, ne permet pas toujours de déterminer le partitionnement approprié de l'espace des données d'entrée. De plus, la forme triangulaire symétrique, souvent utilisée pour représenter la fonction d'appartenance, n'est pas toujours justifiée.

Dans ce paragraphe, nous présentons une méthode de partition floue supervisée, proposée dans [Soua et al., 2013], qui consiste à concevoir une méthode de génération automatique de la partition floue sans fixer aucun paramètre auparavant.

Cette méthode est basée sur une discrétisation supervisée suivie par une construction automatique des fonctions d'appartenance. La discrétisation supervisée prend en compte l'information relative à la classe pour trouver les intervalles appropriés des attributs numériques. Le choix de ce type de discrétisation peut être justifié par le fait que la performance des modèles de classification dépend fortement des points de coupure construits par la discrétisation. La capacité de généralisation des ces modèles peut être significativement améliorée si les meilleurs points de coupure sont choisis [Soua et al., 2013].

Deux méthodes de discrétisation supervisées ont été utilisées dans le travail de [Soua et al., 2013] : la méthode ChiMerge [Chimerge, 1992] basée sur le critère Khi-2, et la méthode MDLP basée sur une heuristique de minimisation de l'entropie, avec ces deux critères : le critère de Fayad et Irani [Fayyad and Irani, 1993] et le critère de Kononenko [Kononenko et al., 1996].

Après le partitionnement du domaine de variation d'un attribut en des sous-intervalles (via la méthode de discrétisation supervisée choisie), les fonctions d'appartenance sont construites automatiquement. Pour cela, les auteurs dans [Soua et al., 2013] ont fixé trois conditions à suivre :

- les fonctions d'appartenances FA sont linéaires
- la somme des FA voisines est égale à 1 : si un exemple x appartient à deux sous-ensembles flous adjacents A_j et A_{j+1} , alors la somme des degrés d'appartenance aux deux sous-ensembles est égale à 1 ($\mu_j^k(x) + \mu_{j+1}^k(x) = 1$)
- les points de chevauchement entre deux FA voisines coïncident avec les points de coupure entre les intervalles obtenus par la discrétisation supervisée.

Pour définir les FA, la forme trapézoïdale a été utilisée. Pour chaque fonction d'appartenance, les points suivants sont à déterminer (voir Figure 14) :

- B_i et C_i : les points de départ et final pour la plage où FA est égale à 1
- A_i et D_i : les points de départ et final pour la plage où FA est supérieure à 1

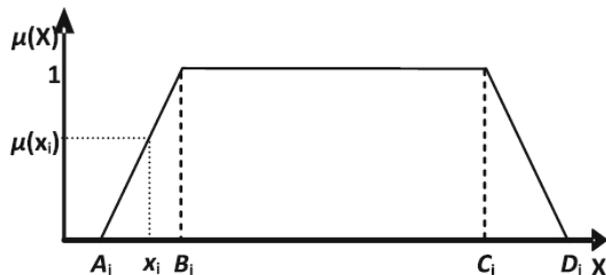


Figure 14 — Représentation d'une fonction d'appartenance trapézoïdale

Rappelons qu'en utilisant une discrétisation supervisée, un nombre différent d'intervalles est déterminé pour chaque attribut. Si ce nombre est égale à k pour un attribut X , alors $k - 1$ points de coupure sont obtenus. Nous notons PC_i le point de coupure entre les intervalles Int_i et Int_{i+1} . Dans ce cas, les points de départ et finaux de chaque FA sont déterminés comme suit :

$$A_1 = B_1 = X_{min} \quad (21)$$

$$C_k = D_k = X_{max} \quad (22)$$

$$A_i = C_{i-1} \text{ avec } i > 1 \quad (23)$$

$$B_i = D_{i-1} \text{ avec } i > 1 \quad (24)$$

$$C_i = PC_i - overlap \text{ avec } i < k \quad (25)$$

$$D_i = PC_i + overlap \text{ avec } i < k \quad (26)$$

Le paramètre *overlap* représente le degré de chevauchement des FA de deux termes adjacents. Ce paramètre est fixé à 0.2 dans [Sousa et al., 2013].

La figure 15 représente un exemple de partition floue pour un attribut X obtenue par la méthode de partition floue supervisée ainsi décrite. Les points de coupure et les fonctions d'appartenance sont illustrés dans cette figure. En outre, la figure 16 représente un exemple de grille floue obtenue par cette méthode pour un problème à deux attributs ; le premier attribut est partitionné en 5 sous-ensembles flous et le deuxième en 3 sous-ensembles flous.

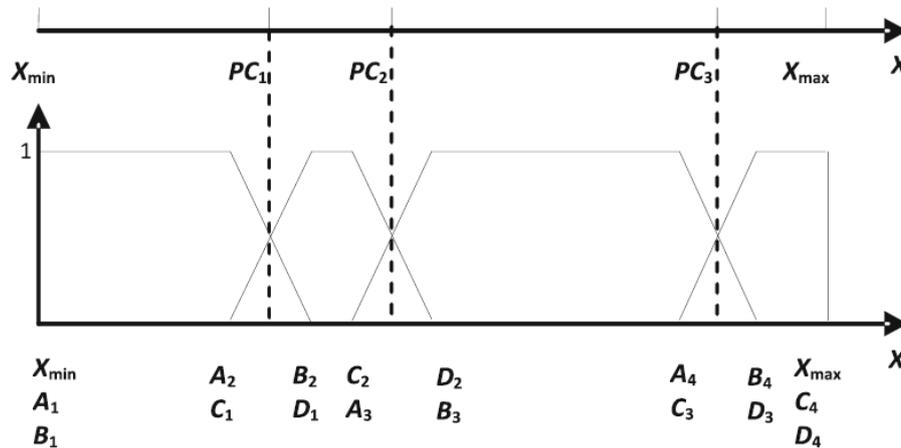


Figure 15 — Construction d'une partition floue supervisée

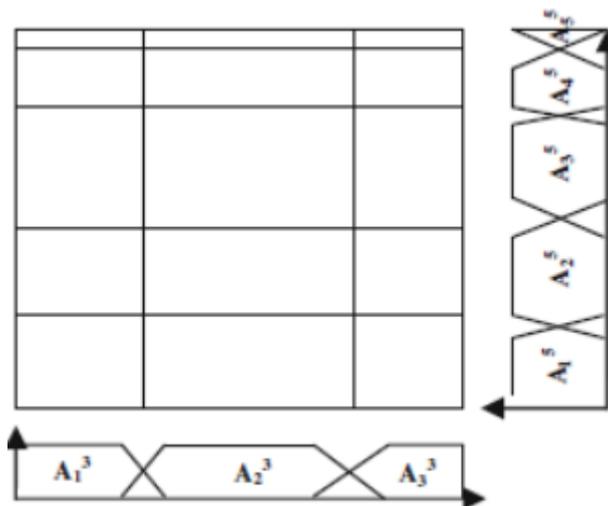


Figure 16 — Grille floue obtenue par la méthode de la partition floue supervisée

1.3.2 Phase de classification

On s'intéresse dans cette section à la phase d'inférence floue, appelée aussi phase de raisonnement ou de classification floue. Celle-ci décide, à partir de la base de règles fournie ou générée, la classe associée à un individu de classe inconnue. Généralement, la méthode de classification utilisée consiste à associer au nouvel objet la classe de la règle gagnante, celle qui concorde au mieux avec l'objet à classer [Ishibuchi et al., 1992] [Ishibuchi and Yamamoto, 2005]. Cette méthode est connue sous le nom de "Seule règle gagnante" (Single Winner Rule). D'autres travaux ont utilisé une autre méthode de classification, appelée "méthode de vote pondéré" (Weighted Vote Method), qui consiste à prendre en compte le vote de plusieurs règles qui concordent avec l'exemple à classer [Ishibuchi et al., 1999] [Cordón et al., 1999]. On décrit dans ce qui suit quelques travaux des deux types de classification.

1.3.2.1 Méthode de la seule règle gagnante

A l'issue de la phase d'apprentissage, une base de règles floues, notée S_R , est obtenue. Nous prenons toujours le cas d'un problème à deux dimensions X_1 et X_2 , et nous considérons un nouvel objet à classer $E' = (X'_1, X'_2)$. La méthode de la seule règle gagnante consiste à :

1. Calculer α_{y_t} pour chaque classe ; $t = 1, 2, \dots, C$ par :

$$\alpha_{y_t} = \max(\mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k / y_{ij}^k = y_t; R_{ij}^k \in S_R) \quad (27)$$

2. Trouver la classe y_a qui maximise α_{y_t} , soit :

$$\alpha_{y_a} = \max(\alpha_{y_1}, \alpha_{y_2}, \dots, \alpha_{y_C}) \quad (28)$$

Le produit $\mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k$ est calculé pour chaque règle R_{ij}^k . Ce produit reflète le degré d'association de la règle avec l'objet à classer. Le résultat de l'inférence floue est alors la conséquence de la règle gagnante, celle qui possède le produit $\mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k$ maximum. La classification de l'objet E' est rejetée dans deux cas : le premier cas correspond au fait que deux ou plusieurs classes atteignent ce maximum, le deuxième cas est lorsqu'il n'existe aucune règle floue dont $\mu_i^k(X'_1) \times \mu_j^k(X'_2) > 0$ et $CF_{ij}^k > 0$, c.à.d. aucune règle floue n'est compatible avec l'exemple à classer.

1.3.2.2 Méthode de vote pondéré

Dans [Ishibuchi et al., 1992], le raisonnement flou a été basé sur le choix d'une seule règle floue qui possède le meilleur degré d'association avec l'exemple à classer. L'exemple est donc associé à la classe qui apparaît dans la conséquence de cette règle. Une autre méthode de raisonnement flou a été proposée dans [Ishibuchi et al., 1999] ; cette méthode est basée sur le vote de toutes les règles floues qui sont compatibles avec l'exemple à classer. Chaque règle floue R_{ij}^k vote pour sa classe y_{ij}^k avec un poids. Ce poids correspond au degré d'association de la règle avec l'objet à classer. Ainsi, pour chaque classe y_t , la somme des différents poids des règles qui votent pour cette classe est comptée afin d'obtenir le degré α_{y_t} .

En considérant la base de règles floues S_R , et un nouvel objet $E' = (X'_1, X'_2)$, la méthode de vote de règles multiples se déroule selon les étapes suivantes [Ishibuchi et al., 1999] :

1. Calculer le degré d'association α_{y_t} de l'exemple E' avec chaque classe y_t ; $t = 1, 2, \dots, C$:

$$\alpha_{y_t} = \sum_{R_{ij}^k \in S_R \text{ et } y_{ij}^k = y_t} \mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k \quad (29)$$

2. Trouver la classe y_a qui maximise α_{y_t} , soit :

$$\alpha_{y_a} = \max(\alpha_{y_1}, \alpha_{y_2}, \dots, \alpha_{y_C}) \quad (30)$$

Quand une règle floue avec un petit poids, $\mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k$ n'a aucun effet sur le résultat de classification dans le cas de la méthode de raisonnement basée sur une seule règle gagnante ; une telle règle possède un effet qui, bien qu'il soit petit, participe à la méthode de raisonnement floue basée sur le vote multiple.

Comme dans [Ishibuchi et al., 1999], les auteurs dans [Cordón et al., 1999] ont proposé une méthode de raisonnement flou qui prend en compte le vote de toutes les règles floues qui sont compatibles avec l'exemple à classer. Dans ce dernier travail, les auteurs considèrent aussi que le choix de la règle ayant le degré d'association maximal, seule, peut détériorer la performance de classification puisqu'elle exclut la contribution de toutes les autres règles qui peuvent, ensemble, avoir un degré d'association agrégé supérieur.

Les deux méthodes [Ishibuchi et al., 1999] et [Cordón et al., 1999] utilise la méthode de vote pondéré. Cependant, quand la fonction d'agrégation dans [Ishibuchi et al., 1999] correspond à la sommation des différents poids, les auteurs dans [Cordón et al., 1999] ont proposé six autres fonctions d'agrégation et ont analysé leurs effets sur la capacité de généralisation du système.

La méthode de raisonnement de [Cordón et al., 1999] comprend les étapes suivantes :

Nous prenons toujours le cas d'un problème bidimensionnel. Nous notons $B_{R_{ij}^k}$ le degré de compatibilité de l'exemple E' avec la règle R_{ij}^k :

$$B_{R_{ij}^k} = \mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k \quad (31)$$

La T-norme Produit est utilisée dans cette formule (31) ; d'autres T-normes peuvent être utilisées.

Le degré d'association α_{y_t} de l'exemple E' avec chaque classe y_t est noté :

$$\alpha_{y_t} = f(B_{R_{ij}^k} / R_{ij}^k \in S_R \text{ et } y_{ij}^k = y_t) \quad (32)$$

Où f est la fonction d'agrégation utilisée. Si f est égale à l'opérateur maximum (équation 27), alors cette méthode rejoint la méthode de raisonnement de la seule règle gagnante [Ishibuchi et al., 1992] ; si f est égale à l'opérateur Somme (équation 29) alors la méthode de raisonnement correspond à la méthode de vote multiple de [Ishibuchi et al., 1999]. Différentes autres formes de f ont été proposées dans [Cordón et al., 1999].

Soient (B_1, \dots, B_s) les degrés d'association positifs de l'exemple E' avec les règles R_{ij}^k de classe y_t . s est donc le nombre de règles de la classe y_t qui sont compatibles avec l'exemple E' . Nous présentons ici deux fonctions d'agrégation proposées dans [Cordón et al., 1999]. Le lecteur intéressé par les autres fonctions d'agrégation peut trouver plus de détails dans [Cordón et al., 1999].

1. L'addition normalisée :

$$f_1(B_1, B_2, \dots, B_s) = \frac{\sum_{i=1}^s B_i}{f_{1max}} \quad (33)$$

Où $f_{1max} = \max_{h=1, \dots, C} \sum_{i=1}^{S_h} B_i$; S_h est le nombre de règles ayant la classe y_h comme conclusion; (B_1, \dots, B_{S_h}) sont les degrés d'association de l'exemple E' avec les règles de la classe y_h .

La fonction normalisée revient à comptabiliser la somme des degrés B_i des règles ayant la classe y_t en partie conclusion et à la diviser par le maximum des sommes des degrés des règles pour toutes les classes.

2. La moyenne arithmétique :

$$f_2(B_1, B_2, \dots, B_s) = \frac{\sum_{i=1}^s B_i}{s} \quad (34)$$

1.3.3 SCBRF et les bases à haute dimension

Généralement, le nombre de règles floues générées dans les SCBRF augmente exponentiellement en fonction du nombre d'attributs, spécialement dans le cas des systèmes d'apprentissage à base de grille floue. Prenons le cas de la grille floue simple comme choix de partition floue (le cas le plus simple), et supposons que chaque attribut est partitionné en k sous-ensembles flous, le nombre de règles floues générées dans ce cas est égal à k^N (où N est le nombre d'attributs du problème considéré). Un nombre élevé d'attributs N conduit donc à une explosion du nombre de règles générées dans ces systèmes. Par exemple, dans le cas d'une base de données ayant 13 attributs, comme la base Wine (la description des bases testées est fournie dans le chapitre 5), où chaque attribut est partitionné en 5 sous-ensembles flous, il existe plus d'un billion de régions floues dans la grille simple et donc plus d'un billion de règles floues. Même si un autre type de partition floue est utilisé (partition floue rectangulaire, efficace, supervisée, ...), le nombre de règles reste important puisqu'il dépend d'abord du nombre d'attributs. Par exemple, dans le cas d'une partition floue supervisée, le nombre de sous-ensembles flous n'est pas le même pour tous les attributs. Soit k_i le nombre de sous-ensembles flous pour le i^{me} attribut ($i = 1, \dots, N$), le nombre de règles floues générées est dans ce cas : $k_1 \times k_2 \times \dots \times k_N$.

L'explosion du nombre de règles floues peut dégrader la compréhensibilité du modèle de classification. En effet, l'avantage des règles floues consiste à utiliser des termes linguistiques qui sont interprétables et compréhensibles par l'être humain. Cependant, une règle qui contient une vingtaine de conditions, même linguistiques, dans sa prémisse n'est plus compréhensible. De ce fait, le nombre d'attributs figurant dans la même prémisse doit être diminué. D'autre part, la génération d'un nombre important de règles floues peut affecter le temps de réponse nécessaire aussi bien à la phase d'apprentissage qu'à la phase de classification. Le travail expérimental dans [Sousa et al., 2013] a montré que la génération des règles floues se bloque pour les bases de données de dimensions supérieures ou

égales à 13 attributs, et ce dans les systèmes de classification à base de grille floue simple. En fait, le problème de génération de règles apparaît dès que le nombre d'attributs dépasse une dizaine d'attributs. Nous nous intéressons donc dans cette thèse aux travaux qui cherchent à diminuer la complexité des SCBRF sur les bases à haute dimension. Nous traitons spécifiquement le cas des bases de données qui comportent une dizaine voire une centaine d'attributs. Manipuler ces bases de données par les systèmes de classification à base de grilles floues présente un enjeu dans plusieurs travaux de recherche.

Afin de réduire le nombre de règles générées dans les SCBRF et afin de réduire la complexité de ces systèmes, plusieurs solutions ont été proposées dans la littérature. Nous présentons dans ce qui suit certaines des solutions proposées.

1.3.3.1 Réduction de dimension

Dans plusieurs domaines scientifiques, le nombre d'attributs dépasse largement le nombre d'observations, tel est le cas des problèmes de biopuces où les variables (les gènes) sont plus nombreuses que les observations [Lazar et al., 2012, Ding and Peng, 2005]. Ces attributs ne sont pas nécessairement tous utiles ; par contre, certains attributs peuvent provoquer du bruit ou fournir des informations redondantes. Éliminer les attributs redondants ou inutiles s'avère donc nécessaire pour ce cas de problème. Une des solutions les plus intuitives à ce problème est la réduction de dimension. Cela consiste à appliquer soit une sélection des attributs les plus significatifs, soit une extraction et transformation d'attributs.

Cet aspect ne fait pas partie de notre étude de recherche, raison pour laquelle nous allons nous contenter de fournir une brève description de ces méthodes sans entrer dans les détails.

Sélection d'attributs

La sélection d'attributs consiste à sélectionner un sous-ensemble d'attributs qui peut efficacement décrire les exemples d'entrée et réduire les informations redondantes tout en garantissant de bons résultats de prédiction [Chandrashekar and Sahin, 2014]. Selon [Guyon and Elissee, 2003], les algorithmes de sélection d'attributs visent à améliorer la performance de prédiction, fournir des résultats plus rapides et moins coûteux et fournir une meilleure compréhensibilité et intelligibilité du modèle de prédiction.

Deux catégories d'algorithmes de sélection d'attributs se distinguent : les méthodes de filtre ("Filter methods") et les méthodes d'enveloppe ("Wrapper methods"). Les méthodes de filtre sélectionnent les variables en se basant sur les caractéristiques intrinsèques des données indépendamment du processus de classification [Ladha and Deepa, 2011]. Parmi les méthodes de filtre, les approches de [Peng et al., 2005] et [Fleuret, 2004] utilisent l'information mutuelle comme critère de sélection d'attributs et celle de [Yu and Liu, 2003] utilise une mesure de corrélation afin de sélectionner le sous-ensemble d'attributs pertinents et non redondants.

Contrairement aux méthodes de filtre, les méthodes "Wrapper" utilisent un algorithme de classification comme une enveloppe permettant d'évaluer le pouvoir de prédiction du sous-ensemble d'attributs sélectionné [Ladha and Deepa, 2011]. Dans [Ferchichi et al., 2009], une méthode de sélection d'attributs a été proposée en utilisant le classifieur SVM (Support Vector Machines) combiné avec un algorithme génétique pour sélectionner l'ensemble optimal de variables.

Extraction d'attributs

Les méthodes d'extraction d'attributs consistent à construire un ensemble de nouvelles variables, à partir des variables de départ, qui contient la plus grande part possible de l'information tout en réduisant la dimension [Benediktsson et al., 2003]. Quand les méthodes de sélection d'attributs risquent parfois de perdre une information pertinente (lorsqu'un très grand nombre d'attributs d'origine est fourni et un ensemble très restreint d'attributs est requis après la sélection), les méthodes d'extraction d'attributs utilisent toutes les informations fournies pour construire le nouvel ensemble d'attributs. Ces méthodes procèdent par application de transformations sur les attributs d'origine pour générer d'autres attributs plus significatifs [Khalid et al., 2014]. Ces méthodes peuvent être linéaires ou non-linéaires, supervisées ou non-supervisées. Parmi les méthodes d'extraction d'attributs les plus connues, on trouve l'ACP (Analyse en Composantes Principales) introduite dans [Pearson, 1901] et la LDA (Analyse Discriminante Linéaire) [Zhao et al., 2012, Ye and Ji, 2009]. D'autres alternatives ont été également proposées comme l'ICA [Hyvarinen and Oja, 2000], l'ISOMAP [Lee and Verleysen, 2007], etc.

1.3.3.2 Sélection de règles pertinentes

Comme le nombre de règles générées par les SCBRF peut être très important, une sélection des règles pertinentes peut réduire la complexité de ces systèmes et également augmenter leur performance. Une méthode de sélection des règles pertinentes par le concept d'oubli a été proposée dans [Nozaki et al., 1994]. Cette méthode consiste à construire un système de classification compact et performant en oubliant les règles non pertinentes qui n'ont aucun effet sur la classification des observations à classer. L'utilisation d'un algorithme génétique (AG) pour réduire le nombre de règles floues a été largement utilisée dans la littérature. D'abord, cette idée a été proposée dans [Ishibuchi et al., 1994] et [Ishibuchi et al., 1995b] dans le but d'éliminer les règles non-significatives. Puis, elle a été étendue pour le cas d'algorithmes de sélection à deux-objectifs (two-objective genetic rule selection) dans les travaux [Ishibuchi et al., 1995a, Ishibuchi et al., 1997]. Ces algorithmes de sélection, toujours basés sur les AG, cherchent à trouver un compromis entre le nombre de règles générées et la performance de classification du système. La sélection de règles à base d'algorithme génétique a été de nouveau étendue dans le cas de trois ou plusieurs objectifs pour prendre aussi en considération la longueur des règles générées [Ishibuchi et al., 2001a, Ishibuchi and Yamamoto, 2004, Alcalá et al., 2007].

Divers autres travaux plus récents se sont aussi focalisés sur l'utilisation d'algorithmes

génétiques pour construire un ensemble compact et performant de règles floues. GBML [Ishibuchi et al., 2005] et HGBML [Ishibuchi et al., 2005] proposent différents critères pour évaluer les règles floues basés sur les mesures support et confiance. L'évaluation dans ces travaux ne porte pas sur la qualité de chaque règle prise seule, mais des règles combinées en blocs sont évaluées ensemble afin de prendre en compte l'effet de combinaison des règles floues. Dans [Mansoori et al., 2008], un autre algorithme basé sur les AG, nommé SGERD, a été proposé. Dans cette méthode, un critère d'évaluation des règles floues basé sur la mesure support a été utilisé ; en outre, une fonction de renforcement du critère d'évaluation a été proposé pour distinguer plus efficacement les meilleures règles floues. Cette fonction met en valeur les degrés de compatibilité des exemples de test qui sont correctement classifiés par une règle floue et modifie par la suite la mesure d'évaluation de cette règle.

Toutes ces méthodes garantissent une réduction considérable du nombre de règles floues, cependant, le problème de la complexité de génération de ces règles demeure puisque la sélection est appliquée après la phase de génération des règles.

1.3.3.3 Regroupement d'attributs

Afin de réduire le nombre de règles floues générées dans les SCBRF, une autre solution intéressante consiste à regrouper les attributs en sous-blocs plus restreints d'attributs et traiter chaque bloc indépendamment. Cette approche, appelée Regroupement d'attributs, présente l'avantage de ne pas risquer d'éliminer des attributs qui peuvent être intéressants pour le modèle de classification. De plus, si un grand nombre d'attributs est difficile à traiter par les SCBRF, cette méthode de regroupement d'attributs permet de créer des sous-problèmes de complexité réduite ; les attributs de chaque sous-problème sont traités séparément. Cela permet de réduire la complexité des SCBRF et de réduire ainsi le nombre de règles générées sans écarter aucun attribut. Cette approche a été initialement introduite dans un cadre non flou dans [Borgi, 1999]. Puis, ce travail a été étendu dans un cadre flou, pour la génération de règles dans les SCBRF [Soua et al., 2013]. L'approche de regroupement d'attributs se base sur le concept des ensembles d'apprentissage artificiel, qui repose sur la combinaison des décisions de plusieurs apprenants pour améliorer l'exécution du système global [Valentini and Masulli, 2002]. Cette approche est au coeur de notre travail, le chapitre suivant lui est donc consacré.

Conclusion

Nous avons fourni, dans ce chapitre, une présentation générale des problèmes d'apprentissage automatique et nous nous sommes focalisés sur les méthodes d'apprentissage à base de règles, notamment les méthodes d'extraction des Règles d'Association et les méthodes de classification à base de règles floues. Chacune de ces deux méthodes d'apprentissage concerne un axe différent ; la première est une méthode d'apprentissage non-supervisée qui

cherche à détecter des associations intéressantes entre des données non-étiquetées, alors que la deuxième correspond à une méthode d'apprentissage supervisée qui traite les caractéristiques des données labelisées pour construire un modèle de prédiction performant.

Dans notre travail de thèse, nous avons étudié une manière de combiner les deux méthodes d'apprentissage présentées dans ce chapitre (les RAs et les SCBRF), pour former un classifieur compact et performant tout en réduisant la complexité des SCBRF.

Dans ce chapitre, nous avons également attiré l'attention sur le fait que la génération des règles floues dans les SCBRF met en place une complexité importante dès que le nombre d'attributs traités dépasse une dizaine d'attributs. Nous avons ainsi exposé quelques solutions proposées dans la littérature pour réduire cette complexité. Parmi ces solutions, nous nous intéressons à l'approche de regroupement d'attributs que nous détaillons dans le chapitre suivant.

Méthodes d'ensemble et Regroupement d'attributs

Introduction

Comme nous avons vu dans le chapitre précédent, l'un des problèmes majeurs des systèmes de classification à base de grille floue est l'explosion du nombre de règles générées quand le nombre d'attributs augmente. La sélection d'attributs et la sélection de règles sont parmi les solutions les plus connues et les plus utilisées pour la réduction de la complexité des systèmes de classification. L'approche de sélection de règles peut considérablement réduire le nombre de règles générées en éliminant les règles non pertinentes et les règles redondantes. Néanmoins, le temps d'apprentissage reste important puisque la sélection des règles est lancée après la phase d'apprentissage et de génération de règles. La méthode de sélection d'attributs peut quant à elle réduire le nombre de règles générées ainsi que le temps d'apprentissage, cependant elle peut parfois exclure des attributs potentiellement intéressants pour le modèle de classification.

Dans le cadre de cette thèse, nous nous intéressons à l'approche de regroupement d'attributs qui permet de décomposer un problème de complexité importante en des sous-problèmes de complexités inférieures. Cette méthode permet de réduire le nombre de règles floues ainsi que le temps nécessaire à la phase d'apprentissage et de classification. Elle permet également de conserver la totalité des informations sans avoir à exclure aucun attribut.

La méthode de regroupement d'attributs apparaît comme une méthode de pré-traitement ou de transformation de données. Elle fait partie des méthodes d'ensembles qui reposent sur l'idée de combiner plusieurs classifieurs afin d'améliorer la prédiction du système global. Pour cela, nous présentons d'abord dans ce chapitre le concept des ensembles d'apprentissage artificiel et nous exposons quelques méthodes d'ensembles proposées dans la littérature. Nous décrivons par la suite l'approche de regroupement d'attributs comme étant une approche d'ensemble qui visent à partitionner l'ensemble des attributs descriptifs en des sous-groupes d'attributs et à affecter chaque sous-groupe à un apprenant. Afin de créer les sous-groupes d'attributs, plusieurs mesures de dépendances peuvent être utilisées. Nous présentons, dans la troisième partie de ce chapitre, quelques méthodes de

recherche de dépendances entre les attributs. Enfin, nous présentons un exemple de méthode d'apprentissage ensembliste, la méthode SIFCO [Soua et al., 2013], qui est basée sur l'approche de regroupement d'attributs et qui utilise la recherche de corrélation linéaire pour détecter les sous-groupes d'attributs liés. C'est de cette méthode que s'inspire le travail présenté dans le cadre de cette thèse.

2.1 Ensembles d'apprentissage artificiel

Les ensembles d'apprentissage artificiel désignent une très vaste famille d'algorithmes d'apprentissage supervisé dont le concept de base consiste en la combinaison des décisions de plusieurs apprenants pour former la décision finale. Selon Valentini et al. [Valentini and Masulli, 2002], la considération des opinions de plusieurs apprenants peut améliorer la performance du système global.

L'utilisation des méthodes d'ensemble est motivée par les raisons suivantes :

- Si la performance d'un modèle de prédiction sur les données utilisées lors de la phase d'apprentissage est élevée, cela ne signifie pas que le modèle possède une bonne capacité de généralisation (la performance de prédiction du modèle sur de nouvelles données différentes des données d'apprentissage). Combiner plusieurs classifieurs qui ont traité différents sous-ensembles de données peut améliorer la capacité de généralisation du système [Polikar, 2006]. Imaginons par exemple le cas d'un diagnostic médical ; consulter les opinions de différents experts médecins est meilleur que suivre le diagnostic d'un seul médecin qui peut avoir une expérience différente de celles des autres médecins [Polikar, 2006].
- Dans certain cas, le volume de données est très important pour être traité par un seul classifieur. Ainsi, répartir les données en des sous-ensembles moins volumineux peut réduire la complexité de la tâche d'apprentissage et la rendre plus efficace et rapide.
- Si nous avons différents ensembles de données fournis par plusieurs sources différentes, alors il s'avère impossible de traiter des données de sources et natures différentes par un même classifieur. Dans ce cas, chaque ensemble de données peut être traité par un classifieur différent ; les sorties de chaque classifieur sont ensuite combinées.
- L'un des principaux avantages des méthodes d'ensemble est la diversité. La diversité a été définie dans [Prudhomme and Lallich, 2007] comme étant la capacité des différents apprenants d'un ensemble à ne pas commettre les mêmes erreurs lors de la prédiction. La diversité est surtout obtenue lorsque des classifieurs de types différents sont utilisés et combinés pour traiter un seul problème.

Nous distinguons trois catégories de méthodes d'ensembles. Dans la première catégorie, plusieurs classifieurs de types différents sont appliqués sur les données d'apprentissage et les décisions des différents classifieurs sont combinées pour former la décision

finale. En effet, le choix du meilleur classifieur pour un problème donné n'est pas toujours évident. Pour cette raison, différents types de classifieurs sont considérés. Selon Dietterich [Dietterich, 2000], un modèle de classification très performant peut être généré par la combinaison de plusieurs classifieurs moins performants. Dans le travail de [Lazkano and Sierra, 2003] par exemple, les auteurs ont proposé un classifieur hybride qui combine les réseaux bayésiens avec l'algorithme des plus proches voisins. Dans la deuxième catégorie de méthodes d'ensemble, le même type de classifieur est utilisé, mais en fixant à chaque fois des paramètres différents. Cette méthode est surtout utile pour les classifieurs qui sont très dépendants à certains paramètres d'entrée. Par exemple, dans les réseaux de neurones par rétro-propagation, les poids initiaux du réseau sont fixés aléatoirement et ce choix affecte considérablement le résultat de la rétro-propagation [Kolen and Pollack, 1991]. Dans [Maclin and Shavlik, 1995], il a été démontré que la combinaison de plusieurs réseaux de neurones avec des poids différents pour chaque réseau peut améliorer la capacité de prédiction et de généralisation de ces systèmes. Finalement, la troisième catégorie consiste à utiliser le même type de classifieur mais cette fois-ci avec une projection différente des données d'entrée pour chaque classifieur.

Nous nous intéressons dans le cadre de cette thèse à la troisième catégorie où la même méthode d'apprentissage est utilisée sur une projection différente de données. L'idée de ce type de méthodes d'ensembles consiste à répartir l'information entre plusieurs apprenants, chaque apprenant réalise la phase d'apprentissage sur l'information qui lui a été fournie, et les opinions "individuelles" des différents apprenants sont ensuite combinées pour atteindre une décision finale. L'information à répartir entre plusieurs apprenants peut correspondre aux exemples d'apprentissage, aux attributs descriptifs ou encore aux classes. Nous décrivons par la suite quelques travaux de cette catégorie de méthodes d'ensembles selon le type d'information à répartir.

2.1.1 Manipulation des exemples d'apprentissage

Cette méthode consiste à utiliser le même algorithme d'apprentissage plusieurs fois, chaque fois avec un sous-ensemble différent des données d'apprentissage. Cette méthode est spécialement intéressante pour les algorithmes d'apprentissage instables : les algorithmes dont la décision varie de façon importante en réponse à de petits changements dans les données d'apprentissage [Dietterich, 2000]. Le Bagging [Breiman, 1996] et le Boosting [Freund, 1995] sont parmi les méthodes d'ensemble les plus connues qui manipulent les données d'apprentissage. La méthode Bagging consiste à construire L sous-ensembles de données sélectionnées aléatoirement de l'ensemble de départ. La sélection se fait d'une manière aléatoire avec remplacement ; c.à.d, les sous-ensembles construits peuvent contenir des données dupliquées et d'autres données peuvent être omises (oubliées). La méthode Boosting quant à elle génère les sous-ensembles de données en se basant sur la performance des classifieurs en série. En d'autres termes, elle choisit chaque sous-ensemble d'apprentissage de manière à favoriser les instances qui ont été mal-classifiées par les classifieurs précé-

dents. L'algorithme AdaBoost [Freund and Schapire, 1996] est le plus fameux algorithme de l'approche Boosting. La méthode Boosting a été appliquée dans le contexte des SCBRF dans plusieurs travaux tels que [Amouzadi and Mirzaei, 2010], [Del Jesus et al., 2004] et [Hoffmann, 2001]. En particulier, les auteurs dans [Amouzadi and Mirzaei, 2010] proposent d'utiliser l'algorithme Adaboost pour combiner plusieurs classifieurs "faibles" appelés en série et d'adopter l'algorithme génétique pour chaque classifieur.

2.1.2 Manipulation des attributs descriptifs

La deuxième possibilité consiste à considérer la totalité des exemples d'apprentissage mais en considérant une partie des attributs descriptifs pour chaque classifieur. L'idée est de partitionner l'ensemble des attributs descriptifs en des sous-groupes d'attributs et de traiter chaque sous-groupe par un classifieur. Cette méthode est intéressante surtout dans le cas de problèmes de grandes dimensions où les algorithmes d'apprentissage ne sont pas capables de traiter un nombre volumineux d'attributs.

Parmi les méthodes ensemblistes basées sur le partitionnement en des groupes d'attributs, nous citons la méthode Random Subspace [Ho, 1998, Skurichina and Duin, 2002]. Cette méthode considère plusieurs classifieurs où chaque classifieur traite un sous-ensemble d'attributs sélectionnés aléatoirement. Random Subspace a été utilisée dans [Cordon et al., 2008] en combinaison avec la méthode Bagging pour construire un modèle de règles floues. Les tests expérimentaux dans [Cordon et al., 2008] ont montré que l'utilisation des méthodes d'ensemble donne lieu à des résultats intéressants en globalité. Cependant, l'utilisation de Random Subspace n'a pas toujours donné de bons résultats, et les auteurs ont affirmé que la sélection aléatoire des attributs n'est pas toujours significative et qu'une heuristique de sélection comme par exemple l'information mutuelle peut être plus intéressante. Ainsi, afin de choisir les meilleurs classifieurs parmi tous les classifieurs générés, les auteurs dans [Cordon et al., 2008] ont proposé un algorithme génétique pour sélectionner l'ensemble optimal de classifieurs dans le but de garder un compromis entre la bonne performance et la réduction de la complexité du système. D'autres travaux ont utilisé les algorithmes d'optimisation multi-objectifs (Multi-Objective Evolutionary Algorithm : MOEA) pour sélectionner un ensemble compact des classifieurs générés [Nojima and Ishibuchi, 2006, Cao et al., 2011]. Les objectifs recherchés dans ces méthodes consistent à réduire la taille des classifieurs combinés et garantir la diversité (en éliminant les classifieurs similaires). Pour ce faire, les auteurs de [Nojima and Ishibuchi, 2006] ont proposé un critère de sélection basé sur l'entropie et les auteurs de [Cao et al., 2011] ont proposé une méthode d'élagage des classifieurs à base de la réduction d'erreur (reduce-error based ensemble pruning Method).

SIFCO [Sousa et al., 2013] est une autre méthode d'apprentissage supervisé ensembliste qui utilise une recherche de corrélation linéaire entre les attributs numériques pour former les sous-ensembles d'attributs. Chaque groupe d'attributs corrélés est traité par un classifieur (un système d'apprentissage à base de règles floues) et les sorties des différents

classifieurs sont ensuite combinées pour former un modèle de prédiction final. L'intuition de cette méthode est que le traitement des attributs corrélés en bloc constitue une synthèse de l'information portée par une partie des données ; la redondance de l'information contribue à la réduction du bruit présent sur chaque attribut [Soua et al., 2013]. La même idée de partitionner l'ensemble des attributs en des sous-groupes d'attributs en utilisant une recherche de corrélation linéaire a été utilisée dans la méthode SUCRAGE [Borgi, 1999]. Cette méthode utilise deux méthodes de raisonnement non-flou (raisonnement exact et raisonnement approximatif) pour construire des règles de classification non-floues.

Dans [Prudhomme and Lallich, 2007], une autre méthode d'ensemble a été proposée qui, contrairement à SIFCO, partitionne les attributs en des groupes non-corrélés. Le but de ce travail est d'utiliser une carte auto-organisatrice (Self Organizing Map : SOM) pour chaque groupe d'attributs ; ces cartes fournissent une représentation différente et pertinente des données. Enfin, la prédiction est réalisée par un vote des différentes cartes.

Ces différentes méthodes permettent ainsi de contourner les problèmes liés aux espaces de grandes dimensions tout en conservant la totalité des attributs.

2.1.3 Manipulation des sorties

Dans ce troisième type de méthodes d'ensemble, la répartition des données entre les apprenants se fait par répartir les classes qui représentent les sorties d'un problème de classification. Plusieurs méthodes d'ensemble ont été proposées afin de simplifier le traitement des problèmes ayant un nombre important de classes [Krawczyk et al., 2015, Galar et al., 2011]. Il s'agit de combiner des classifieurs avec des représentations différentes et plus simples de l'attribut *Classe*. La méthode ECOC (Error Correcting Output Code) représente l'une des méthodes les plus caractéristiques de cette catégorie de méthodes [Dietterich and Bakiri, 1995]. Elle se base sur la combinaison de L classifieurs binaires ; chaque classifieur traite un sous-problème de deux méta-classes où chaque méta-classe consiste en une combinaison de la moitié des classes d'origine sélectionnées aléatoirement. Les auteurs dans [Dietterich and Bakiri, 1995] montrent que cette technique améliore la performance de l'algorithme d'arbre de décision C4.5 ainsi que les réseaux de neurones par rétro-propagation dans plusieurs problèmes de classification. Une deuxième méthode appartenant à cette catégorie de méthodes d'ensemble consiste à traiter une classe par rapport aux autres (one-against-all class binarization). Cette méthode consiste à transformer un problème de C classes en C sous-problèmes binaires ; chaque sous-problème utilise les exemples de sa classe comme des exemples positifs et les exemples de toutes les autres classes comme des exemples négatifs. Cette technique a été proposée dans le cadre des réseaux de neurones dans [Ou and Murphey, 2007] et dans le cas des machines à vecteur de support (SVM) dans [Rifkin and Klautau, 2004].

2.2 Regroupement d'attributs dans les SCBRF

Nous nous intéressons dans ce rapport aux méthodes d'ensemble qui divisent les attributs descriptifs en des sous-blocs et fournissent un sous-bloc d'attributs à un classifieur. Nous appelons ce type de méthodes les méthodes de regroupement d'attributs. Dans cette section, nous discutons l'utilisation de ces méthodes dans le contexte des SCBRF et l'effet de cette approche sur le nombre et la qualité des règles floues générées.

2.2.1 Principe de l'approche de regroupement

Nous nous intéressons dans le cadre de cette thèse aux systèmes de classification à base de règles floues, spécifiquement les méthodes à base de grille floue. Comme nous l'avons indiqué précédemment, un grand nombre d'attributs descriptifs peut considérablement affecter la capacité de prédiction de ces systèmes et l'intelligibilité du modèle de classification. Nous nous focalisons dans ce chapitre sur l'approche de regroupement d'attributs comme étant une solution intéressante de ce problème. L'approche de regroupement d'attributs se base sur le concept des ensembles d'apprentissage artificiel ; l'information à répartir dans cette approche correspond aux attributs descriptifs. L'idée est de partitionner l'ensemble des attributs de départ en plusieurs groupes d'attributs, chaque classifieur utilise un groupe d'attributs et construit une base de règles locale, puis les différentes bases locales obtenues sont combinées pour former le modèle final de classification.

Dans le cadre des systèmes de classification à base de règles floues, l'approche de regroupement d'attributs a été utilisée dans [\[Soua et al., 2013\]](#) afin de réduire la complexité de ces systèmes. L'utilisation de l'approche de regroupement d'attributs comme phase de pré-traitement de données dans [\[Soua et al., 2013\]](#) a montré une réduction conséquente du nombre de règles générées sans trop altérer les taux de bonnes classifications.

2.2.2 Nombre de règles optimisé

Considérons le cas d'un problème de classification à N attributs où chaque attribut est partitionné en k sous ensembles flous. Nous avons vu précédemment que le nombre de règles générées par les SCBRF (en utilisant une grille floue simple) est égale à k^N . Nous souhaitons maintenant analyser le nombre de règles générées par les SCBRF combinés avec l'approche de regroupement d'attributs. Le problème d'apprentissage est ainsi divisé en G sous-problèmes g_1, g_2, \dots, g_G . Chaque sous-problème g_i contient N_i attributs. La même démarche de génération de règles des SCBRF est utilisée pour chaque sous-problème et le même nombre de sous-ensembles flous est utilisé pour partitionner les attributs. Le nombre de règles générées par cette approche, notée NR_{regrp} , est alors égal à :

$$NR_{regrp} = \sum_{i=1}^G k^{N_i} \quad (35)$$

Il a été démontré dans [Soua et al., 2013] que si les groupes d'attributs, issus de l'approche de regroupement, forment une partition de l'ensemble des attributs de départ ($\sum_{i=1}^G N_i = N$), alors :

$$NR_{regrp} = \sum_{i=1}^g k^{n_i} \leq \prod_{i=1}^g k^{n_i} = k^{\sum_{i=1}^g n_i} = k^n \quad (36)$$

Par conséquent, l'utilisation de l'approche de regroupement d'attributs garantit la réduction du nombre de règles et ce dans le cas où les groupes d'attributs générés forment une partition de l'ensemble des attributs de départ. Dans le pire des cas, tous les attributs seront dépendants et regroupés ensemble dans le même groupe ; dans ce cas le nombre de règles est égal au même nombre obtenu par les SCBRF classique (k^N). Dans l'autre cas extrême, aucune association ne sera détectée par la phase de regroupement ; dans ce cas, chaque attribut sera considéré comme indépendant et traité seul par un classifieur. Le nombre de règles générées dans ce cas est $N * k$ règles ; il représente le nombre minimal de règles qu'on peut obtenir. Il est à noter que ce nombre représente le meilleur des cas si nous nous intéressons uniquement à la réduction du nombre de règles. Cependant, si nous considérons de plus le critère de la performance de classification, les règles générées dans ce cas ne sont pas nécessairement significatives et performantes.

Le but de la phase de regroupement d'attributs consiste alors en la recherche d'un compromis entre un nombre de règles réduit et une performance de classification élevée. Pour ce faire, la méthode de regroupement d'attributs doit être capable de générer les associations les plus significatives entre les attributs et de former des groupes d'attributs ayant un grand pouvoir de prédiction.

2.2.3 Taille réduite des prémisses

La réduction du nombre de règles générées ne représente pas le seul avantage de l'utilisation de la phase de regroupement d'attributs dans les SCBRF. En effet, cette approche améliore aussi l'intelligibilité des règles floues générées en réduisant la taille de leurs prémisses. En effet, nous avons vu dans la section 1.3.1.2 qu'une règle floue correspondant au sous-espace $A_i^k \times A_j^k$ est de la forme :

$$si X_1 est A_i^k et X_2 est A_j^k alors E = (X_1, X_2) appartient à y_{ij}^k avec CF = CF_{ij}^k \quad (37)$$

Cette règle est obtenue pour un problème à deux attributs X_1 et X_2 . Imaginons maintenant que nous traitons un problème à 60 attributs avec un SCBRF classique, la règle contient dans ce cas 60 conditions dans sa prémisse. Cela rend la règle non-lisible par l'être humain. L'approche de regroupement d'attributs construit des groupes d'attributs de taille réduite et un ensemble de règles floues sera construit pour chaque groupe d'attributs. Cette méthode garantit alors que les règles obtenues sont plus intelligibles car de taille réduite.

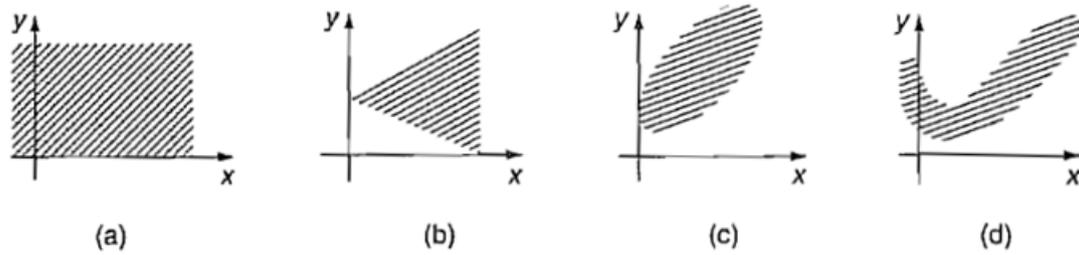


Figure 17 — Corrélation et dépendance entre variables numériques [Saporta, 2006]

2.3 Dépendance entre les attributs

Nous avons vu dans la section précédente que l'approche de regroupement d'attributs permet de grouper des attributs liés et les traiter séparément. La question qui se pose maintenant est comment choisir les attributs qui seront regroupés ensemble ? Quelle méthode utiliser pour définir un groupe d'attributs comme dépendants ? Nous étudions dans ce paragraphe quelques méthodes de recherche de dépendance entre les variables fournies dans la littérature.

Nous nous focalisons sur les mesures de dépendances entre deux variables. Il s'agit de découvrir des relations ou des associations intéressantes entre deux variables. Deux variables sont dites associées lorsqu'un changement dans les valeurs de l'une engendre un changement dans les valeurs de l'autre [Liebertrau, 1983].

Selon la nature des données traitées, différentes méthodes et mesures ont été proposées dans la littérature afin de détecter la dépendance entre les variables. Nous distinguons deux types de données principales : les données quantitatives et les données qualitatives. Lorsqu'il s'agit de variables quantitatives, on peut attribuer un certain ordre entre les valeurs d'une variable. Dans ce cas, une augmentation (ou diminution) des valeurs d'une variable peut être corrélée avec une augmentation (ou diminution) d'une autre variable. On parle donc de mesures de corrélation entre des variables quantitatives. Cependant, lorsqu'il s'agit de variables qualitatives, qui s'expriment par l'appartenance à une catégorie ou modalité, on parle de mesures d'association plutôt que de corrélation.

2.3.1 Mesures de corrélation entre des variables quantitatives

D'abord, afin de caractériser graphiquement la notion de corrélation entre deux variables continues, un nuage de points peut être utilisé [Saporta, 2006]. La figure 17 présente 4 nuages de points : le premier montre une absence de corrélation (a), le deuxième illustre une absence de liaison en moyenne mais pas en dispersion (b), le troisième montre une corrélation linéaire (c) et le dernier présente une corrélation non-linéaire (d). Il existe beaucoup de mesures de corrélation dans la littérature ; le coefficient de corrélation de Pearson et le coefficient des rangs de Spearman sont parmi les mesures les plus connues. Nous ne présentons dans ce travail que ces deux dernières mesures. Pour cela, nous sup-

posons qu'on traite un problème à p individus et 2 variables X et Y .

2.3.1.1 Coefficient de corrélation de Pearson

Ce coefficient a été introduit par "Bravais-Pearson" dans [Pearson, 1926] ; il mesure le caractère plus ou moins linéaire du nuage de points [Saporta, 2006]. Ce coefficient de corrélation linéaire, désigné par r , est défini comme suit :

$$r = \frac{\frac{1}{p} \sum_{i=1}^p (x_i - \bar{x})(y_i - \bar{y})}{S_X S_Y} \quad (38)$$

où x_i représente la i^{me} valeur de X et y_i représente la i^{me} valeur de Y . S_X et S_Y sont les écarts-types de X et Y : $S_X^2 = \frac{1}{p} \sum_{i=1}^p (x_i - \bar{x})^2$ et $S_Y^2 = \frac{1}{p} \sum_{i=1}^p (y_i - \bar{y})^2$.

La valeur du coefficient r ne peut avoir qu'une valeur comprise entre -1 et +1. Si la valeur absolue de r est égale à 1 ($|r| = 1$), alors une corrélation linéaire exacte existe entre les deux variables.

Il est à noter que le coefficient de corrélation de Pearson n'est pas toujours robuste. L'auteur dans [Saporta, 2006] note que le coefficient r est en particulier très sensible aux individus extrêmes (voir Figure 18). De plus, même si ce coefficient n'est sensé détecter que les corrélations linéaires, sa valeur peut être élevée dans le cas de corrélations non linéaires, ce qui est inattendu ! Par exemple, les trois nuages de la figure 19 possèdent tous le même coefficient de corrélation ($r = 0.82$) bien que seul le premier nuage reflète réellement une corrélation linéaire.

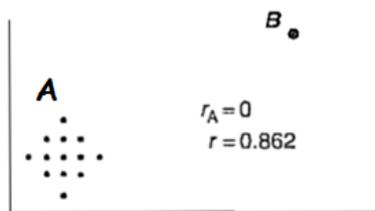


Figure 18 — Sensibilité du coefficient de Pearson aux individus extrêmes



Figure 19 — Différentes formes de corrélation avec le même coefficient r

2.3.1.2 Coefficient des rangs de Spearman

Une variable mesurable n'est pas toujours numérique, néanmoins on peut avoir des variables ordinales qui ne sont pas continues mais qui possèdent un ordre quelconque sur l'ensemble des valeurs. Le coefficient de Pearson n'est pas utilisable dans ce cas. Le coefficient de corrélation des rangs de Spearman peut être utilisé pour mesurer la liaison entre deux variables ordinales.

Il faut d'abord assigner à chaque valeur son rang selon chaque variable. Étudier la liaison entre deux variables X et Y revient donc à comparer les classements issus de ces deux variables. Pour les données numériques, il est aussi facile de remplacer les valeurs des variables par leurs rangs et puis d'appliquer le coefficient de corrélation de Spearman selon la définition suivante :

$$r_s = 1 - \frac{6 \sum_{i=1}^p d_i^2}{p(p^2 - 1)} \quad (39)$$

Où $d_i = \text{rang}(x_i) - \text{rang}(y_i)$

Selon Saporta [Saporta, 2006], les coefficients de corrélation de rangs sont en fait des coefficients de dépendance monotone et ne cherchent pas à détecter seulement les corrélations linéaires. Par exemple, le nuage de points de la figure 20 donne un coefficient de corrélation linéaire $r = 0.85$ mais un coefficient de Spearman $r_s = 1$. Même dans le cas de corrélation linéaire, le coefficient de Spearman est préférable au coefficient linéaire de Pearson car il est plus robuste et moins sensible aux valeurs extrêmes.

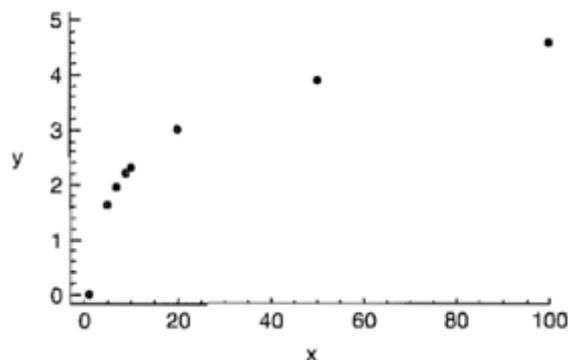


Figure 20 — Corrélation monotone non-linéaire

2.3.2 Mesures d'association entre des variables qualitatives

Une variable qualitative s'exprime par l'appartenance à une catégorie parmi un ensemble de catégories (ou de modalités) où aucun ordre n'existe entre ces catégories. C'est pour cette raison qu'on peut également les appeler variables catégoriques. On considère toujours le cas de p individus et 2 variables X et Y qui sont qualitatives. X possède r modalités, notées x_1, x_2, \dots, x_r , et Y possède s modalités, notées y_1, y_2, \dots, y_s .

2.3.2.1 Tableau de contingence

Les données sont présentées sous la forme d'un tableau à r lignes et s colonnes, appelé tableau de contingence, qui renferme les effectifs p_{ij} (où $X = x_i$ et $Y = y_j$).

	y_1	y_2	...	y_j	y_s	
x_1	p_{11}	p_{12}			p_{1s}	$p_{1.}$
x_2	p_{21}	p_{22}			p_{2s}	$p_{2.}$
...						
x_i				p_{ij}		$p_{i.}$
...						
x_r	p_{r1}	p_{r2}			p_{rs}	$p_{r.}$
	$p_{.1}$	$p_{.2}$		$p_{.j}$	$p_{.s}$	p

Tableau 5 — Table de contingence

Les $p_{i.}$ et les $p_{.j}$ s'appellent respectivement marges en lignes et marges en colonnes :
 $p_{i.} = \sum_j p_{ij}$ et $p_{.j} = \sum_i p_{ij}$

On appelle tableau des profils-lignes le tableau des fréquences conditionnelles $\frac{p_{ij}}{p_{i.}}$ et tableau des profils-colonnes le tableau des fréquences conditionnelles $\frac{p_{ij}}{p_{.j}}$.

Lorsque tous les profils-lignes sont identiques, on peut parler d'indépendance entre X et Y puisque la connaissance de X ne change pas les distributions conditionnelles de Y . Dans ce cas, tous les profils-colonnes sont également identiques.

2.3.2.2 Test d'indépendance de Chi-deux

Le test de Chi-deux est classiquement utilisé pour mesurer l'hypothèse d'indépendance entre deux variables qualitatives. Pour un tableau à r lignes et s colonnes, le test de Chi-deux, noté χ^2 , s'écrit comme suit :

$$\chi^2 = \sum_{i=1}^p \sum_{j=1}^p \frac{(p_{ij} - \frac{p_{i.}p_{.j}}{p})^2}{\frac{p_{i.}p_{.j}}{p}} \quad (40)$$

La valeur de χ^2 est nulle dans le cas de l'indépendance et non nulle sinon.

2.3.2.3 Le coefficient V de Cramer

Le coefficient V de Cramer est une autre mesure d'association très populaire qui se base sur le coefficient de Chi-deux. L'avantage de V de Cramer est qu'elle offre une normalisation des valeurs entre 0 et 1. Dans ce cas, une valeur près de 0 signifie une faible association entre les variables tandis qu'une mesure proche de 1 signifie une forte relation.

Le coefficient V de Cramer se définit comme suit [Cramer, 1999] :

$$V = \sqrt{\frac{\chi^2}{p(l-1)}} \quad (41)$$

Où p représente la taille de la population et l désigne le nombre de lignes ou de colonnes (le minimum entre eux).

2.3.2.4 Information mutuelle

L'information mutuelle de deux variables X et Y , notée $I(X; Y)$, est une quantité qui mesure la dépendance mutuelle de deux variables [Moshe, 2006]. Elle mesure à quel niveau la connaissance de X réduit notre incertitude sur Y et mutuellement la connaissance de Y réduit notre incertitude sur X . L'information mutuelle est non-négative ($I(X; Y) > 0$) et elle est symétrique ($I(X; Y) = I(Y; X)$). Cette valeur est égale à 0 si les deux variables sont strictement indépendantes et augmente avec leur dépendance. L'information mutuelle est définie dans [Yao, 2003] comme suit :

$$I(X; Y) = \sum_{x \in V_X} \sum_{y \in V_Y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \quad (42)$$

où $P(x, y)$ est la distribution de probabilité jointe de X et Y , et $P(x)$ et $P(y)$ sont respectivement les distributions de probabilité marginale de X et de Y . V_X représente l'ensemble des valeurs possibles de X et V_Y désigne l'ensemble des valeurs possibles de Y . Le lecteur intéressé par cette mesure peut trouver plus de détails dans [Yao, 2003].

Le problème de l'utilisation de l'information mutuelle IM réside dans la méthode d'estimation des densités de probabilité. Une méthode intéressante d'estimation de l'IM a été proposée dans [Kraskov et al., 2004] ; elle se base sur l'utilisation de l'entropie de Shannon. Dans ce cas, l'information mutuelle s'écrit de la façon suivante :

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \quad (43)$$

Où H représente l'entropie de Shannon et est définie comme suit :

$$H(X) = - \sum_{x \in V_X} P(x) \times \log_2 P(x) \quad (44)$$

La valeur de l'information mutuelle dépend généralement des modalités des variables considérées [Crémilleux, 1991]. Pour cela, plusieurs normalisations de l'information mutuelle ont été proposées [Kvalseth, 1987, Hou, 1996]. Parmi ces mesures normalisées, nous citons le coefficient d'incertitude symétrique SU [Kvalseth, 1987] qui se définit comme suit :

$$SU(X, Y) = \frac{2I(X; Y)}{H(X) + H(Y)} \quad (45)$$

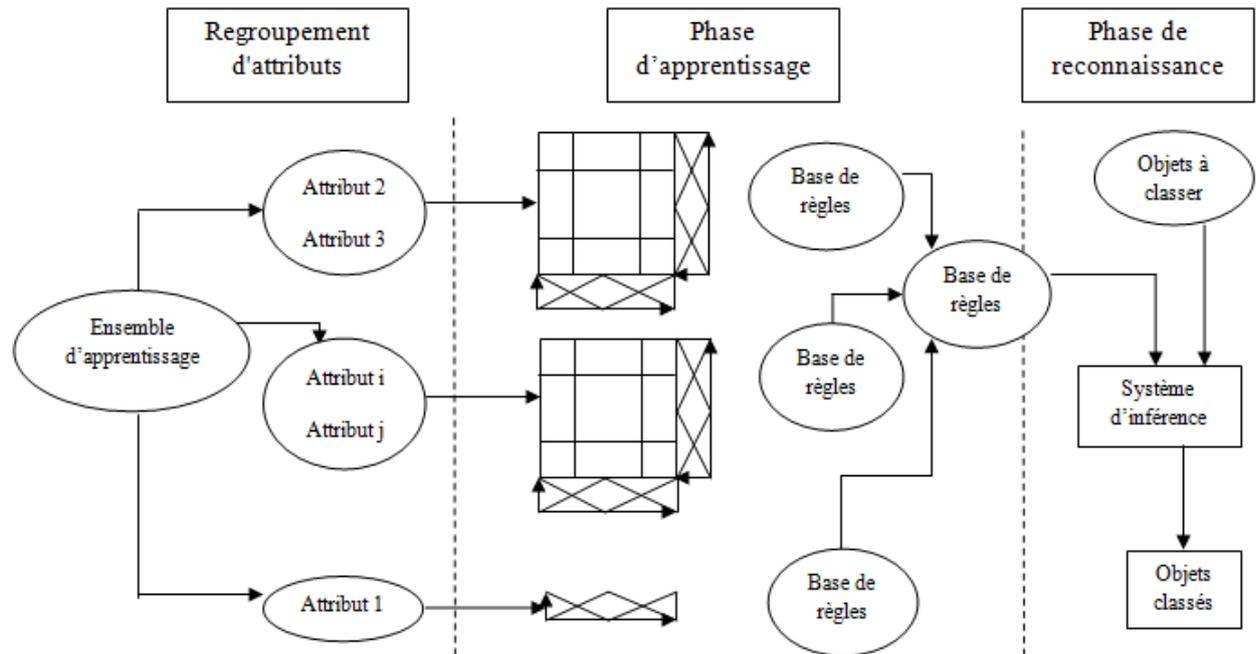


Figure 21 — Étapes de la méthode SIFCO

2.4 Méthode SIFCO

Nous présentons dans ce paragraphe une méthode d'ensemble flou qui utilise l'approche de regroupement d'attributs, la méthode SIFCO [Soua et al., 2013]. La figure 21 illustre les 3 phases principales de la méthode SIFCO :

- La phase de regroupement d'attributs : c'est dans cette phase que l'ensemble des attributs descriptifs sont divisés en groupes.
- La phase d'apprentissage : chaque groupe d'attributs est traité par un classifieur qui génère sa base de règles floues ; les différentes bases locales sont ensuite combinées pour former la base de règles globale.
- La phase de reconnaissance ou de classification : dans cette phase, la base de règles totale ainsi générée est utilisée pour classer un nouvel objet.

2.4.1 La phase de regroupement d'attributs

Les groupes d'attributs liés sont formés par une recherche de corrélation entre les attributs. Les données traitées dans les SCBRF sont quantitatives et le coefficient de corrélation linéaire de "Bravais-Pearson" a été utilisé dans SIFCO. Les attributs linéairement corrélés sont regroupés et traités ensemble par un classifieur. En plus, afin d'assurer la diversité des mesures de corrélation utilisées, le coefficient V de Cramer a été également utilisé. Une discrétisation régulière des attributs numériques a été appliquée dans le cas de l'utilisation du coefficient de Cramer afin de transformer les attributs en variables qualitatives.

Des tests expérimentaux avec les deux mesures ont été élaborés dans [Soua et al., 2013] en utilisant 5 bases de données différentes. Dans toutes les bases testées, nous remarquons que la mesure V de Cramer ne donne aucune association intéressante entre les attributs numériques ; ainsi chaque attribut est traité seul par un classifieur. Par contre, plusieurs associations ont été détectées par la mesure de corrélation linéaire de Pearson. Pour cette raison, nous nous intéressons dans le reste de cette thèse à la mesure de Pearson dans SIFCO vu que notre recherche cible la détection d'associations intéressantes entre les attributs.

La première étape consiste à calculer la matrice de corrélation R ; $R = (r_{i,j})$ avec $1 \leq i \leq N$, $1 \leq j \leq N$ et N le nombre total d'attributs.

$$R = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & r_{1,N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{N,1} & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

L'élément $r_{i,j}$ est le coefficient de corrélation de Bravais-Pearson entre les attributs X_i et X_j .

La deuxième étape consiste au seuillage de la matrice R . Deux attributs X_i et X_j sont dits corrélés si la valeur absolue de $r_{i,j}$ est supérieure à un seuil θ prédéfini. La matrice seuillée $R_\theta = (rs_{i,j})$ est donc construite de la manière suivante :

Si $|r_{i,j}| > \theta$ alors $rs_{i,j} = 1$ et les attributs X_i et X_j sont corrélés.

Sinon $rs_{i,j} = 0$ et les attributs X_i et X_j ne sont pas corrélés.

La dernière étape consiste à déterminer, à partir de la matrice seuillée R_θ , les sous-ensembles d'attributs corrélés. Pour cela, la matrice est vue comme un graphe : les noeuds du graphe sont les attributs et deux noeuds sont reliés par un arc si et seulement si ils sont corrélés. Par exemple, si nous avons la matrice seuillée R_θ ci-dessous, le graphe de la figure 22 présente les groupes d'attributs associés selon cette matrice.

$$R_\theta = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

2.4.2 La phase d'apprentissage

Pour chaque groupe d'attributs corrélés, un ensemble de règles floues est généré. La partition floue simple et la partition floue supervisée sont utilisées dans [Soua et al., 2013] pour construire la grille floue. Une présentation détaillée de la partition floue supervisée et

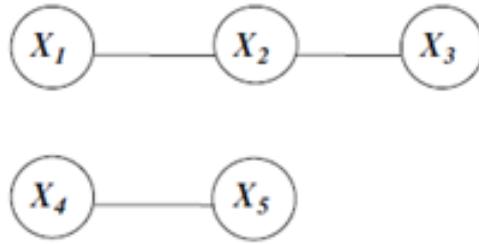


Figure 22 — Graphe des attributs corrélés

des fonctions d'appartenance utilisées dans cette méthode est fournie dans le paragraphe 1.3.1.3. Les tests expérimentaux dans [Soua et al., 2013] ont montré que l'utilisation de la partition floue supervisée est plus appropriée et contribue à l'amélioration des résultats de classification.

Ensuite, la méthode heuristique d'Ishibuchi [Ishibuchi and Yamamoto, 2005, Ishibuchi et al., 1992] est utilisée pour construire les règles floues. Les attributs corrélés apparaissent dans une même prémisse. Des règles de prémisses de taille différente et de combinaisons différentes d'attributs sont alors construites. Les règles floues générées par les différents classifieurs sont combinées pour former la base de règles globale. Le découpage en des groupes d'attributs liés garantit la génération d'un nombre de règles inférieur à celui généré par les SCBRF classiques [Ishibuchi and Yamamoto, 2005, Ishibuchi et al., 1992, Ishibuchi et al., 1994] (voir paragraphe 2.2). En outre, une propriété intéressante des règles générées par la méthode SIFCO est que ces règles ne sont pas contradictoires, ni redondantes [Soua et al., 2013]. Ceci est garanti par le fait que les groupes d'attributs construits forment une partition de l'ensemble des attributs de départ; en d'autres termes, aucune intersection ou chevauchement n'existe entre ces groupes. Ainsi, il n'est pas nécessaire de simplifier la base de règles générées pour éliminer les redondances et contradictions.

2.4.3 La phase de reconnaissance

La base de règles globale générée est utilisée pour classer un nouvel objet. La méthode de la seule règle gagnante est utilisée dans SIFCO pour choisir la classe à associer au nouvel exemple. En d'autres termes, la conséquence de la règle R_{ij}^k gagnante, celle qui possède le produit $\mu_i^k(X'_1) \times \mu_j^k(X'_2) \times CF_{ij}^k$ maximum, est assignée au nouvel exemple $E' = (X'_1, X'_2)$ (voir paragraphe 1.3.2.1).

Conclusion

Dans ce chapitre, nous avons présenté un état de l'art succinct sur les méthodes d'ensemble d'apprentissage où la combinaison de différents classifieurs est utilisée pour améliorer la performance des systèmes de classification. Plus précisément, nous nous sommes

intéressés aux méthodes d'ensemble qui adoptent le partitionnement des attributs descriptifs en des groupes d'attributs liés et utilisent un classifieur flou pour chaque groupe. Cette démarche aide ainsi à réduire la complexité des problèmes à haute dimension. Nous avons décrit dans ce chapitre quelques méthodes de recherche de corrélation et d'association entre les variables. Parmi ces mesures, le coefficient linéaire de Pearson et le coefficient V de Cramer ont été appliqués dans une méthode d'ensemble flou, la méthode SIFCO, pour former les groupes d'attributs liés. Toutes les mesures de corrélation et d'association étudiées dans ce chapitre sont basées sur des notions de statistique. Dans [Saporta, 2006], l'auteur a attiré l'attention sur le fait que la non-corrélation entre deux variables ne signifie pas forcément qu'elles sont indépendantes. Il peut évidemment exister d'autres types de liaisons entre les variables qui ne peuvent pas être calculées par les méthodes et mesures issues de l'approche statistique. Nous proposons ainsi dans le cadre de cette thèse une nouvelle méthode qui détecte les associations entre les variables aussi bien numériques que catégoriques. Contrairement aux mesures statistiques déjà mentionnées, cette méthode se base sur des concepts de data mining, notamment sur le concept des règles d'association et de génération d'itemsets fréquents. Nous décrivons dans le chapitre suivant notre méthode de regroupement d'attributs.

Regroupement d'attributs par RA dans les SCBRF : la méthode SIFRA

Introduction

Nous nous sommes intéressés, dans le cadre de cette thèse, aux méthodes d'ensemble appliquées aux systèmes d'apprentissage à base de règles floues. Plus précisément, nous nous focalisons sur les méthodes d'ensemble qui décomposent les attributs d'entrée en des sous-groupes d'attributs et traitent chaque sous-groupe séparément par un classifieur. Ainsi, le problème de classification est décomposé en des sous-problèmes de complexité inférieure. Ces méthodes sont intéressantes surtout dans le cas des SCBRF puisque la génération des règles floues par les SCBRF classiques est compliquée dès que le nombre d'attributs traités dépasse une dizaine d'attributs. Parmi ces méthodes d'ensemble, nous avons évoqué la méthode *Random Subspace* qui sélectionne aléatoirement des attributs et les traite en blocs. Comme mentionné précédemment, la sélection aléatoire des attributs n'est pas toujours significative ; par contre une heuristique de sélection peut être plus intéressante [Cordon et al., 2008]. La méthode SIFCO [Soua et al., 2013] est une autre méthode ensembliste qui utilise une recherche de corrélation linéaire pour regrouper les attributs numériques et traiter chaque groupe d'attributs corrélés par un classifieur. L'hypothèse sous-jacente de cette méthode est que les attributs qui sont liés et qui partagent une information redondante peuvent être utilisés ensemble pour contribuer à la construction d'une base de règles locale. Néanmoins, nous attirons l'attention sur le fait que d'autres types de liaisons entre les attributs peuvent exister, autre que la corrélation linéaire. Par exemple, la figure 23 représente quelques types de liaisons non linéaires entre deux attributs X et Y [Kenny, 1979]. Nous proposons dans cette thèse une nouvelle méthode de regroupement d'attributs qui n'utilise pas la recherche de corrélation linéaire, mais se base sur le concept des règles d'association et plus précisément sur la génération des itemsets fréquents [Agrawal et al., 1993]. Les algorithmes de génération d'itemsets fréquents déterminent les associations intéressantes entre les attributs en analysant leurs apparitions simultanées dans les enregistrements de la base de données. Cette méthode peut être très intéressante pour les bases de données pour lesquelles il n'existe aucune

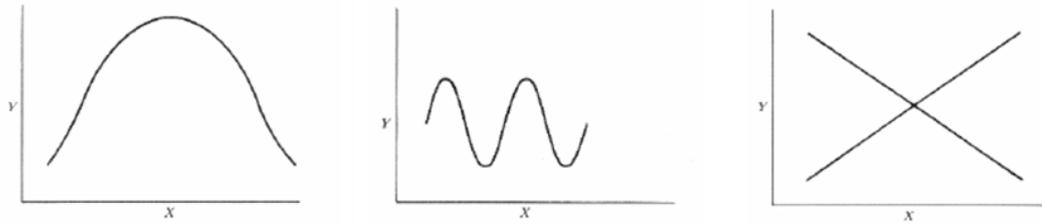


Figure 23 — Relations non linéaires : Curvilinéaire, Cyclique et Liaisons croisées

[Kenny, 1979]

relation de type corrélation linéaire entre les attributs. L'objectif de ce travail est de développer une méthode de regroupement d'attributs qui est capable de détecter des associations intéressantes entre les attributs sans chercher une forme ou un type spécifique d'association. En effet, le choix des groupes d'attributs liés peut affecter le modèle de prédiction ; les associations détectées doivent contribuer de manière efficace à la capacité de prédiction du système.

Pour développer notre méthode de regroupement d'attributs, nous nous sommes inspirés des travaux de [Agrawal et al., 1999] qui proposent un algorithme de "*Subspace Clustering*" en se basant sur le concept des Règles d'Association. Le travail de [Agrawal et al., 1999] s'inscrit dans le cadre de l'apprentissage non supervisé (le *Clustering*) ; il cherche à identifier les attributs qui forment ensemble des sous-espaces pertinents pour la partition de données (*Clustering*). Les auteurs ont proposé d'utiliser le concept des RA, et plus précisément la technique de l'algorithme Apriori, pour identifier les sous-espaces qui possèdent des régions denses en données. Nous pensons que la recherche de régions denses dans des sous-espaces d'attributs pourrait aussi être intéressante dans notre cas (dans les problèmes d'apprentissage supervisé) ; ces régions denses peuvent révéler des informations intéressantes pour la classification et peuvent être utilisées pour la génération des règles floues. Les régions denses, détectées par l'algorithme Apriori ou un autre algorithme de génération des itemsets fréquents, pourraient donc être utilisées pour déduire les groupes d'attributs liés. Inspirés du travail de [Agrawal et al., 1999], nous développons notre méthode de regroupement d'attributs basée sur la génération d'itemsets fréquents, et nous l'appliquons dans le contexte des systèmes d'apprentissage supervisé à base de règles floues.

Dans ce chapitre, nous présentons dans la première partie la méthode de *Subspace Clustering* de laquelle s'inspire notre travail. La deuxième partie du chapitre est consacrée à la présentation générale de notre système d'apprentissage SIFRA. Notre contribution, qui consiste à développer une méthode de regroupement d'attributs basée sur le concept des Règles d'Association, est décrite dans la troisième section de ce chapitre. Les différentes étapes de la méthode y sont finement détaillées. Ensuite, nous fournissons des discussions sur les différents choix et critères proposés dans notre méthode SIFRA. En particulier, la partie 4 de ce chapitre est dédiée à l'analyse et la discussion des méthodes de discrè-

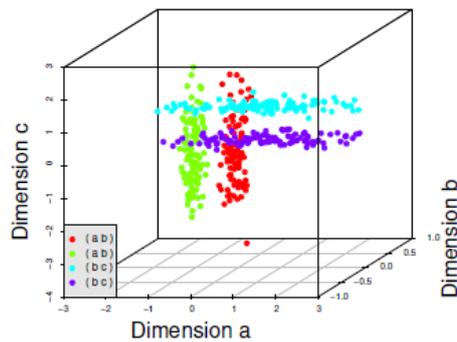


Figure 24 — Exemple de base de données avec quatre *clusters* qui chevauchent entre eux [Parsons et al., 2004]

tisation qui peuvent être utilisées dans notre méthode de regroupement d'attributs. Par ailleurs, parmi les étapes de notre méthode de regroupement d'attributs, des critères ont été proposés pour choisir les groupes d'attributs liés les plus intéressants. Une discussion sur le choix de ces critères est menée dans la cinquième partie de ce chapitre. Enfin, nous présentons dans la section 6 une analyse du coût global de la méthode proposée et nous suggérons quelques pistes pour réduire ce coût.

3.1 Les Règles d'Association pour le *Subspace Clustering*

Le *Subspace Clustering* est une méthode d'apprentissage non-supervisée qui vise à construire une partition des données d'apprentissage en des groupes (des "clusters") dans des sous-espaces de faible dimension. En effet, lorsque le nombre d'attributs est grand, certaines dimensions non pertinentes peuvent masquer des *clusters* intéressants [Agrawal et al., 1999]. Voyons par exemple la figure 24, quatre *clusters* peuvent être identifiés dans ce problème ; cependant, le chevauchement entre eux rend leur identification compliquée lorsque tous l'espace est considéré. Des projections sur deux dimensions différentes à chaque fois sont illustrées dans la figure 25. D'après cette figure, il convient de noter que les deux premiers *clusters* (le vert et le rouge) sont clairement séparés dans le sous-espace de dimensions a et b (Figure 25-a). Les deux autres *clusters* (le bleu et le violet) sont réunis dans la figure 25-a, mais ils sont facilement identifiés dans le sous-espace de dimensions b et c (Figure 25-b). Donc, pour trouver les différents clusters dans cette base, il suffit à chaque fois de considérer le sous-espace approprié.

Dans ce contexte, l'objectif du *Subspace Clustering* est la recherche des *clusters* dans les sous-espaces appropriés. Plusieurs méthodes ont été proposées pour identifier des sous-espaces pertinents pour la recherche des *clusters*. Parmi ces méthodes, certaines ont utilisé la densité des données pour caractériser la pertinence des sous-espaces, d'autres ont utilisé

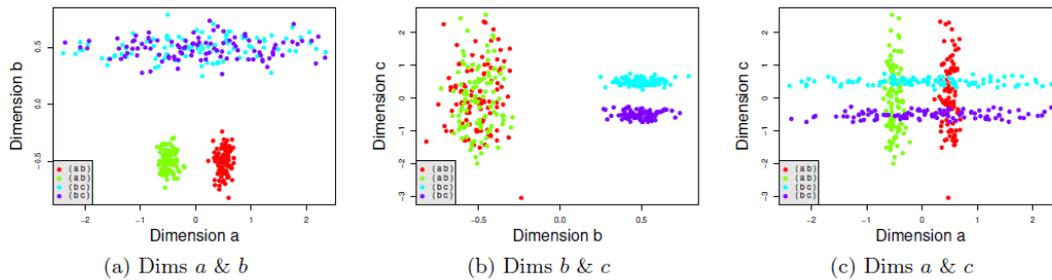


Figure 25 — Les données projetées sur trois sous-espaces différents
 [Parsons et al., 2004]

la mesure d'entropie.

Dans le cadre de cette thèse, nous nous intéressons à un algorithme spécifique du *Subspace Clustering* : l'algorithme CLIQUE [Agrawal et al., 1999]. Cet algorithme s'inspire du concept des Règles d'Association et identifie les sous-espaces qui contiennent des *clusters* intéressants en se basant sur la technique de l'algorithme Apriori. La propriété d'anti-monotonie d'Apriori est adaptée dans le contexte du *Subspace Clustering* comme suit :

"Si un *cluster* est identifié dans un espace de dimension k , il est aussi identifié dans tous les sous-espaces de dimension $k - 1$."

Pour ce faire, les attributs numériques sont d'abord partitionnés en des intervalles de longueur égale. Le nombre d'intervalles est fixé par l'utilisateur. Un exemple est donné dans la figure 26; l'axe des abscisses représente les intervalles de l'attribut "âge" et l'axe des ordonnées représente les intervalles de l'attribut "salaire". Une unité correspond à la case formée par un intervalle de chaque attribut (voir unité u dans la figure 26). Dans un espace d'une seule dimension, l'unité est formée par un seul intervalle.

Chaque unité est caractérisée par la mesure *Sélectivité*. La sélectivité d'une unité u ($Selectivity(u)$) correspond à la fraction de données contenues dans u . Si la sélectivité d'une région u est supérieure à un seuil τ défini par l'utilisateur, alors u est une unité dense.

La technique d'Apriori est utilisée pour trouver les unités denses dans les différents sous-espaces. Les étapes sont résumées comme suit :

1. déterminer les unités denses de dimension 1 (c'est à dire en prenant chaque attribut à part) en accédant à la base de données et en calculant la sélectivité de chaque unité.
2. à partir des unités denses de dimension $(k - 1)$, déterminer les unités candidates de dimension k en utilisant la procédure d'auto-jointure de Apriori. De plus, comme l'algorithme Apriori, l'ensemble des unités candidates sont analysées et toutes les unités qui possèdent au moins une projection à l'espace $(k - 1)$ qui n'est pas dense sont écartées.

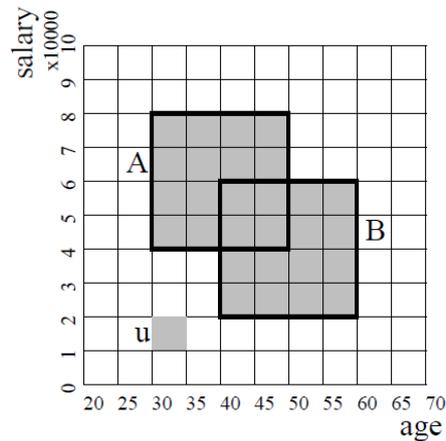


Figure 26 — Identification de *clusters* [Agrawal et al., 1999](#)

3. calculer la sélectivité des unités candidates, en traversant une nouvelle fois la base de données.
4. déterminer les unités denses de dimension k (celles dont la sélectivité est supérieure à τ).
5. répéter l'étape 2 jusqu'à ce qu'aucune unité candidate ne peut être générée.

Le concept de base de l'algorithme CLIQUE consiste donc à détecter les sous-espaces qui contiennent des unités denses en données. Puis de calculer la couverture de ces sous-espaces ; la couverture est définie comme étant la fraction des données qui sont couvertes par les unités denses dans le sous-espace en question. Ensuite, les sous-espaces qui possèdent les couvertures les plus importantes sont gardés.

Après avoir identifié les sous-espaces pertinents, les clusters de chaque sous-espace sont construits. Pour cela, l'algorithme CLIQUE cherche, pour chaque sous-espace, les unités denses qui sont adjacentes et combine ces unités pour former un *cluster*. La figure [26](#) représente un exemple d'application de l'algorithme CLIQUE. Les petites cases grises présentent des unités denses (la case u par exemple). Dans le sous-espace formé par les attributs "âge" et "salaire", deux clusters A et B sont identifiés par la combinaison des unités denses adjacentes.

3.2 Méthode d'apprentissage supervisée par regroupement d'attributs : la méthode SIFRA

Dans le cadre de cette thèse, nous proposons une méthode d'apprentissage supervisée qui se base d'une part sur la génération de règles floues et qui utilise d'autre part le concept des ensembles d'apprentissage artificiel. Notre méthode relève des méthodes d'ensemble qui décomposent les attributs d'entrée en des sous-groupes d'attributs. Elle traite chaque

sous-groupe par un système d'apprentissage supervisé à base de règles floues. Comme la méthode SIFCO [Sousa et al., 2013], notre méthode se compose de trois grandes phases (voir Fig. 27) : la phase de regroupement d'attributs, la phase d'apprentissage et la phase de classification. La phase de regroupement d'attributs est une phase de pré-traitement des données durant laquelle les données qui seront traitées ensemble sont déterminées. La phase d'apprentissage consiste à générer les règles floues ; une base de règles locale est construite pour chaque sous-ensemble de données et les différentes bases locales sont réunies pour former la base de règles globale. Finalement, durant la phase de classification, la base de règles floues générée précédemment est utilisée par un moteur d'inférence pour décider de la classe à attribuer à une nouvelle observation.

Notre contribution réside au niveau de la phase de regroupement d'attributs (la première phase) ; nous proposons une nouvelle méthode de regroupement d'attributs basée sur le concept des Règles d'Association. Notre idée consiste à utiliser le concept de génération des itemsets fréquents afin de déterminer des associations intéressantes entre les attributs numériques. Nous expliquons, ci-dessous, les étapes générales de chaque phase de notre méthode SIFRA. Une description détaillée de la nouvelle méthode de regroupement d'attributs est fournie dans la section 3.3.

3.2.1 Phase de regroupement d'attributs

Généralement, les algorithmes de génération des itemsets fréquents sont utilisés pour déterminer les associations intéressantes entre des variables binaires. Dans le cas où des données numériques sont utilisées, une discrétisation des variables numériques en des intervalles est d'abord requise, puis les associations entre ces intervalles sont détectées à l'aide d'un algorithme de génération des itemsets fréquents. Cependant, le but de notre travail est de détecter les associations entre les attributs numériques et non pas entre leurs intervalles. Pour cela, nous nous sommes inspirés des travaux de [Agrawal et al., 1999] où l'algorithme CLIQUE a été proposé. L'algorithme CLIQUE cherche les attributs qui forment ensemble un sous-espace pertinent pour le *Clustering* de données, et ce en analysant les unités denses qui y sont contenues. Pour cela, les attributs numériques sont partitionnés en des intervalles de longueur égale et une procédure basée sur la technique de l'algorithme Apriori est utilisée pour déterminer toutes les unités denses en passant d'un sous-espace de taille k à un sous-espace de taille $k + 1$, et ce en utilisant la propriété d'anti-monotonie d'Apriori. Ensuite, la couverture du sous-espace (qui correspond à la somme des densités des différentes unités détectées) est calculée. Les sous-espaces qui possèdent une couverture importante sont gardés et considérés comme des sous-espaces intéressants pour le *Clustering*.

Nous utilisons la même idée de détection des régions denses dans des sous-espaces d'attributs et de sommation de leurs densités ; nous adoptons cette idée dans le cadre des problèmes d'apprentissage supervisé à base de règles floues pour former les groupes d'attributs liés.

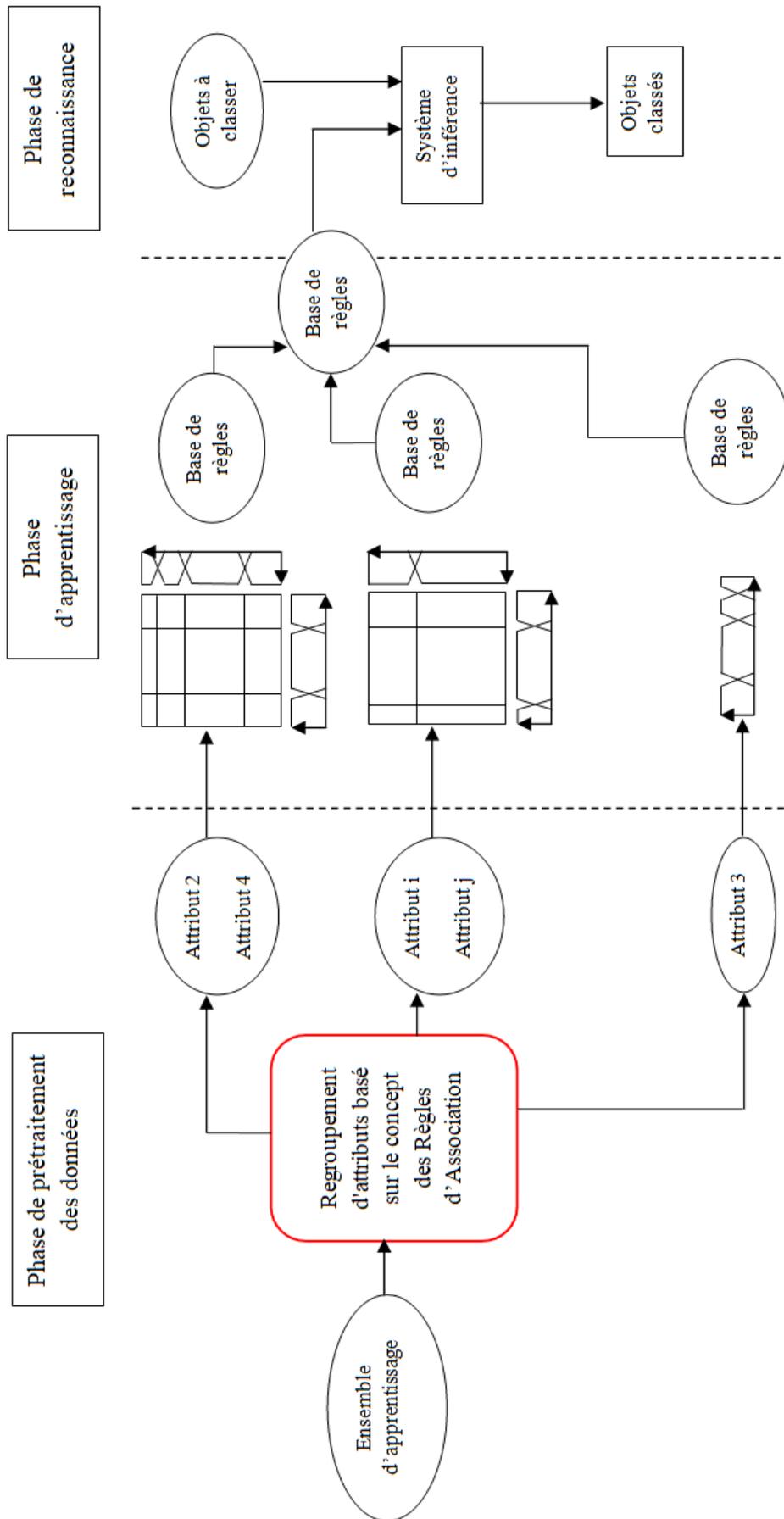


Figure 27 — Étapes de la méthode SIFRA

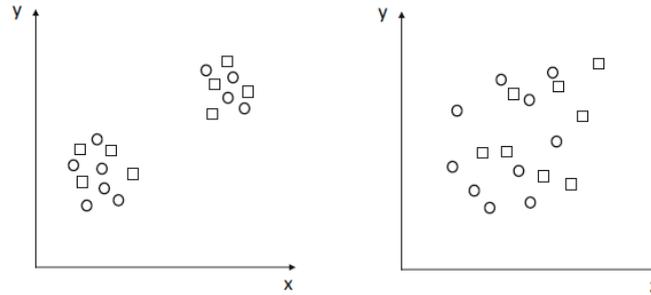


Figure 28 — *Subspace Clustering* : deux *clusters* détectés dans le sous-espace constitué par les attributs X et Y, aucun cluster détecté dans le sous-espace constitué par les attributs Y et Z

En effet, les groupes d'attributs qui vont être associés dans un classifieur apparaîtront dans la prémisse d'une même règle floue. La figure 28 présente un exemple de sous-espace $\{X, Y\}$ détecté par la méthode *Subspace Clustering* en comparaison avec un sous-espace sans aucun cluster pour les attributs Y et Z. Nous pensons que les règles floues qui sont construites en utilisant le groupe d'attributs $\{X, Y\}$ peuvent être plus efficaces et intéressantes que celles générées avec le groupe d'attributs $\{Y, Z\}$ puisque les données représentées dans le premier sous-espace partagent des caractéristiques similaires et peuvent aider à la prédiction de nouveaux objets.

En se basant sur les travaux de [Agrawal et al., 1999](#), nous proposons d'utiliser le concept des Règles d'Association et plus précisément de la génération des itemsets fréquents pour détecter la co-occurrence des données dans les sous-espaces d'attributs. Dans la méthode que nous proposons, au lieu de développer des étapes en suivant la technique de Apriori, nous utilisons directement un algorithme de génération des itemsets fréquents (Apriori ou autre algorithme) pour détecter les régions denses. Les régions denses correspondent bien aux itemsets fréquents générés, et la sélectivité d'une région revient au support de l'itemset fréquent correspondant. Ces régions denses seront par la suite utilisées pour déduire les groupes d'attributs liés. Un degré d'association β est défini pour chaque groupe d'attributs ; ce degré d'association remplace la notion de couverture d'un sous-espace dans l'algorithme CLIQUE. Finalement, nous ajoutons une étape de sélection des groupes d'attributs liés puisque le nombre de groupes qui peuvent être détectés par cette méthode peut être important. Nous décrivons, en détails dans la section 3.3, les différentes étapes de l'approche proposée.

3.2.2 Phase d'apprentissage

Notre méthode SIFRA est une méthode d'ensemble qui combine plusieurs apprenants ; chaque apprenant utilise une projection différente de données pour former une base de règles floues de tailles et de prémisses différentes. Chaque apprenant aura, comme espace des entrées, un des groupes d'attributs liés qui sont générés lors de l'étape précédente. Les différents apprenants utilisent la même méthode d'apprentissage pour construire

leurs bases de règles locales. La méthode d'apprentissage à base de grille floue est utilisée dans le cadre de ce travail [Ishibuchi et al., 1992, Ishibuchi and Yamamoto, 2005]. Nous adoptons la partition floue supervisée vu qu'elle a prouvé son efficacité dans les travaux de [Soua et al., 2013]. La discrétisation supervisée MDLP de Fayyad et Irani [Fayyad and Irani, 1993] est utilisée pour partitionner les attributs numériques et la fonction d'appartenance trapézoïdale est utilisée pour définir les sous-ensembles flous. En utilisant cette méthode de discrétisation supervisée, des partitions de taille différentes peuvent être obtenues pour chaque attribut. Ainsi, des grilles floues rectangulaires sont obtenues pour chaque groupe d'attributs (voir Figure 27). Dans la figure 27, un exemple de grille floue rectangulaire est obtenu pour le groupe d'attributs $\{X_2, X_4\}$ où trois sous-ensembles flous sont définis pour l'attribut X_2 et quatre sous-ensembles flous pour l'attribut X_4 .

Après avoir partitionné l'espace des entrées en des sous-espaces flous en utilisant la partition floue supervisée, nous avons utilisé la méthode heuristique d'Ishibuchi [Ishibuchi et al., 1999, Ishibuchi et al., 1992, Ishibuchi et al., 1994] pour construire une règle floue pour chaque sous-espace (voir section 1.3.1.2). Les équations 7 et 8 de la section 1.3.1.2 ont été utilisées pour calculer le degré de certitude des règles floues. D'autres méthodes heuristiques pour le calcul du degré de certitude peuvent être utilisées dans de prochains travaux, notamment celles basées sur la mesure Confiance [Ishibuchi and Yamamoto, 2005].

3.2.3 Phase de classification

Les bases de règles locales générées par les différents apprenants sont combinées pour former une base de règles globale. Cette base globale est utilisée pour classifier de nouvelles données de classes inconnues. La méthode d'inférence floue utilisée dans cette thèse est la méthode de la seule règle gagnante (voir section 1.3.2.1). Nous avons choisi cette méthode afin de garder les mêmes méthodes d'apprentissage et de classification utilisées dans SIFCO [Soua et al., 2013]. Cela nous permet de comparer notre méthode de regroupement d'attributs avec celle de SIFCO. Cependant, comme les règles floues sont construites sur des projections différentes des données et puisque notre méthode combine les opinions de plusieurs apprenants, l'utilisation de la méthode de vote pondéré (section 1.3.2.2) comme méthode d'inférence floue pourrait être envisagée dans de prochains travaux.

3.3 Une nouvelle méthode de regroupement d'attributs basée sur les Règles d'Association

Notre première contribution dans cette thèse consiste à proposer une nouvelle méthode de regroupement d'attributs en se basant sur le concept des RA. Cette méthode sera capable de trouver des associations intéressantes entre les attributs ; ces associations peuvent être de différents types et pas seulement de type corrélation linéaire. L'utilisation d'un

algorithmes de génération des itemsets fréquents ne permet pas de déterminer directement les attributs liés. Pour cela, des étapes supplémentaires sont développées et décrites dans cette section. Notre méthode de regroupement d'attributs contient trois phases principales (voir Figure 29) : une phase de génération des itemsets fréquents, une phase de détermination des attributs liés, et une dernière phase pour la sélection des groupes d'attributs finaux. Par ailleurs, comme les données traitées dans les SCBRF sont quantitatives, une phase de discrétisation est requise afin de transformer les variables numériques en des valeurs booléennes et pouvoir appliquer par la suite l'algorithme de génération des itemsets fréquents. Il convient de noter que d'autres solutions ont été proposées dans la littérature pour appliquer les RA sur des données numériques sans passer par la phase de discrétisation. Une étude et discussion de ces méthodes est fournie plus loin dans ce chapitre (dans la section 3.4). Les différentes étapes de notre méthode de regroupement d'attributs par

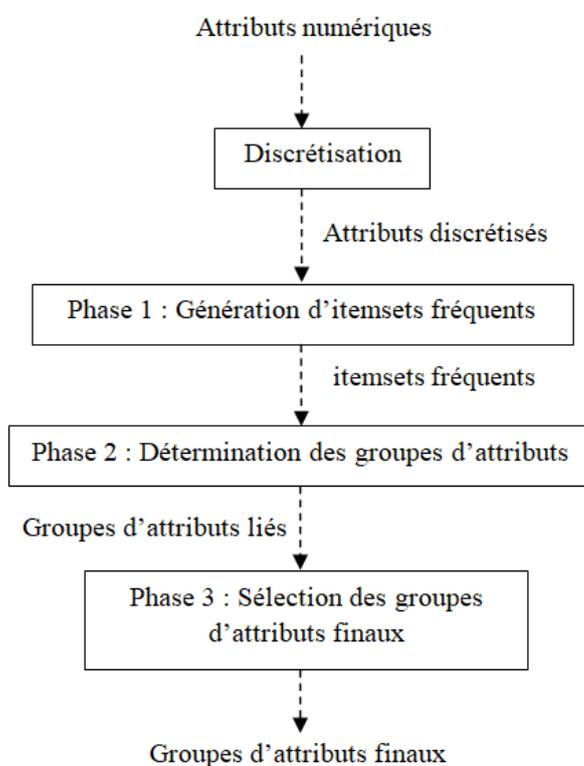


Figure 29 — Étapes de la méthode de regroupement d'attributs de SIFRA

RA sont décrites dans ce qui suit. Nous terminons ensuite cette partie par la présentation d'un exemple complet afin de mieux illustrer la méthode proposée.

3.3.1 Étape 1 : Génération des itemsets fréquents

L'existence d'une liaison entre deux variables dépend de la réponse à la question : est-ce que la connaissance des valeurs de l'une permet de prédire les valeurs de l'autre ? Le concept des RA répond à cette question en associant les valeurs qui apparaissent souvent ensemble dans les transactions de la base de données considérée.

Dans ce travail, nous nous intéressons seulement au premier sous-problème des RA qui consiste en la génération des itemsets fréquents. Pour déterminer les itemsets fréquents, plusieurs algorithmes ont été proposés dans la littérature, à savoir AIS [Agrawal et al., 1993], Apriori [Agrawal and Srikant, 1994], FP-Growth [Han et al., 2000], Eclat [Zaki et al., 1997], etc. Nous adoptons dans cette partie l'algorithme Apriori [Agrawal and Srikant, 1994] étant donné que c'est l'un des algorithmes les plus connus et les plus utilisés. L'utilisation d'autres algorithmes est discutée dans la partie 3.6 de ce chapitre.

Dans ce travail, les attributs sont continus; nous recourons alors au codage disjonctif complet des attributs. Nous rappelons que cette idée a été proposée dans [Srikant and Agrawal, 1996] pour mapper le problème des RA quantitatives au problème classique des RA booléennes et pouvoir appliquer n'importe quel algorithme de génération d'itemsets fréquents (voir section 1.2.2.2). En utilisant une méthode de discrétisation régulière à intervalles égaux (EWI), les attributs numériques sont partitionnés en des intervalles que nous assimilons à des valeurs booléennes. Plusieurs autres méthodes de discrétisation peuvent être appliquées; nous discutons dans le paragraphe 3.4 quelques méthodes possibles. L'algorithme Apriori est ensuite appliqué sur les intervalles des attributs discrétisés et des itemsets fréquents sont ainsi obtenus : les itemsets dont le support dépasse un seuil *minsupp* prédéfini par l'utilisateur. Les itemsets fréquents obtenus correspondent à l'ensemble des intervalles liés.

Nous mentionnons que chaque itemset fréquent découvert par Apriori (ou autre algorithme de génération des itemsets fréquents) est composé par au plus un intervalle de chaque attribut; puisque les intervalles sont issus d'une discrétisation régulière et aucun chevauchement n'existe entre les intervalles, il est alors impossible de trouver un itemset fréquent qui possède deux intervalles d'un même attribut (car cet itemset, s'il existe, possède un support nul : un exemple ne peut pas avoir deux valeurs différentes pour un attribut donné). Pour cela, les itemsets qui peuvent être découverts sont de longueur N au maximum (lorsque le nombre total d'attributs est N). Des 1-itemsets, 2-itemsets, ..., N -itemsets sont ainsi obtenus.

3.3.2 Étape 2 : Détermination des attributs liés

Dans l'étape précédente, nous avons déterminé les groupes d'intervalles liés. Notre but étant de faire un regroupement des attributs et non pas de leurs intervalles, nous nous proposons de développer une procédure qui permet de déterminer la liaison entre un groupe d'attributs à partir des liaisons trouvées entre leurs intervalles.

Nous définissons pour cela une grille d'association qui représente les associations entre les valeurs (intervalles) d'un groupe d'attributs. Chaque axe de la grille concerne un attribut. La figure 30 présente 3 exemples de grille avec deux attributs X_1 et X_2 ; X_1 est décomposé en 6 intervalles ($Int_{11}, \dots, Int_{16}$), et X_2 en 5 intervalles ($Int_{21}, \dots, Int_{25}$). Dans cet exemple, les itemsets obtenus à l'issue de l'application de l'algorithme Apriori

concernent des couples d'intervalles des attributs X_1 et X_2 . Quand deux valeurs forment un itemset fréquent, la case correspondant à leur intersection est grisée : on appelle cette case une région liée.

La liaison entre deux intervalles de deux attributs n'entraîne pas forcément la liaison entre les deux attributs puisque d'une part, ces attributs peuvent avoir très peu de régions liées (exemple (2) de la figure 30) et d'autre part, le nombre de données dans ces régions peut être très faible par rapport au nombre total de données (exemple (3) de la figure 30). Pour cela, le nombre de régions liées et la densité de ces régions sont deux facteurs à prendre en compte pour déterminer si des attributs sont liés.

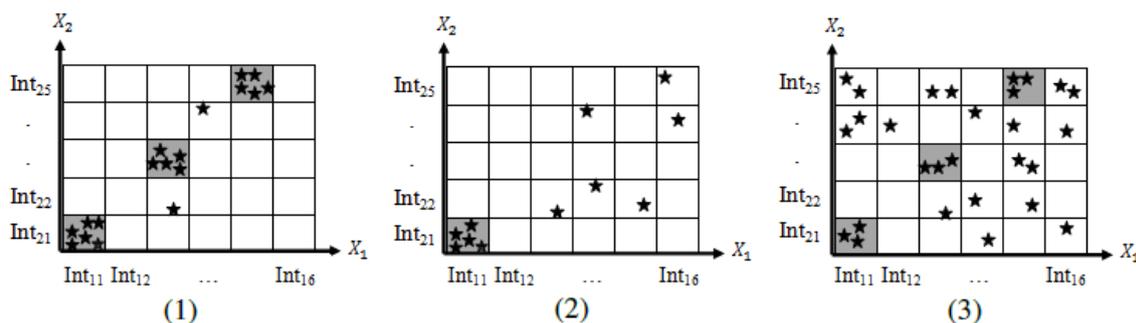


Figure 30 — Exemples de grille d'association

Nous commençons d'abord par définir le poids d'une région, appelé aussi coefficient de pondération. Une première proposition consiste à définir le poids d'une région comme étant la densité des données contenues dans cette région, ce qui est égal au support de l'itemset correspondant. D'autres paramètres sont pris en compte plus tard dans la définition du poids d'une région liée (voir chapitre 4). Le poids de la région U_{ij} formée par les deux intervalles Int_{1i} et Int_{2j} est donc défini par :

$$w(U_{ij}) = \text{support}(Int_{1i} \cup Int_{2j}) = \frac{\text{card}(Int_{1i} \times Int_{2i})}{m} \quad (46)$$

où $\text{card}(Int_{1i} \times Int_{2i})$ est le nombre d'enregistrements qui appartiennent à la région U_{ij} et m est le nombre total d'enregistrements.

Ensuite, à partir des régions liées détectées et de leurs poids, nous déduisons l'association entre chaque groupe d'attributs. Pour cela, nous nous sommes inspirés de l'idée de [Agrawal et al., 1999] qui considère qu'un groupe d'attributs forme un sous-espace intéressant s'il possède une couverture importante en données. La couverture d'un sous-espace est définie par la somme des densités des différentes régions détectées dans ce sous-espace. Dans notre cas, nous considérons qu'un ensemble d'attributs forme un groupe d'attributs liés s'il possède un bon nombre de régions liées et si la somme de leurs poids dépasse un seuil spécifique. Par rapport au travail d'Agrawal, on introduit la notion de poids d'une région liées. Bien que la définition de cette notion de poids revient à la même notion de sélectivité d'une unité dense dans [Agrawal et al., 1999] (puisqu'elle considère aussi la densité de données), elle pourrait aussi prendre en compte d'autres paramètres (comme l'homogénéité d'une région définie dans le chapitre 4).

Enfin, nous définissons le degré d'association d'un groupe d'attributs par la somme des poids des différentes régions liées. Seules les régions liées sont prises en compte ; les densités des autres régions ne sont pas comptabilisées. Ainsi, le degré d'association β correspondant au groupe formé par les deux attributs X_1 et X_2 s'écrit comme suit :

$$\beta_{X_1 X_2} = \sum_{\substack{i=1..k_1 \\ j=1..k_2}} w(U_{ij}) \times r_{ij} \quad (47)$$

avec :

- k_1, k_2 : sont les tailles des partitions respectives des attributs X_1 et X_2
- r_{ij} est le coefficient qui représente la région U_{ij} ; $r_{ij} = \begin{cases} 1 & \text{si } U_{ij} \text{ est une région liée} \\ 0 & \text{sinon} \end{cases}$

Dans le cas général d'un ensemble d'attributs $X = \{X_{n1}, X_{n2}, \dots, X_{nl}\}$, le degré d'association de ces l attributs est :

$$\beta_X = \sum_{\substack{i_1=1..k_1 \\ i_2=1..k_2 \\ \vdots \\ i_l=1..k_l}} w(U_{i_1 i_2 \dots i_l}) \times r_{i_1 i_2 \dots i_l} \quad (48)$$

avec :

- k_1, k_2, \dots, k_l : sont respectivement les tailles des partitions de $X_{n1}, X_{n2}, \dots, X_{nl}$
- $r_{i_1 i_2 \dots i_l}$ est le coefficient qui représente la région $U_{i_1 i_2 \dots i_l}$; $r_{i_1 i_2 \dots i_l} = \begin{cases} 1 & \text{si } U_{i_1 i_2 \dots i_l} \text{ est une région liée} \\ 0 & \text{sinon} \end{cases}$

Finalement, un seuil d'association β_{min} , au delà duquel on considère que les attributs de l'ensemble X sont liés, est défini par l'utilisateur :

- Si $\beta \geq \beta_{min}$ alors les attributs sont associés
- Si $\beta < \beta_{min}$ alors les attributs ne sont pas associés

3.3.3 Étape 3 : Sélection des groupes d'attributs finaux

La détermination des groupes d'attributs associés est basée sur le principe de l'algorithme Apriori ; elle fournit ainsi tous les groupes d'attributs associés de différentes tailles. Il est à noter que ces groupes ne constituent pas forcément une partition de l'ensemble des attributs de départ : chaque attribut peut appartenir à plusieurs groupes en même temps. Cependant, pour garantir la réduction du nombre de règles générées, les groupes d'attributs obtenus doivent former une partition de l'ensemble des attributs de départ, autrement dit, toute intersection ou chevauchement entre les groupes formés doit être évitée (voir équation (36) dans chapitre 2). Pour cette raison, nous proposons de filtrer les groupes d'attributs obtenus afin de supprimer les intersections entre eux. Nous proposons les deux critères de sélection suivants :

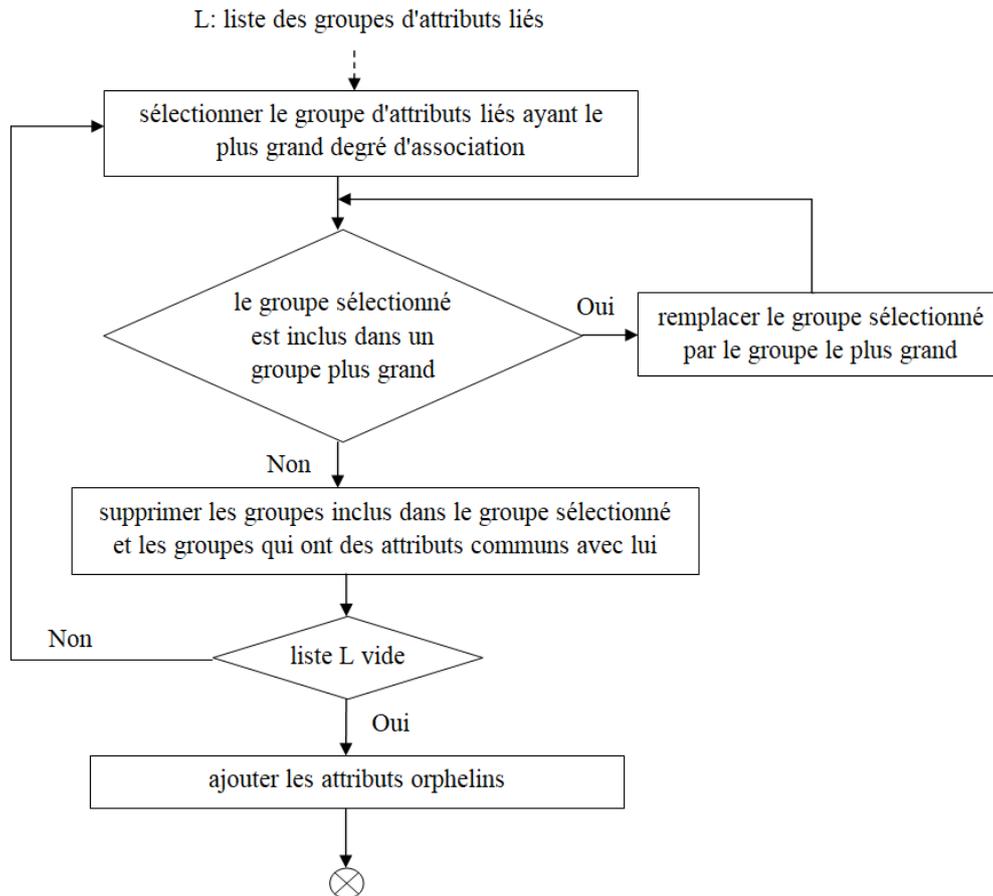


Figure 31 — Étapes de sélection des groupes d'attributs finaux.

1. plus le degré d'association β est grand, plus l'association est forte.
2. les groupes qui contiennent plus d'attributs sont préférés ; les règles avec des prémisses de tailles plus longues sont ainsi obtenues.

Dans ce travail, nous émettons l'hypothèse que les attributs les plus liés sont les plus intéressants et peuvent améliorer la classification. Pour cela nous avons privilégié d'abord les groupes qui possèdent les degrés d'association les plus importants. Dans un deuxième temps, les groupes les plus longs sont aussi privilégiés. En effet, selon [Dehzangi et al., 2007], l'augmentation de la longueur des règles améliore la précision de classification.

Il convient aussi de noter que la construction de groupes d'attributs de longueur importante risque d'augmenter le coût de la génération des règles floues et d'en diminuer l'intelligibilité. En contre partie, le choix des groupes de très petite taille peut produire des règles très générales et diminuer la performance de classification. Afin de mieux expliquer le choix de nos deux critères, une analyse fine de ces critères et de leur effet sur les résultats de notre méthode SIFRA est fournie dans la section 3.5.

Les différentes étapes de sélection des groupes d'attributs finaux sont décrites dans le diagramme de la figure 31 : d'abord, à partir de la liste L de tous les groupes liés, le groupe

d'attributs g_i qui possède le degré d'association le plus important est déterminé. Ensuite on vérifie si ce groupe est inclus dans un autre groupe plus long, dans ce cas, seul le groupe le plus long est gardé. De cette manière, l'association détectée dans le groupe g_i est gardée et d'autres attributs y sont aussi ajoutés. Enfin, pour éliminer tout chevauchement entre les groupes d'attributs, les groupes d'attributs qui sont en intersection avec le groupe choisi sont écartés de la liste L . Ces étapes sont répétées jusqu'à ce qu'aucun groupe d'attributs ne reste dans la liste de départ L .

Il est aussi à noter que, en utilisant un algorithme de génération des itemsets fréquents, les intervalles dont le support est inférieur au seuil *minsupp* sont exclus. Ainsi, certains attributs peuvent ne jamais apparaître et donc ne figurer dans aucun groupe. Ces attributs que nous qualifions d'orphelins, même s'ils ne sont associés à aucun autre attribut, restent utiles pour la phase de classification. Pour cette raison, chacun de ces attributs sera ajouté à la liste des groupes d'attributs finaux comme un attribut indépendant et sera par la suite traité seul par un classifieur.

3.3.4 Exemple illustratif

Afin d'illustrer la méthode de regroupement d'attributs proposée, nous détaillons un exemple concret exécuté sur la fameuse base de données Iris. Le choix de cette base est motivé par sa petite taille et son faible nombre d'attributs ce qui permet une explication plus facile des différentes étapes de la méthode. La base Iris contient 150 exemples d'apprentissage répartis sur 3 classes ; chaque classe désigne un type de la plante Iris (Iris Setosa, Iris Versicolour, Iris Virginica). Les données d'apprentissage sont décrites par 4 attributs numériques :

- sepallength (SL) : la longueur du sépale mesurée en centimètre
- sepalwidth (SW) : la largeur du sépale mesurée en centimètre
- petallength (PL) : la longueur du pétale mesurée en centimètre
- petalwidth (PW) : la largeur du pétale mesurée en centimètre

Comme mentionné dans la description de l'approche de SIFRA, la méthode de regroupement d'attributs contient trois phases principales : la génération d'itemsets fréquents, la détermination des attributs liés et la sélection des groupes d'attributs finaux. Cependant, avant de commencer ces étapes, une discrétisation des attributs numériques en des intervalles est d'abord requise. Dans cet exemple, la discrétisation régulière à 3 intervalles est utilisée pour partitionner les attributs numériques comme suit :

- l'attribut SepalLength est partitionné en 3 intervalles égaux SL1, SL2 et SL3
- l'attribut SepalWidth est partitionné en 3 intervalles SW1, SW2 et SW3

1-itemsets	2-itemsets	3-itemsets	4-itemsets
{SL1} 0.393	{SL1, SW2} 0.246	{SL1, SW2, PL1} 0.24	{SL1, SW2, PL1, PW1} 0.24
{SL2} 0.473	{SL1, PL1} 0.313	{SL1, SW2, PW1} 0.24	
{SW1} 0.313	{SL1, PW1} 0.313	{SL1, PL1, PW1} 0.313	
{SW2} 0.586	{SL2, SW1} 0.206	{SL2, PL2, PW2} 0.22	
{PL1} 0.333	{SL2, SW2} 0.246	{SW2, PL1, PW1} 0.24	
{PL2} 0.36	{SL2, PL2} 0.26		
{PL3} 0.306	{SL2, PW2} 0.253		
{PW1} 0.333	{SL2, PW3} 0.2		
{PW2} 0.36	{SW1, PL2} 0.2		
{PW3} 0.306	{SW1, PW2} 0.206		
	{SW2, PL1} 0.24		
	{SW2, PW1} 0.24		
	{PL1, PW1} 0.333		
	{PL2, PW2} 0.32		
	{PL3, PW3} 0.266		

Tableau 6 — Itemsets fréquents générés par Apriori

- l'attribut PetalLength est partitionné en 3 intervalles PL1, PL2 et PL3
- l'attribut PetalWidth est partitionné en 3 intervalles PW1, PW2 et PW3

Nous décrivons dans ce qui suit les étapes de chacune des 3 phases principales.

Phase 1 : Génération des itemsets fréquents

L'algorithme Apriori est appliqué sur les intervalles issus de la phase de discrétisation précédente. Évidemment, chaque itemset fréquent découvert par Apriori est composé par au plus un intervalle de chaque attribut : 2 intervalles du même attribut ne peuvent pas apparaître dans un même itemset fréquent puisque le support de cet itemset est évidemment égal à 0. En effet, puisqu'aucun chevauchement n'est possible entre les intervalles issus d'une discrétisation régulière, il s'avère impossible de trouver une observation qui contient une valeur appartenant à deux intervalles du même attribut en même temps. Pour cette raison, la longueur maximale d'un itemset découvert par Apriori est égale à 4 (4 attributs numériques donc 4 intervalles associés au maximum).

Ainsi, Apriori génère des itemsets fréquents d'un seul intervalle (1-itemsets), de deux intervalles liés (2-itemsets), de trois intervalles liés (3-itemsets) et de quatre intervalles liés (4-itemsets). Le Tableau 6 représente les itemsets fréquents obtenus dans cet exemple ; chaque itemset est suivi par la valeur de son support. Nous avons choisi la valeur 0.2 pour le seuil *minsupp*.

Phase 2 : Détermination des attributs liés

Les itemsets fréquents obtenus lors de la phase précédente présente les liaisons locales entre les intervalles. À partir de ces liaisons locales, nous déduisons les associations entre

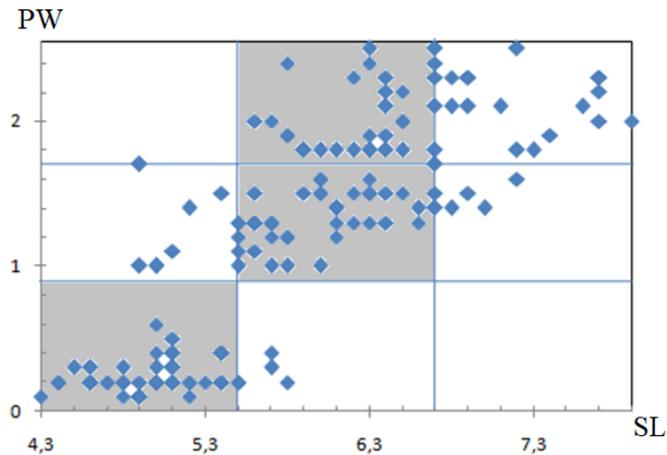


Figure 32 — Grille d'association du groupe d'attributs {SL, PW}

les attributs numériques. Pour cela, nous construisons une grille d'association pour chaque groupe d'attributs potentiel. Prenons par exemple l'itemset {SL1, PW1}, celui-ci contient deux intervalles liés : un intervalle de l'attribut SL et un intervalle de l'attribut PW. Nous cherchons par la suite tous les autres itemsets correspondant au groupe d'attributs {SL, PW}. Ce groupe d'attributs dispose de trois itemsets fréquents : {SL1, PW1}, {SL2, PW2} et {SL2, PW3} ; chaque itemset fréquent correspond à une région liée dans la grille d'association. Ainsi, la grille d'association qui correspond au groupe d'attributs {SL, PW} est représentée dans la figure 32. Cette grille contient 3 régions liées (colorées en gris).

Le poids de chaque région liée correspond à son support. Par conséquent, le degré d'association du groupe d'attributs {SL, PW} est calculé comme suit :

$$\beta_{SL,PW} = \sum_{\substack{i=1..3 \\ j=1..3}} w(U_{ij}) \times r_{ij} = 0.313 \times 1 + 0 + 0 + 0 + 0.253 \times 1 + 0 + 0 + 0.2 \times 1 + 0 = 0.766 \quad (49)$$

Si nous choisissons la valeur 0.7 pour le seuil d'association β_{min} , alors le groupe d'attributs {SL, PW} possède un degré d'association supérieur au seuil β_{min} et il est donc considéré comme un groupe d'attributs liés.

Nous procédons de la même manière pour calculer les degrés d'association des autres groupes d'attributs potentiels et nous ne gardons que les groupes ayant un degré d'association qui dépasse le seuil β_{min} (ici 0.7). Nous obtenons alors les groupes d'attributs liés suivants :

- {SL} avec $\beta = 0.866$
- {SW} avec $\beta = 0.9$
- {PL} avec $\beta = 1$
- {PW} avec $\beta = 1$
- {SL, PW} avec $\beta = 0.766$
- {PL, PW} avec $\beta = 0.919$

Phase 3 : Sélection des groupes d'attributs finaux

Comme mentionné dans la section [3.3.3](#), les groupes d'attributs liés obtenus peuvent contenir des attributs communs. Par exemple, l'attribut PW apparaît dans trois groupes différents : seul dans un groupe {PW}, réuni avec l'attribut SL (dans le groupe {SL, PW}) ou encore associé avec l'attribut PL (dans le groupe {PL, PW}). Cependant, une partition de l'ensemble des attributs de départ est requise. Pour cela, la méthode de sélection des groupes d'attributs finaux que nous avons proposée dans [3.3.3](#) privilégie d'abord les groupes avec le degré d'association le plus grand et ensuite les groupes les plus longs (qui contiennent plus d'attributs).

En se basant sur ces deux critères de sélection et en appliquant les étapes de l'algorithme décrit dans la figure [31](#), nous commençons par sélectionner le premier groupe d'attributs ayant le degré d'association le plus grand : le groupe {PL}. Ensuite, nous vérifions si le groupe {PL} est inclus dans un autre groupe plus grand ; cela nous amène à sélectionner le groupe d'attributs {PL, PW}. Après, nous ajoutons ce groupe d'attributs {PL, PW} à la liste des groupes d'attributs finaux et nous supprimons tous les groupes qui sont en intersection avec lui. Ainsi, les groupes d'attributs {PL}, {PW} et {SL, PW} sont éliminés. À cette étape, les groupes d'attributs restants sont {SL} et {SW}. Nous appliquons les mêmes opérations sur l'ensemble des groupes d'attributs restants. Puisque {SW} possède le degré d'association le plus grand, et puisqu'il n'est inclus dans aucun autre groupe, il est alors ajouté directement à la liste des groupes d'attributs finaux. Finalement, il reste un seul groupe {SL} dans la liste des groupes liés. Ce groupe est aussi ajouté directement à la liste des groupes d'attributs finaux.

Par conséquent, l'ensemble des groupes d'attributs finaux est : {PL, PW}, {SW} et {SL}. Nous vérifions à la fin que tous les attributs figurent dans cet ensemble de groupes et qu'aucun attribut n'est oublié. La phase suivante consiste alors en la phase d'apprentissage où une base de règles floues est construite pour chaque groupe d'attributs.

Discussion

Chaque groupe d'attributs obtenu est traité séparément dans la phase d'apprentissage afin de générer une base de règles locale. Ainsi, pour le premier groupe d'attributs, nous obtenons des règles avec deux attributs dans leurs prémisses (les attributs PL et PW) ; pour le deuxième groupe d'attributs, nous obtenons des règles avec un seul attribut SW ; finalement pour le troisième groupe d'attributs, nous obtenons des règles avec un seul attribut dans leurs prémisses, l'attribut SW.

Ainsi, avec la méthode de génération des règles par grille floue simple [\[Ishibuchi et al., 1992\]](#), la base de règles globale générée est :

```
[[PetalLength, A1], [PetalWidth, A1], 0, 0.9890196940043313]
[[PetalLength, A2], [PetalWidth, A2], 1, 0.6617559469595251]
[[PetalLength, A2], [PetalWidth, A3], 2, 0.7338281380446755]
[[PetalLength, A3], [PetalWidth, A2], 2, 0.4598435090192899]
```

```
[[PetalLength, A3], [PetalWidth, A3], 2, 0.9240297751598812]
[[SepalWidth, A1], 1, 0.3537858756231272]
[[SepalWidth, A2], 2, 0.027932959299344848]
[[SepalWidth, A3], 0, 0.762195081476091]
[[SepalLength, A1], 0, 0.6193724580378023]
[[SepalLength, A2], 1, 0.12816553957260549]
[[SepalLength, A3], 2, 0.6982758722156944]
```

Chaque règle est représentée sous la forme :

[[attribut_₁, N° du sous-ensemble flou], ... , [attribut__k, N° du sous-espace flou], Le numéro de la catégorie de la classe, le degré de certitude CF]

Le 'N° du sous-ensemble flou' résulte de la partition floue utilisée dans la phase d'apprentissage (voir paragraphe [1.3.1.1](#)). Si nous utilisons une partition floue simple de taille 3, le numéro du sous-ensemble flou peut ainsi avoir l'une des valeurs $\{A_1, A_2, A_3\}$. Il est à noter que la taille de la partition floue n'est pas nécessairement égale à la taille de la discrétisation utilisée dans la phase de regroupement.

Dans l'exemple précédent, nous avons trouvé seulement 6 règles floues pour le groupe d'attributs $\{\text{PetalLength}, \text{PetalWidth}\}$ au lieu d'avoir 9 règles ($k^{N_i} = 3^2$; avec k le nombre de sous-ensembles flous et N_i le nombre d'attributs dans le groupe $g_i = \{PL, PW\}$). Ceci est dû au fait qu'aucun exemple d'apprentissage n'existe dans les sous-espaces correspondant aux trois règles manquantes.

3.4 Étude des méthodes de discrétisation

Nous nous focalisons dans ce paragraphe sur la première étape de notre méthode de regroupement d'attributs qui consiste à discrétiser les attributs numériques (voir Fig [29](#)). Il est d'abord à noter qu'il existe deux niveaux différents de discrétisation dans notre système SIFRA :

- Dans la phase de regroupement d'attributs, les attributs numériques doivent être discrétisés en des intervalles afin d'être par la suite mappés en des valeurs booléennes. Cela permet d'appliquer l'algorithme de génération d'itemsets fréquents sur les intervalles obtenus.
- Dans la phase d'apprentissage (paragraphe [3.2.3](#)), la discrétisation est utilisée pour partitionner le domaine des attributs numériques en des sous-ensembles flous utilisés par la suite pour construire la grille floue. La discrétisation floue (régulière ou supervisée) est utilisée à ce niveau : la discrétisation floue ne sépare pas deux intervalles adjacents par un seul point de coupure, cependant, elle utilise une fonction d'appartenance pour définir le degré d'appartenance des données à chaque intervalle (voir figures [51](#) et [52](#) de l'annexe [A](#)).

Nous nous intéressons ici à la discrétisation utilisée dans la phase de regroupement d'attributs. Contrairement à la phase d'apprentissage, nous choisissons dans cette partie une discrétisation classique (non floue) pour ne pas compliquer d'avantage la phase de regroupement d'attributs. Une discrétisation simple en des intervalles distincts est utilisée dans ce travail pour pouvoir appliquer un algorithme de génération d'itemsets fréquents et détecter les régions denses. Néanmoins, une extension de cette méthode à des sous-ensembles flous pourrait être envisagée dans de prochains travaux.

D'autre part, quelque soit le type de la partition utilisée (floue ou classique), plusieurs méthodes de discrétisation peuvent être utilisées dans cette phase. Dans la version basique de SIFRA (présentée ci-dessus), une discrétisation régulière à intervalles égaux a été proposée (3.3.1). D'autres types de discrétisation peuvent toutefois être utilisés, notamment la discrétisation à fréquences égales ou encore la discrétisation supervisée. Un autre type de discrétisation, la discrétisation multi-variée, pourrait aussi être intéressante dans le cadre des problèmes de génération des règles d'association. Cette méthode de discrétisation prend en compte les inter-dépendances entre les attributs (comment ces variables interagissent entre eux) lors de la détermination des points de coupure entre les intervalles. Nous attirons l'attention sur le fait que le choix de la discrétisation a un effet considérable sur le résultat des groupes d'attributs générés. En effet, détecter des régions denses dans un sous-espace nécessite d'avoir choisi auparavant les points de coupure appropriés. Nous présentons dans le chapitre suivant, quelques méthodes de discrétisation qui peuvent être intéressantes pour l'exécution de notre méthode SIFRA.

Il convient aussi de noter que, même si une partition floue est utilisée dans la phase d'apprentissage et une partition simple est proposée dans la phase de regroupement, la même méthode de discrétisation peut néanmoins être utilisée dans les deux phases. Ainsi, les régions denses détectées par la phase de regroupement d'attributs seront les mêmes que les régions construites par la grille floue (les sous-espaces flous) et utilisées pour la génération des règles floues. La méthode de discrétisation supervisée peut dans ce cas être intéressante à utiliser dans les deux niveaux en même temps puisqu'elle prend en compte l'information liée à la classe. Pour confirmer cette hypothèse, nous présentons dans le chapitre 5 des tests expérimentaux réalisés avec la même méthode de discrétisation ou deux méthodes de discrétisation différentes dans les deux phases (la phase de regroupement d'attributs et la phase d'apprentissage).

Finalement, la question suivante peut être posée : est ce que le passage par une phase de discrétisation est indispensable dans notre méthode de regroupement ? Pourquoi ne pas utiliser directement un algorithme de génération d'itemsets fréquents applicable sur des données numériques, comme C-Apriori [Zhou et al., 2013] et Min-Apriori [Han et al., 1997], sans passer par la phase de discrétisation ? La réponse à cette question est évidemment affirmative. Un algorithme de génération d'itemsets fréquents applicable sur des données numériques, tel que C-Apriori et Min-Apriori peut être utilisé sur les attributs numériques dans notre approche. Cependant, nous avons choisi d'appliquer une étape de discrétisation pour les raisons suivantes :

1. L'idée de notre méthode de regroupement d'attributs à base des RA consiste à détecter les régions denses dans des sous-espaces formés par des attributs liés. Pour cela, une discrétisation des attributs en des intervalles est indispensable pour former les régions dans des sous-espaces et pouvoir analyser leurs densités.
2. Les algorithmes tels que C-Apriori et Min-Apriori sont directement applicables sur des données numériques. L'intégration de données qualitatives ou d'autres types de données n'est pas possible. Cependant, bien que les données traitées dans le cadre des SCBRF soient numériques, nous envisageons de construire une méthode de regroupement d'attributs qui pourrait être aussi appliquée dans d'autres domaines d'apprentissage où des données qualitatives peuvent exister. Par exemple, notre méthode de regroupement d'attributs a été utilisée dans le cadre du système SUCRAGE [Borgi, 1999] qui est un système d'apprentissage supervisé non-flou [Zaatour et al., 2016]. La méthode SUCRAGE utilise par défaut une méthode de regroupement d'attributs par recherche de corrélation linéaire, ce qui limite son utilisation à des données numériques. L'utilisation de notre méthode de regroupement d'attributs au lieu de la recherche de corrélation linéaire dans [Zaatour et al., 2016] a permis d'étendre la méthode SUCRAGE à des données aussi bien numériques que qualitatives.

3.5 Analyse des critères de sélection des groupes d'attributs finaux

La dernière phase de la méthode de regroupement de SIFRA consiste à sélectionner un sous-ensemble restreint et compact de groupes d'attributs afin d'éliminer les redondances et les chevauchements entre les groupes d'attributs liés obtenus (voir paragraphe 3.3.3). Rappelons que notre objectif était d'obtenir une partition de l'ensemble des attributs de départ. Nous rappelons aussi que, dans cette phase, nous nous sommes basés sur deux critères principaux pour la sélection : le premier critère concerne le degré d'association β et le deuxième critère concerne la taille ou la longueur des groupes d'attributs obtenus. Ensuite, les attributs qui n'apparaissent pas dans les groupes d'attributs liés sont pris chacun seul et traités indépendamment. Nous visons, dans ce paragraphe, à analyser les différents critères et choix pris pour cette phase. Plus précisément, les questions suivantes sont à évoquer :

- a- Obtenir une partition est-il indispensable ?
- b- Le choix des groupes les plus longs est-il pertinent ?
- c- L'ajout des attributs seuls (qui n'apparaissent dans aucun groupe d'attributs) est-il pertinent ?

Choix de la partition

La réduction du nombre de règles floues n'est garanti que si les groupes d'attributs sélectionnés forment une partition de l'ensemble des attributs de départ (voir section 2.2). Par ailleurs, un grand nombre de groupes d'attributs peut être généré par la méthode de regroupement d'attributs de SIFRA ; la prise en compte de tous ces groupes peut provoquer une explosion du nombre de règles générées. Pour former une partition de l'ensemble des attributs de départ, nous éliminons toutes les redondances et les chevauchements entre les groupes d'attributs liés. Une redondance apparaît lorsqu'un groupe d'attributs est inclus dans un autre groupe (voir exemple (a) ci-dessous). Quant au chevauchement, il concerne l'intersection entre deux groupes d'attributs, c'est à dire lorsqu'un même attribut apparaît dans deux groupes différents (voir exemple (b) ci-dessous).

Exemple (a) : $\{X_1, X_2\}$
 $\{X_1, X_2, X_4\}$

Exemple (b) : $\{X_1, X_2\}$
 $\{X_1, X_3\}$

Les associations redondantes entre les groupes d'attributs peuvent contribuer à la génération de règles redondantes. Selon [Leondes, 2000], une règle peut incorporer une autre règle si elles possèdent la même conclusion mais l'une d'elles contient plus de conditions ; ces deux règles sont considérées comme redondantes. Par exemple, si nous considérons les deux règles R_1 et R_2 ci-dessous (construites à partir de groupes d'attributs de l'exemple (a)), nous notons que la règle R_1 est une généralisation de la règle R_2 . Donc, ces deux règles peuvent être considérées comme redondantes. Par conséquent, le choix de la partition garantit d'une part une réduction du nombre de règles générées et d'autre part l'élimination de toutes les règles redondantes.

R_1 : SI X_1 est faible et X_2 est moyen ALORS y_1 avec $CF = 0.8$

R_2 : SI X_1 est faible et X_2 est moyen et X_4 est faible ALORS y_1 avec $CF=0.95$

Regardons maintenant l'exemple (b), si les degrés d'association des groupes d'attributs $\{X_1, X_2\}$ et $\{X_1, X_3\}$ sont respectivement 0.89 et 0.88, la méthode de sélection va choisir le premier groupe, car il possède le degré β le plus grand, et écarter le deuxième. Cependant, les degrés d'association sont très proches ; une association importante est ainsi écartée. Le problème qui se pose alors consiste à trouver un compromis entre la réduction du nombre de règles qui doit être garantie et la prise en compte des associations importantes. Les critères que nous avons choisi pour la phase de sélection garantissent bien la réduction du nombre de règles. En revanche, des associations qui peuvent être intéressantes sont écartées. Nous verrons aussi, dans le paragraphe suivant, qu'il reste difficile de décider

quel est le groupe le plus important entre deux groupes de tailles différentes ! Nous proposons comme perspective d'utiliser une méthode basée sur les algorithmes génétiques pour sélectionner les groupes d'attributs finaux. En effet, les algorithmes génétiques ont été utilisés dans plusieurs travaux pour sélectionner les règles pertinentes [Alcalá et al., 2007] [Ishibuchi et al., 1997] [Ishibuchi and Yamamoto, 2004]. Les algorithmes génétiques ont été aussi utilisés dans [Cordon et al., 2008] pour sélectionner un ensemble optimal de classifieurs parmi plusieurs classifieurs générés en utilisant les méthodes d'ensemble Bagging et Random subspace. Nous envisageons d'appliquer la même idée de sélection par algorithme génétique, dans de prochains travaux, dans le but de choisir le sous-ensemble optimal des groupes d'attributs liés. Sans passer par une partition, les groupes d'attributs finaux pourront être sélectionnés en gardant un compromis entre un nombre de règles réduit et une bonne performance du système. Ces deux conditions seront à prendre en compte dans la fonction *fitness* de l'algorithme génétique.

Le choix des groupes les plus longs est-il pertinent ?

Nous expliquons dans ce paragraphe les raisons pour lesquelles nous avons choisi de privilégier les groupes d'attributs les plus longs. En effet, la longueur des groupes d'attributs est utilisée comme deuxième critère de sélection après l'importance du degré d'association β . Il faut d'abord attirer l'attention sur le fait que, généralement, plus la longueur des groupes d'attributs augmente plus les valeurs des degrés d'association β diminuent. En effet, en utilisant un algorithme de génération d'itemsets fréquents tel que Apriori, une caractéristique intéressante des itemsets générés est que le support d'un itemset de taille k est toujours inférieur au support de son sous-ensemble de taille $(k - 1)$ [Agrawal et al., 1993]. Cela se généralise au niveau des attributs : la valeur du degré d'association d'un groupe d'attributs de longueur k est généralement inférieure au degré d'association d'un groupe d'attribut qui est un sous-ensemble de celui-là. En prenant en compte cette caractéristique, si l'importance du degré d'association β est le seul critère utilisé lors de la phase de sélection des groupes d'attributs finaux, alors seuls les groupes les plus petits vont être sélectionnés puisqu'ils possèdent les degrés d'association les plus importants. Cela risque d'écarter des associations intéressantes entre les attributs et aussi de produire des règles trop générales. Comme nous visons à détecter les différentes associations possibles entre les attributs et dans le but de ne pas écarter des associations qui peuvent être intéressantes, nous avons choisi de privilégier les groupes les plus longs. En outre, les travaux de [Dehzangi et al., 2007] ont montré que l'augmentation de la longueur des règles peut améliorer la précision de classification.

Le choix des groupes d'attributs les plus longs présentent des avantages, néanmoins il possède aussi des inconvénients.

D'une part, une longueur très importante des groupes d'attributs est à éviter puisque celle-ci alourdit et complique la phase d'apprentissage à base de grille floue. En fait, la taille des groupes d'attributs obtenus dépend principalement du seuil *minsupp* : une

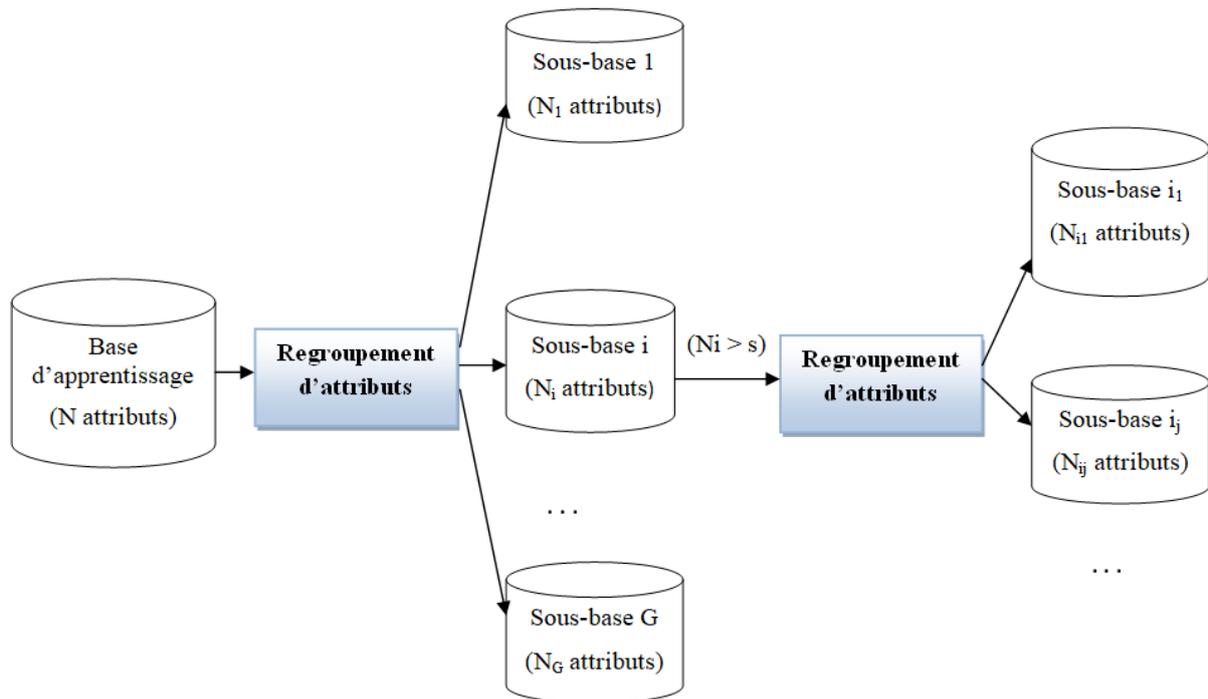


Figure 33 — Regroupement d'attributs récursif

valeur très faible de *minsupp* engendre la génération d'un nombre plus important d'itemsets fréquents avec aussi une taille plus importante des itemsets [Agrawal et al., 1993]. En contre partie, une haute valeur de *minsupp* peut causer la perte de plusieurs associations intéressantes. Donc, le choix du seuil *minsupp* doit être bien étudié et adapté à la taille des bases de données. Une analyse détaillée de l'effet de ce seuil sur les résultats de regroupement d'attributs et de classification est fournie dans le chapitre 5. Si la meilleure valeur de *minsupp* n'est toutefois pas choisie et si un groupe d'attributs avec une longueur importante est généré, une solution possible consiste à réitérer le processus de regroupement d'attributs sur la sous-base en question et re-construire de nouveau des sous-groupes d'attributs pour cette sous-base (voir figure 33). Dans ce cas, il faut absolument augmenter la valeur du seuil *minsupp* pour ne pas obtenir le même groupe d'attributs (tous les attributs liés). Pour appliquer cette idée, il suffit de mettre un seuil s sur le nombre d'attributs réunis dans un même groupe et vérifier, pour chaque groupe d'attributs obtenus, si le nombre d'attributs liés dépasse le seuil s . Si c'est le cas, on refait la méthode de regroupement d'attributs sur ce groupe en question afin de former des groupes plus petits. Nous avons mentionné auparavant que la génération d'une base de règles floues par un SCBRF classique devient coûteuse et difficile dès qu'on dépasse une dizaine d'attributs. Nous pouvons ainsi choisir la valeur 10 pour le seuil s .

D'autre part, nous avons choisi l'importance du degré d'association β comme premier critère de sélection puisque le but de notre méthode de regroupement d'attributs consiste à regrouper en blocs les attributs les plus liés. Ensuite, passer aux groupes d'attributs les plus longs présente notre deuxième critère car cela peut améliorer la précision du

système. Cependant, cette procédure présente des failles qui doivent être étudiées. Prenons comme exemple les groupes d'attributs liés et leurs degrés d'association de l'exemple (c) ci-dessous. Puisque le groupe d'attributs $\{X_1, X_2\}$ possède le degré β le plus grand, il est d'abord sélectionné. Puis, il sera remplacé par le groupe $\{X_1, X_2, X_4\}$ puisque celui-ci englobe le premier. Tous les autres groupes sont supprimés à cause de la procédure de sélection mise en place pour obtenir une partition (Figure 31). Cela garantit que l'association la plus forte $\{X_1, X_2\}$ est prise en compte et ajoute de plus un autre attribut qui est aussi fortement associé aux deux premiers attributs. Cependant, si nous regardons le dernier groupe d'attributs $\{X_1, X_3, X_4\}$, nous remarquons que son degré d'association est plus important que le groupe sélectionné et malgré ça il a été écarté! En voyant ce résultat, le lecteur pourrait se demander pourquoi ne pas commencer la phase de sélection par l'analyse des groupes les plus longs. En effet, les petits groupes d'attributs possèdent généralement les associations les plus fortes et ces groupes d'attributs ne possèdent pas toujours des sur-ensembles. Voyons l'exemple (d), le groupe d'attributs $\{X_1, X_5\}$ possède un degré d'association très important et il n'est inclus dans aucun autre groupe. Dans ce cas, commencer par l'analyse des groupes les plus longs va nous amener à sélectionner le groupe $\{X_1, X_3, X_4\}$, puisqu'il est le groupe le plus long ayant le degré d'association le plus grand, et va par la suite écarter l'association $\{X_1, X_5\}$ puisqu'elle est en intersection avec le groupe sélectionné. Pour conclure, il s'avère nécessaire de mener des tests expérimentaux avec tous ces choix afin de savoir ce que l'on doit privilégier et évaluer la validité des critères que nous avons optés (voir chapitre 5, section 5.3).

Finalement, nous attirons l'attention sur le fait que le degré β caractérise les liaisons entre les attributs, mais il ne reflète pas le pouvoir prédictif du classifieur correspondant à chaque groupe. En d'autres termes, il se peut que le groupe d'attributs qui possède le degré d'association le plus important ne représente pas le meilleur groupe pour la classification : même si ce degré prend en compte la co-occurrence de données dans un sous-espace et détecte les régions denses, il ne reflète pas forcément le pouvoir prédictif de ce groupe par rapport à la classe. Pour remédier à ce problème, une idée intéressante consiste à définir un autre critère de sélection qui mesure la liaison entre les attributs d'un groupe spécifique avec l'attribut *Classe*. Cette idée est présentée dans le chapitre 4.

Exemple (c) :	$\{X_1, X_2\}$	0.9
	$\{X_1, X_3\}$	0.89
	$\{X_1, X_2, X_4\}$	0.84
	$\{X_1, X_3, X_4\}$	0.86
Exemple (d) :	$\{X_1, X_5\}$	0.98
	$\{X_1, X_2\}$	0.9
	$\{X_1, X_3\}$	0.89
	$\{X_1, X_2, X_4\}$	0.84
	$\{X_1, X_3, X_4\}$	0.86

L'ajout des attributs seuls est-il pertinent ?

Nous rappelons que, lors de la génération des itemsets fréquents, des intervalles peuvent n'apparaître dans aucun 1-itemset fréquent lorsqu'ils possèdent des supports faibles (plus petits que *minsupp*). Par ailleurs, des attributs peuvent être "ratés" s'ils ne sont associés à aucun autre attribut ou si les supports de leurs intervalles sont très faibles. Dans notre méthode SIFRA, nous avons choisi d'ajouter ces attributs, que nous avons qualifiés d'orphelins, dans la liste des groupes d'attributs finaux. Chacun de ces attributs est considéré comme indépendant et est traité seul par un classifieur. Cependant, on pourrait penser que le fait que ces attributs ne possèdent aucune région liée et n'apparaissent dans aucun groupe d'attributs liés signifie qu'ils ne sont pas intéressants pour la phase de classification. De ce fait, on pourrait considérer que ces attributs sont non-significatifs et les éliminer de la phase de classification. Dans ce cas, notre méthode SIFRA ne soit plus considérée comme une méthode de regroupement d'attributs seulement mais aussi une méthode de sélection d'attributs qui élimine les attributs non-pertinents.

En revanche, nous attirons l'attention une autre fois sur le fait que la méthode SIFRA cherche les liaisons et les inter-dépendances entre les attributs, mais elle n'évalue pas le pouvoir prédictif de chaque attribut par rapport à la classe. En d'autres termes, bien que les attributs "ratés" ne reflètent pas des caractéristiques communes des données avec d'autres attributs et ne présentent pas des régions denses en données qui peuvent être détectées, mais il se peut qu'ils possèdent un bon pouvoir prédictif par rapport à l'attribut classe. D'autre part, il se peut que le choix de la discrétisation utilisée n'était pas approprié pour cet attribut, c'est pourquoi les données se sont vu éparpillées dans les intervalles construits à travers la méthode de discrétisation adoptée. De ce fait, choisir d'autres méthodes de discrétisation peut modifier toutes les valeurs des supports et des degrés d'association et peut par la suite modifier les groupes d'attributs générés. Une analyse plus détaillée des méthodes de discrétisation appropriées est fournie dans le chapitre suivant.

Pour conclure, un attribut qui n'est pas associé avec d'autres attributs et qu'il n'apparaît dans aucun groupe d'attributs ne signifie pas qu'il est inutile pour la phase de classification. Pour ces raisons, nous avons choisi dans la phase de sélection (partie [3.3.3](#)) d'ajouter ces attributs ratés dans la liste des groupes d'attributs finaux et de traiter chacun indépendamment par un apprenant.

3.6 Analyse du coût de la méthode de regroupement d'attributs

Dans cette partie, nous étudions le coût théorique de la méthode de regroupement d'attributs proposée dans ce chapitre. Une analyse empirique du temps d'exécution est aussi fournie dans le chapitre [5](#). Nous rappelons que l'approche de regroupement d'attributs a été proposée dans cette thèse afin de contourner le problème de la complexité

importante des systèmes à base de règles floues. Cette approche a été proposée comme étant une phase de pré-traitement des données afin de décomposer le problème complexe des SCBRF en des sous-problèmes de complexité inférieure. De ce fait, il faut absolument que la méthode de regroupement d'attributs ne présente pas elle-même des coûts d'exécution très importants. Le but de cette section est donc de dégager les principales limites de cette méthode en terme de complexité et du coût d'exécution et de proposer par la suite des solutions envisageables pour réduire le coût global de cette méthode.

3.6.1 Coût de la méthode de regroupement d'attributs

La méthode de regroupement d'attributs possède trois phases principales ; nous analysons dans cette partie le coût de chaque phase à part.

- Phase de génération des itemsets fréquents

Dans la première version de notre approche SIFRA, nous avons utilisé l'algorithme Apriori comme étant le plus connu des algorithmes de génération d'itemsets fréquents. Cependant, le choix de l'algorithme Apriori n'est toutefois pas approprié pour les bases de grandes dimensions puisqu'il possède un coût important lorsqu'il est appliqué sur les grandes bases [Jiang and Wang, 2011, Li and Shang, 2013, Busarov et al., 2016]. Selon [Tan et al., 2005], les étapes les plus coûteuses de Apriori sont :

- La génération des itemsets candidats : 2^N itemsets candidats peuvent être générés pour une base à N attributs.
- Le calcul des supports des différents itemsets candidats : cette opération nécessite plusieurs accès à la base de données pour balayer les transactions de la base et mettre à jour le support de chaque itemset candidat, et cela pour chaque ensemble de k-itemsets candidats C_k . Supposons que le coût de l'opération de mise à jour du support est de $O(1)$, l'opération de comptage du support coûte $\sum_{k=2}^w m \times |C_k| \simeq O(\sum_{k=2}^w |C_k|)$ où m est le nombre de transactions et w est la longueur maximale des itemsets fréquents obtenus.

Ainsi, le coût de l'algorithme Apriori peut être affectée par les facteurs suivants [Tan et al., 2005] :

- Le nombre d'attributs : plus le nombre d'attributs est grand, plus le nombre d'itemsets candidats générés est grand,
- Le nombre de transactions : puisqu'Apriori fait plusieurs lectures de la base de données pour compter le support de chaque itemset, le temps d'exécution augmente avec l'augmentation du nombre de transactions,

- Le seuil *minsupp* : le choix du seuil *minsupp* est aussi un facteur très important dans la phase de génération d'itemsets fréquents. Le choix d'une faible valeur de *minsupp* provoque la génération d'un nombre important d'itemsets candidats et fréquents et augmente aussi la longueur des itemsets fréquents générés. En revanche, le choix d'une grande valeur de *minsupp* risque d'écarter des associations intéressantes entre les itemsets ou même conduire à un ensemble vide d'itemsets fréquents. Il faut donc choisir la valeur de *minsupp* de manière à trouver un compromis entre la diminution du nombre d'itemsets générés et l'augmentation de la performance de classification. Une méthode de paramétrage automatique peut être suggérée dans de prochains travaux dans le but de fixer d'une manière automatique la valeur optimale du seuil *minsupp* pour le jeu de données utilisé. Un algorithme génétique peut par exemple être proposé pour cela.

- Phase de détermination des attributs liés

La deuxième phase du regroupement d'attributs de SIFRA consiste à déterminer les groupes d'attributs liés en analysant les associations trouvées entre les intervalles. Pour cela, la liste des itemsets fréquents, nommée F et générée lors de la première phase de SIFRA, est parcourue plusieurs fois (au maximum $|F|$ fois). À chaque itération, on cherche dans F les itemsets qui correspondent à un même groupe d'attributs et on calcule leur degré d'association β . Donc, le coût de cette phase est dans le pire des cas $O(|F|^2)$. Ce coût dépend principalement du nombre d'itemsets fréquents ($|F|$) générés par la première phase de SIFRA (phase de génération des itemsets fréquents). Il est à noter qu'un nombre très important d'itemsets fréquents pourrait être généré dans certaines bases de données (où le nombre d'attributs est grand). Par exemple, une base de données peut contenir 40 attributs (ou même plus), dans ce cas, 1.099.511.627.776 (2^{40}) itemsets fréquents peuvent être générés. Traiter ce nombre dans la seconde phase de SIFRA peut évidemment être compliqué et peut affecter considérablement le temps de réponse de notre méthode SIFRA. Pour cela, réduire le nombre d'itemsets fréquents générés présente une solution envisageable pour réduire le coût de cette phase.

- Phase de sélection des groupes d'attributs finaux

La dernière phase du regroupement d'attributs consiste à sélectionner les groupes d'attributs finaux. Cette phase requiert le parcours de la liste des groupes d'attributs liés, nommée L , obtenue par la deuxième phase (phase de détermination des attributs liés). Le parcours de cette liste se fait en plusieurs itérations jusqu'à analyser tous les groupes d'attributs ($|L|$ fois dans le pire des cas). À chaque itération, le parcours de la liste L des groupes d'attributs liés se fait une première fois pour sélectionner le groupe ayant le degré d'association le plus important, une deuxième fois pour vérifier si ce groupe sélectionné est inclus dans un autre groupe plus grand, et une troisième fois pour supprimer les intersections et les chevauchements parmi les groupes d'attributs restants (la liste L est donc parcourue trois fois dans chaque itération). Ainsi, le coût de cette phase est de grandeur $|L| \times 3|L| = 3|L|^2 \simeq O(|L|^2)$ où $|L|$ représente le nombre d'éléments dans la liste

des groupes d'attributs liés.

3.6.2 Quelques solutions pour réduire le coût de la méthode SIFRA

À travers cette analyse, nous notons que la première phase de l'approche de regroupement d'attributs (la phase de génération d'itemsets fréquents) est la plus coûteuse puisque l'algorithme Apriori possède une complexité exponentielle lorsqu'il est appliqué sur des problèmes de grand nombre d'attributs. Le nombre énorme d'itemsets candidats, d'une part, et l'accès multiple à la base de données, d'autre part, alourdit le temps d'exécution de cette phase. À cette fin, le choix d'un algorithme de génération d'itemsets fréquents moins coûteux et plus efficace que Apriori présente une solution intéressante à ce problème. Cela permet de réduire le temps de réponse de la phase de génération des itemsets fréquents. Cependant, le nombre d'itemsets fréquents qui peuvent être générés dans certaines bases de données peut être très important. Cela peut donc alourdir l'exécution de la deuxième phase de SIFRA (phase de détermination des groupes d'attributs liés) puisque le coût de cette dernière dépend principalement de la taille de l'ensemble des itemsets obtenus lors de la phase précédente. Pour cela, il serait intéressant de tenter à réduire ce nombre d'itemsets fréquents en considérant un sous-ensemble compact et suffisant d'itemsets. Ce sous-ensemble peut être obtenu soit en utilisant un algorithme de génération d'itemsets fermés, soit en appliquant un algorithme de génération d'itemsets maximaux.

Une description fine de ces différentes solutions est fournie dans ce qui suit.

- Génération efficace et rapide d'itemsets fréquents

Plusieurs travaux de recherche se sont focalisés sur le développement d'un algorithme de génération d'itemsets fréquents efficace et rapide. Parmi divers algorithmes proposés, nous considérons que FP-Growth présente un choix intéressant pour notre méthode SIFRA. En effet, FP-Growth utilise la fameuse structure de données FP-Tree afin de générer les itemsets fréquents sans passer par la génération d'itemsets candidats et ce en faisant seulement deux accès à la base de données. Par ailleurs, comme mentionné dans la section [1.2.2.3](#), FP-Growth a prouvé sa performance par rapport à Apriori en termes de temps d'exécution dans plusieurs travaux [\[Gyorodi et al., 2004\]](#), [\[Alhamzi et al., 2014\]](#), [\[Garg and Gulia, 2015\]](#). De plus, le fait que FP-Growth ne soit pas influencé par l'augmentation du seuil *minsupp* a aussi motivé notre choix [\[Gyorodi et al., 2004\]](#).

Les différentes étapes de la méthode de regroupement d'attributs sont gardées telles que décrites dans la section [3.3](#). La seule différence réside dans le choix de l'algorithme de génération d'itemsets fréquents dans la première phase de SIFRA (paragraphe [3.3.1](#)) ; l'algorithme FP-Growth est utilisé au lieu de Apriori. Un point important à noter est que le même ensemble d'itemsets fréquents est généré par les deux algorithmes.

- Génération d'itemsets fermés

Les algorithmes de génération d'itemsets fermés génèrent les itemsets fréquents qui ne possèdent aucun sur-ensemble ayant le même support. Ces algorithmes ont été proposés afin d'éviter la génération d'un grand nombre d'itemsets redondants et obtenir un sous-ensemble compact et complet d'itemsets tout en garantissant une meilleure efficacité [Songram, 2012]. En utilisant ces algorithmes, on obtient un nombre réduit d'itemsets avec un coût aussi beaucoup plus réduit. De plus, à travers les itemsets fermés, on peut retrouver tous les itemsets fréquents avec leurs supports [Hadzic et al., 2011]. Donc, le temps de réponse est réduit et aucune information n'est perdue.

Rappelons que la détermination des groupes d'attributs liés (phase 2 décrite dans 3.3.2) se fait en analysant plusieurs fois l'ensemble des itemsets obtenus afin de construire les grilles d'associations et en calculant le degré d'association de chaque groupe d'attributs potentiel. De ce fait, réduire la taille de cet ensemble d'itemsets aide à rendre plus rapide la phase de détermination des attributs liés.

Parmi une diversité d'algorithmes de génération d'itemsets fermés qui ont été proposés dans la littérature, l'algorithme FPclose a été considéré comme l'un des meilleurs algorithmes et a prouvé une haute performance en terme de temps de réponse et de consommation de mémoire. Pour cette raison, nous proposons d'utiliser FPclose dans le cadre de cette thèse pour générer l'ensemble des itemsets fermés. Cet ensemble généré lors de la première phase de SIFRA concerne l'ensemble des intervalles liés, mais non pas tous les intervalles liés. Voyons maintenant l'effet de la génération des itemsets fermés au lieu de tous les itemsets fréquents sur la procédure de détermination des attributs liés (les étapes de la partie 3.3.2).

Nous rappelons que les associations trouvées entre les intervalles sont par la suite utilisées pour déduire les associations globales entre les attributs (qui ont été discrétisés auparavant). Donc, les associations locales qui ont été ratées par l'utilisation de l'algorithme FPclose vont laisser un impact majeur sur les associations globales entre les attributs. Prenons l'exemple de la base Iris présenté dans la section 3.3.4; nous considérons une partie du résultat des itemsets fréquents obtenu dans le tableau 6 afin d'illustrer nos propos. Nous présentons dans l'exemple (e) ci-dessous une partie de l'ensemble des itemsets fréquents obtenus pour la base Iris, alors que l'exemple (f) concerne les itemsets fermés obtenus pour le même ensemble de données. Nous notons que les mêmes itemsets et supports sont obtenus dans les exemples (e) et (f) sauf pour le premier itemset de l'exemple (e); l'itemset $\{SL1, PW1\}$ est absent dans l'ensemble des itemsets fermés (exemple (f)) car il possède un sur-ensemble ($\{SL1, PL1, PW1\}$) qui possède le même support (0.313). Donc, seul l'itemset le plus long a été gardé lors de la génération des itemsets fermés. Maintenant, en comptant le degré d'association β du groupe d'attributs $\{SL, PW\}$ et en supposant que le seuil d'association β_{min} est égal à 0.7, nous observons que ce groupe possède un degré d'association important (supérieur au seuil β_{min}) dans l'exemple (e) c'est à dire lorsque tous les itemsets sont pris en compte, cependant, ce même groupe a "raté" une association importante dans le deuxième exemple (f) ce qui a

réduit son degré d'association et il est par la suite éliminé des groupes d'attributs liés. En outre, son sur-ensemble $\{SL, PL, PW\}$ ne représente pas un groupe d'attributs liés car son degré β est inférieur au seuil β_{min} . Ainsi, l'utilisation des itemsets fermés au lieu des itemsets fréquents peut provoquer la perte de plusieurs associations intéressantes.

Néanmoins, comme mentionné précédemment, à travers les itemsets fermés on peut retrouver tous les itemsets fréquents avec leurs supports et cela en utilisant les supports de leur sur-ensembles [Hadzic et al., 2011]. Dans ce cas, pour chaque groupe d'attributs candidats, nous pouvons compter les supports des associations gardées et retrouver les supports des associations ratées en utilisant les supports de leur sur-ensembles. Cependant, ce traitement supplémentaire, nécessaire pour retrouver tous les itemsets fréquents, peut provoquer un temps de réponse supplémentaire. Des tests expérimentaux, présentés plus tard dans le chapitre 5, seront utilisés pour confirmer cette hypothèse.

Exemple (e) : Tous les itemsets fréquents

$\{SL1, PW1\}$	0.313	}	<i>groupe</i> $\{SL, PW\}$ avec $\beta = 0.766 > \beta_{min}$
$\{SL2, PW2\}$	0.253		
$\{SL2, PW3\}$	0.2		
$\{SL1, PL1, PW1\}$	0.313	}	<i>groupe</i> $\{SL, PL, PW\}$ avec $\beta = 0.533 < \beta_{min}$
$\{SL2, PL2, PW2\}$	0.22		

Exemple (f) : Les itemsets fermés

$\{SL2, PW2\}$	0.253	}	<i>groupe</i> $\{SL, PW\}$ avec $\beta = 0.453 < \beta_{min}$
$\{SL2, PW3\}$	0.2		
$\{SL1, PL1, PW1\}$	0.313	}	<i>groupe</i> $\{SL, PL, PW\}$ avec $\beta = 0.533 < \beta_{min}$
$\{SL2, PL2, PW2\}$	0.22		

Exemple (g) : Les itemsets maximaux

$\{SL2, PW3\}$	0.2	}	<i>groupe</i> $\{SL, PW\}$ avec $\beta = 0.2 < \beta_{min}$
$\{SL1, PL1, PW1\}$	0.313	}	<i>groupe</i> $\{SL, PL, PW\}$ avec $\beta = 0.533 < \beta_{min}$
$\{SL2, PL2, PW2\}$	0.22		

- Génération d'itemsets maximaux

Les algorithmes de génération d'itemsets maximaux cherchent les itemsets fréquents qui ne possèdent aucun sur-ensemble fréquent. Ainsi, seuls les groupes d'attributs de longueurs maximales sont obtenus. Nous prenons le cas du sous-ensemble d'itemsets fréquents de l'exemple (e) ; nous rappelons que ces itemsets présentent une partie de l'ensemble des itemsets fréquents obtenus par Apriori sur la base Iris en utilisant un seuil *minsupp* de valeur 0.2 (Tableau 6). L'application d'un algorithme de génération d'itemsets maximaux sur ce sous-ensemble donne lieu seulement aux groupes d'attributs présentés dans l'exemple

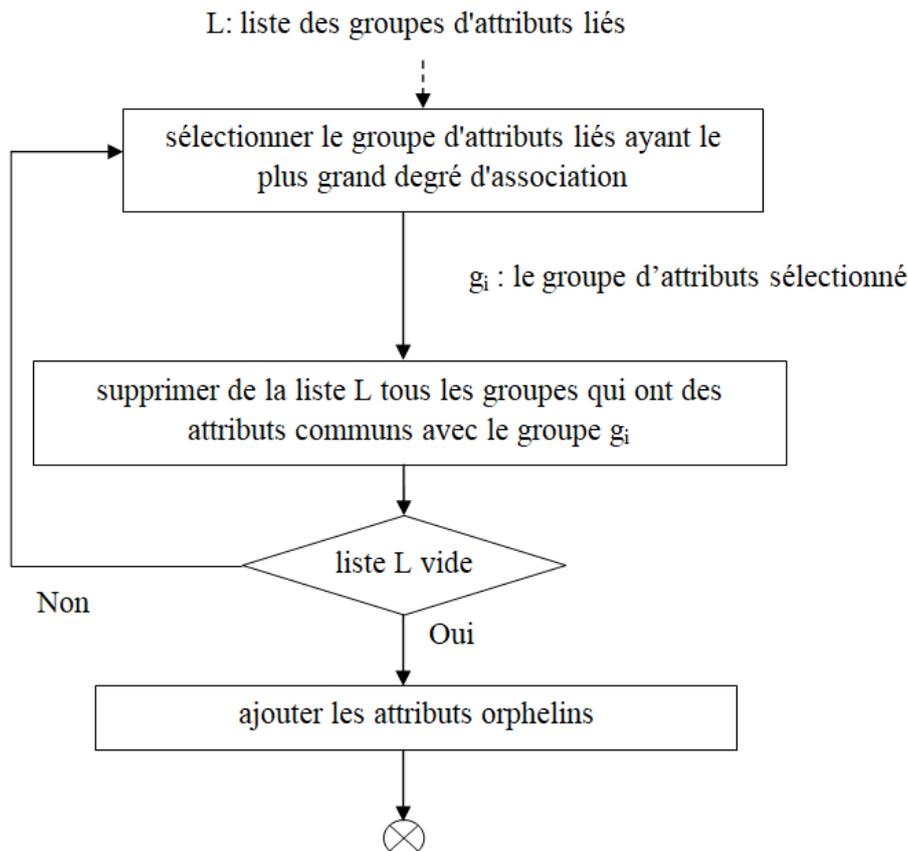


Figure 34 — Sélection des groupes d'attributs finaux dans le cas de la méthode SIFRA avec itemsets maximaux

(g). Les itemsets $\{SL1, PW1\}$ et $\{SL2, PW2\}$ ont été éliminés car ils possèdent des sur-ensembles fréquents (respectivement $\{SL1, PL1, PW1\}$ et $\{SL2, PL2, PW2\}$). Cela peut être intéressant dans notre cas puisqu'on cherche, dans la phase 3 (phase de sélection des groupes d'attributs finaux), à garder les groupes les plus longs. Donc, générer dès le départ les groupes les plus longs réduit la complexité de la troisième phase de filtrage (phase 3 décrite dans la section 3.3.3) : il n'est plus nécessaire de vérifier si un groupe d'attributs liés est inclus dans un autre ; il suffit de choisir le groupe d'attributs avec le degré d'association le plus important et de supprimer par la suite les groupes en intersection avec celui-là. La figure 34 résume ces étapes et remplace dans ce cas ceux de la figure 31.

Pour conclure, l'utilisation des algorithmes de génération d'itemsets maximaux réduit d'une part le coût de la phase de génération d'itemsets, et d'autre part, réduit aussi le coût de la phase de détermination des attributs liés (puisque la taille des itemsets à analyser est réduite), et nous fait donc aussi gagner le coût de la phase de sélection. Cependant, si cette méthode nous aide à choisir directement les groupes d'attributs les plus longs, elle ne donne pas évidemment les mêmes résultats que la première version de SIFRA. Par ailleurs, on risque parfois de rater toutes les associations intéressantes et de ne détecter aucun groupe d'attributs vu que les groupes les plus longs possèdent généralement des degrés d'associations faibles. C'est le cas de l'exemple (g) de la base Iris,

le groupe d'attributs $\{SL, PW\}$ n'est pas considéré comme groupe d'attributs liés parce que son degré d'association devient faible après la suppression des itemsets $\{SL1, PW1\}$ et $\{SL2, PW2\}$. Le groupe d'attributs $\{SL, PL, PW\}$ ne représente pas non plus un groupe d'attributs liés. De plus, il n'est pas possible de retrouver les supports des sous-ensembles des itemsets générés car ils ne possèdent pas nécessairement les mêmes supports. Pour remédier à ce problème, il faut diminuer considérablement la valeur du seuil β_{min} pour ne pas rater les longs itemsets.

Conclusion

Nous avons proposé, dans ce chapitre, une méthode intéressante de regroupement d'attributs qui n'utilise pas des mesures de liaison entre les variables issues de la statistique, mais se base sur des concepts de data mining, précisément sur la génération d'itemsets fréquents. Cette méthode ne cherche pas une forme spécifique d'association entre les attributs, telle que une corrélation linéaire ou une liaison parabolique, etc, mais, elle détecte les sous-espaces d'attributs qui possèdent des régions denses en analysant la co-occurrence des données dans les enregistrements de la base, à travers la génération d'itemsets fréquents. Nous pensons que cette méthode de regroupement d'attributs pourrait être très intéressante surtout pour les bases de données qui ne possèdent pas de corrélations linéaires. Les régions denses détectées à travers cette méthode seront utiles pour la construction des règles floues puisque ces règles sont aussi construites à travers la méthode de grille floue où la prémisse de chaque règle correspond à un sous-espace de la grille (une région floue). En outre, nous pensons que si ces régions denses sont de classes uniformes, c'est à dire qu'elles sont construites en prenant en considération l'information liée à la classe, elles pourraient être beaucoup plus intéressantes pour la phase de classification. Pour cette raison, nous proposons dans le chapitre suivant, une extension de la méthode de regroupement d'attributs de SIFRA qui prend en compte l'information de la classe lors de la phase de regroupement. Cette extension, dite méthode de regroupement supervisé, vise à choisir les groupes d'attributs liés tout en analysant leur pouvoir prédictif.

Extensions de la méthode SIFRA : discrétisations et approches SIFRA-SAR

Introduction

Le présent chapitre porte sur la proposition de deux extensions supervisées de notre méthode de regroupement d'attributs par RA. Le terme supervisé désigne la prise en compte de l'information liée à la classe ; nous proposons de considérer cette information de différentes manières et dans des niveaux différents de la méthode SIFRA. Deux extensions principales ont été proposées dans ce chapitre. D'une part, rappelons qu'une étape de discrétisation est requise dans la méthode de regroupement d'attributs de SIFRA afin de partitionner les attributs numériques en des intervalles et d'appliquer un algorithme de génération d'itemsets fréquents sur ces intervalles. Dans le chapitre précédent, nous avons opté pour une discrétisation régulière simple qui divise les attributs en des intervalles de taille égale. Nous proposons dans ce chapitre une première extension de SIFRA qui consiste à incorporer l'information de la classe dans l'étape de discrétisation des attributs, ce qui revient à utiliser une discrétisation supervisée. D'autre part, nous exposons une deuxième contribution qui consiste à incorporer l'information de la classe, non pas dans l'étape de discrétisation, mais dans la procédure de regroupement d'attributs elle-même. En d'autres termes, la méthode de regroupement d'attributs ne prendra pas en compte seulement les associations existantes entre les attributs, mais aussi les relations des attributs avec l'attribut *Classe*. Cette idée peut être très intéressante dans notre système car il relève d'une méthode d'apprentissage supervisée ; donc prendre en compte l'attribut *Classe* lors de la construction des classifieurs pourrait améliorer les résultats de prédiction.

4.1 SIFRA avec discrétisation supervisée multi-variée

La procédure de discrétisation peut avoir un effet significatif sur le résultat du regroupement d'attributs et sur la précision des classifieurs construits à partir des groupes d'attributs obtenus. En effet, les points de coupure issus de la méthode de discrétisa-

tion affectent la construction des régions denses et par la suite la génération des groupes d'attributs liés. De ce fait, nous présentons dans cette partie une étude des différentes méthodes de discrétisation proposées dans la littérature et nous analysons l'effet de quelques méthodes choisies sur notre approche de regroupement d'attributs.

4.1.1 Méthodes de discrétisation

Certaines méthodes de data mining requièrent un pré-traitement des données afin de s'assurer que celles-ci possèdent un format spécifique requis pour appliquer un algorithme donné. La discrétisation représente une forme de pré-traitement ou de transformation de données, qui consiste à diviser des attributs numériques en des intervalles pour obtenir des variables discrètes requises pour l'application de certains algorithmes de data mining [Moreno et al., 2007]. Plusieurs méthodes de discrétisation ont été proposées dans la littérature. Ces méthodes peuvent être classifiées de différentes manières : supervisées vs non-supervisées, multi-variées vs uni-variées, paramétriques vs non-paramétriques.

La première classification concerne la prise en compte ou pas de l'attribut *Classe* lors du partitionnement des attributs. Quand la discrétisation supervisée considère l'information liée à la classe, les méthodes de discrétisation non-supervisées divisent les attributs en un nombre spécifique d'intervalles quelque soit leur pouvoir discriminant. D'un autre côté, la discrétisation peut être uni-variée ou multi-variée. Les méthodes uni-variées divisent chaque attribut à part, alors que la discrétisation multi-variée construit les intervalles des différents attributs simultanément. Cette dernière considère les inter-dépendances entre les attributs lors de leur partition. Finalement, nous distinguons entre les méthodes de discrétisation paramétriques, qui exigent des paramètres d'entrée fixés par l'utilisateur, comme le nombre d'intervalles requis, et les méthodes non-paramétriques qui utilisent seulement les données traitées et ne nécessitent aucune autre entrée [Yang et al., 2009]. Nous présentons dans la suite quelques méthodes de discrétisation.

4.1.1.1 Discrétisation non-supervisée

Deux méthodes de discrétisation non-supervisée fréquemment utilisées sont la méthode d'intervalles d'amplitudes égales (EWI : Equal-Width Interval) et la méthode d'intervalles de fréquences égales (EFI : Equal-Frequency Interval) [Catlett, 1991]. La première méthode (EWI) divise le domaine de variation d'un attribut en un nombre prédéfini k d'intervalles de largeur égale ; quant à la deuxième (EFI) le divise en un nombre prédéfini d'intervalles contenant le même nombre de valeurs. Ces deux méthodes sont alors paramétriques vu que le nombre d'intervalles doit être prédéfini par l'utilisateur.

4.1.1.2 Discrétisation supervisée

Les méthodes de discrétisation supervisée cherchent à choisir les intervalles qui discriminent au mieux la classe. La mesure d'entropie a été utilisée dans

[Fayyad and Irani, 1993] pour évaluer la discrimination des points de coupure candidats et choisir celui qui minimise l'entropie calculée. Cette méthode de discrétisation, notée MDLP, est une méthode de discrétisation supervisée, non-paramétrique et descendante ; elle partitionne le domaine de variation en deux intervalles en choisissant le meilleur point de coupure, puis re-partitionne chaque intervalle de manière récursive. ChiMerge est une autre méthode de discrétisation supervisée qui est basée sur le critère de Chi-deux [Chimerge, 1992]. Cette méthode utilise le test de Chi-deux pour déterminer si les fréquences de classes dans deux intervalles adjacents sont suffisamment différentes ; si ce n'est pas le cas alors les deux intervalles peuvent être fusionnés. Cette méthode est une méthode ascendante (par fusion) et paramétrique (nécessite de fixer un seuil). Le lecteur intéressé pourra trouver la description d'autres méthodes de discrétisation supervisées dans [Yang et al., 2009].

4.1.1.3 Discrétisation multi-variée

Les méthodes précédemment évoquées sont toutes uni-variées ; elles discrétisent chaque attribut à part indépendamment des autres. Nous décrivons dans ce paragraphe quelques méthodes de discrétisation multi-variées. Cette catégorie présente l'avantage de prendre en considération les inter-dépendances entre les attributs (comment ces variables interagissent entre elles) avant de choisir les points de coupure des intervalles. Prenons comme exemple la pression artérielle de l'être humain [Chao and Li, 2005]. Celle-ci a tendance à augmenter avec l'âge : chez les jeunes, la pression artérielle moyenne est de 120/80 mmHg, alors que chez les personnes âgées elle est de 140/90 mmHg. Par conséquent, il n'est pas approprié de générer un point de coupure tel que 140 mmHg et 90 mmHg pour la pression systolique et la pression diastolique, respectivement. En effet, une pression de 139mmHg/89mmHg chez une personne âgée de 65 ans est considérée comme une tension normale alors que la même pression obtenue pour une personne de 20 ans pourrait être considérée comme une hypertension. Pour conclure, pour discrétiser l'attribut «pression artérielle», il faut évidemment tenir compte de l'attribut «âge». Les méthodes de discrétisation multi-variées sont les seules à prendre en compte ce genre de dépendances entre les variables lors de leur partition.

Cluster-based discretization [Monti and Cooper, 1999] est l'une des premières méthodes de discrétisation multi-variées. Dans cette méthode, les interactions entre les attributs sont analysées pour créer la variable *Cluster* qui représente une pseudo-classe du problème, puis la méthode de discrétisation supervisée de MDLP [Fayyad and Irani, 1993] est utilisée pour former les intervalles des attributs tout en considérant la variable *Cluster* au lieu de la classe d'origine.

La méthode MVD [Bay, 2001] est une autre méthode de discrétisation multi-variée. Dans cette méthode, les attributs sont d'abord partitionnés en des intervalles fins en utilisant une technique de discrétisation simple telle que la discrétisation à intervalles égaux. Puis une fusion des intervalles adjacents est effectuée d'une manière itérative ; les

intervalles fusionnées sont ceux qui représentent des distributions multivariées similaires. L'analyse de la distribution multivariée implique tous les attributs. Cependant, cette approche peut être coûteuse en termes de temps de calcul surtout pour les bases à haute dimension [Mehta et al., 2005].

4.1.1.4 Discrétisation supervisée multi-variée

Une autre catégorie de discrétisation est la discrétisation à la fois supervisée et multi-variée. Ces méthodes bénéficient des avantages des deux types de discrétisation; elles garantissent la prise en compte de l'interdépendance entre les attributs d'une part, et de l'information relative à l'attribut *Classe* d'autre part.

Dans [García et al., 2010], une méthode de discrétisation supervisée multi-variée qui est basée sur le *Clustering* de données, a été proposée. Cette méthode, nommée CBD (Clustering-Based Discretization), utilise l'algorithme k-means pour construire des *clusters* (en évaluant la distance euclidienne entre les données). L'algorithme k-means est mis en oeuvre en considérant tous les attributs en même temps, y compris l'attribut *Classe*, et en donnant plus de poids à ce dernier. Dans le travail de [García et al., 2010], l'attribut *Classe* est une variable numérique, donc incorporer la classe dans l'élaboration des clusters avec l'algorithme k-means en utilisant la distance euclidienne est possible. Cependant, d'autres mesures (autre que la distance euclidienne) doivent être utilisées dans les problèmes où l'attribut *Classe* consiste en une variable qualitative. Afin d'évaluer la méthode de discrétisation CBD, celle-ci a été utilisée avec la méthode de classification associative CMAR et a été comparée à d'autres méthodes de discrétisation, notamment les méthodes EWI, EFI et la méthode supervisée MDLP. Les résultats expérimentaux ont montré une amélioration de la précision de la méthode de classification associative CMAR lorsque la méthode de discrétisation CBD a été utilisée.

Toujours basée sur le *Clustering*, la méthode HyperCluster Finder présente une deuxième méthode de discrétisation supervisée multi-variée [Muhlenbach and Rakotomalala, 2002]. Quand la méthode CBD utilise l'algorithme k-means pour former les *Clusters*, cette méthode utilise le graphe des voisins basé sur la mesure de la distance euclidienne (the Relative Neighborhood Graph of Toussaint [Toussaint, 1980]). Le graphe des voisins est construit en considérant tous les attributs en même temps. Après la génération du graphe des voisins (voir Fig 35-a), les arcs entre 2 individus de classes différentes sont coupés afin d'isoler les groupes (voir Fig 35-b); des *clusters* totalement homogènes sont ainsi créés. Puis, seuls les groupes les plus significatifs, qui possèdent au moins un nombre donné d'individus, sont sélectionnés (ce nombre est, a priori, fixé par l'utilisateur). Enfin, le minimum et maximum de chaque *Cluster* sont projetés sur chaque dimension pour définir les frontières et former les intervalles de chaque attribut.

Une dernière méthode que nous évoquons dans ce paragraphe est la méthode de [Gupta et al., 2010] qui est aussi basée sur le *Clustering* et qui incorpore l'attribut *Classe*

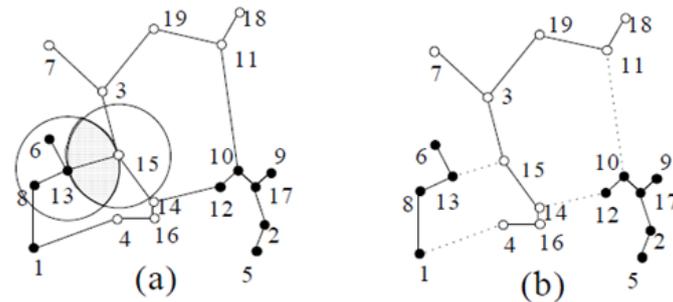


Figure 35 — Graphe de voisins et clusters pour un exemple à deux classes (classe ○ et classe ●) [Muhlenbach and Rakotomalala, 2002]

dans la phase de discrétisation. D’abord, la technique *Clustering* est utilisée pour représenter les interdépendances entre les attributs détectés sous forme de clusters. Deux algorithmes de *Clustering* différents ont été testés : l’algorithme k-means et l’algorithme SNN(Shared Nearest Neighbor clustering). Puis, la discrétisation des attributs est élaborée en prenant en considération en même temps les « clusters » ainsi obtenus et l’information liée à l’attribut *Classe*. En effet, les clusters formés sont considérés comme un nouveau attribut *Classe C’* appelé « pseudo-classe », cette variable donne une information sur le cluster auquel appartient chaque donnée. Le problème devient ainsi un problème à deux attributs : la classe *C* et le pseudo-classe *C’*. Par la suite, la méthode MDLP de Fayyad et Irani [Fayyad and Irani, 1993] est utilisée pour discrétiser les attributs en prenant en considération les deux attributs *Classe* et pseudo-classe (le calcul de l’entropie se fait sur ces deux attributs).

4.1.2 Les méthodes de discrétisation dans le contexte de SIFRA

Nous analysons, dans cette partie, l’adéquation des différentes méthodes de discrétisation évoquées avec notre méthode de regroupement d’attributs. Commençons par la discrétisation régulière à intervalles égaux (EWI) ; elle représente la méthode la plus simple et nécessite le choix du nombre d’intervalles a priori. Il n’est toutefois pas facile de choisir le nombre d’intervalles approprié. En effet, dans le contexte des règles d’association, si la discrétisation régulière est utilisée pour préparer les données numériques, alors le choix du nombre d’intervalles k a un effet considérable sur les règles d’association générées. D’un côté, si une grande valeur de k est choisie, alors de petits intervalles avec très peu d’instances sont construits. Dans ce cas, très peu d’intervalles peuvent aboutir à un support suffisant (qui dépasse le seuil *minsupp*) et un ensemble d’itemsets très réduit est par la suite généré. D’un autre côté, le choix d’une faible valeur de k fournit une partition grossière avec de gros intervalles contenant trop d’instances. Ce deuxième cas extrême peut provoquer la perte de quelques règles intéressantes.

Pour décider quand est ce qu’on doit partitionner un attribut quantitatif et quel nombre d’intervalles doit-on choisir, les auteurs dans [Srikant and Agrawal, 1996] ont dé-

fini une mesure appelée "partial completeness" qui reflète la perte d'information qui peut être provoquée suite à la partition. Dans ce travail, il a été montré que la méthode de discrétisation à fréquences égales (EFI) est plus appropriée pour cette mesure de "partial completeness". Cependant, il a été également noté dans [Srikant and Agrawal, 1996] et dans [Born and Schmidt-Thieme, 2004] que cette méthode de discrétisation ne représente pas toujours le choix optimal et qu'elle n'est pas appropriée pour les données trop asymétriques : elle peut diviser des valeurs adjacentes similaires, pouvant avoir un grand support, en des intervalles différents.

Etant donné que la discrétisation régulière n'est pas toujours appropriée pour notre contexte, nous examinons maintenant les autres catégories de discrétisation : supervisées, multi-variées ou la combinaison des deux types.

Selon [Yang et al., 2009], chaque contexte d'apprentissage possède des caractéristiques différentes et donc exige des stratégies de discrétisation différentes. Notre travail s'inscrit dans deux contextes d'apprentissage différents : l'apprentissage supervisé pour la génération de règles floues de classification et l'apprentissage non-supervisé pour la génération des itemsets fréquents.

- D'une part, notre travail concerne une méthode d'apprentissage supervisée qui procède par la décomposition de l'ensemble des attributs d'entrée en des sous-groupes d'attributs liés. Le résultat de cette méthode étant des règles de classification, donc réunir les attributs qui discriminent au mieux la classe est envisageable. De ce fait, le choix d'une discrétisation supervisée qui prend en compte l'attribut *Classe* lors de la construction des intervalles peut améliorer la précision des règles générées.
- D'autre part, nous utilisons dans ce travail le concept des règles d'association et des algorithmes de génération d'itemsets fréquents pour former les groupes d'attributs liés. Il a été démontré dans plusieurs travaux de la littérature [García et al., 2010] [Moreno et al., 2007] qu'une discrétisation multi-variée qui prend en compte les interdépendances entre les attributs est un choix appropriée compatible avec les problèmes d'extraction des règles d'association. Cette catégorie de discrétisation peut être aussi très intéressante dans notre cas puisque nous cherchons les régions denses dans des sous-espaces d'attributs liés. La discrétisation des attributs a un effet considérable sur la définition des régions denses. Plus la discrétisation est significative et décrit au mieux les interactions entre les attributs, plus les régions détectées auront un sens dans la construction des attributs liés. Par exemple, la figure 36-a représente une discrétisation significative qui a mis en valeur les régions denses dans le sous-espace formé par les attributs X_1 et X_2 . Par contre, une discrétisation régulière à deux intervalles a mis en place des points de coupure arbitraires qui nous font perdre des informations intéressantes sur la manière dont sont distribuées les données dans le sous-espace d'entrée (Figure 36-b).

Pour conclure, nous choisissons d'appliquer une méthode de discrétisation à la fois supervisée et multi-variée dans notre approche de regroupement d'attributs. Les trois

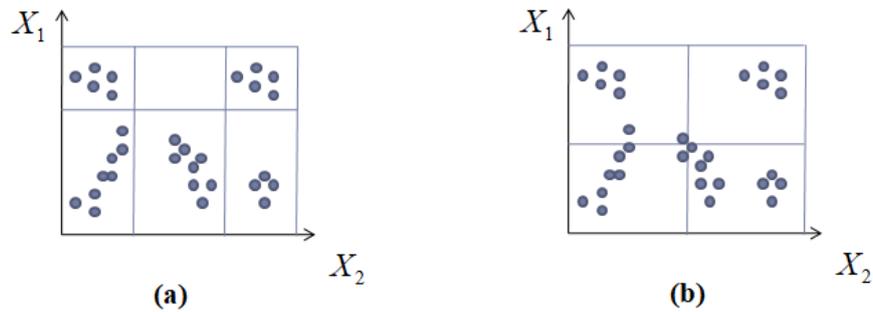


Figure 36 — Discretisation des attributs en rapport avec la distribution des données

méthodes de discrétisation décrites dans le paragraphe 4.1.1.4 sont intéressantes à appliquer dans notre cas. La méthode CBD [García et al., 2010] est simple à implémenter; toutefois, une mesure autre que la mesure de distance euclidienne doit être utilisée puisque notre attribut *Classe* est de type catégoriel. L'algorithme HyperCluster de [Muhlenbach and Rakotomalala, 2002] est aussi intéressant dans notre cas dans la mesure où il peut produire des groupes homogènes (des *clusters* dont les instances sont de même classe). Ces clusters seront utiles dans la formation des liaisons locales de notre grille d'association. Cependant, il risque de produire un nombre très important d'intervalles pour chaque attribut vu que les clusters formés doivent être à 100% homogènes. La méthode "Clustering+MDLP" de [Gupta et al., 2010] pourrait également être appliquée dans notre approche de regroupement d'attributs. Cette méthode présente l'avantage de prendre en compte en même temps l'attribut *Classe* (le côté supervisé de la méthode) et les interdépendances entre les attributs détectés sous forme de *clusters* (le côté multi-varié de la méthode). Nous choisissons d'appliquer cette méthode de discrétisation (la méthode "Clustering+MDLP") dans le contexte de notre méthode de regroupement d'attributs. La simplicité et la disponibilité de l'algorithme k-means et de la méthode de discrétisation MDLP dans les logiciels gratuits a ainsi motivé ce choix. La pertinence de cette méthode appliquée dans le cadre de notre méthode de regroupement d'attributs est analysée dans le chapitre 5 à travers des tests expérimentaux.

4.2 Regroupement d'Attributs Supervisé : les méthodes SIFRA-SAR

Nous avons proposé dans le chapitre précédent une méthode de regroupement d'attributs basée sur le concept des RA et spécifiquement sur la génération d'itemsets fréquents. Cette méthode cherche des associations intéressantes de différents types entre les attributs descriptifs. Dans cette section, nous étendons la méthode de regroupement d'attributs pour déterminer les attributs qui sont d'une part associés entre eux, et d'autre part associés avec l'attribut *Classe*. Cette nouvelle méthode, que nous appelons méthode de regroupement d'attributs supervisé (SAR : Supervised Attributes Regrouping), pro-

fite de l'information fournie par l'attribut *Classe* afin de former les groupes d'attributs appropriés dans l'espoir d'améliorer le pouvoir prédictif du modèle généré.

Nous avons vu que la méthode SIFRA détecte les sous-espaces qui contiennent des régions liées avec une densité importante de données et considère comme liés les attributs qui forment ces sous-espaces. Ces attributs liés sont ensuite regroupés et apparaissent ensemble dans la prémisse d'une même règle. Nous nous focalisons maintenant sur les données qui sont détectées dans les régions denses. Imaginons que les labels de ces données ne sont pas homogènes et que ces données appartiennent à des classes différentes ; dans ce cas, les régions denses détectées peuvent ne pas contribuer efficacement à la tâche de classification. Prenons par exemple le problème présenté dans la figure 37. Dans ce problème, 18 exemples d'apprentissage sont décrits par 3 attributs X_1 , X_2 et X_3 et appartiennent à l'une des 2 classes C_1 et C_2 (C_1 est représentée par les cercles noirs et C_2 est représentée par les triangles blancs). Parmi les 18 exemples d'apprentissage, 8 exemples appartiennent à la classe C_2 et 10 exemples appartiennent à la classe C_1 . La figure 37 représente deux sous-espaces différents : le premier sous-espace est formé par les attributs X_1 et X_2 et le deuxième sous-espace concerne les attributs X_3 et X_1 .

D'après la figure 37, nous pouvons noter que l'utilisation du premier sous-espace (Fig 37-a) a permis de détecter deux régions denses qui contiennent des données qui sont classifiées de manière assez hétérogène. C'est à dire, les données de chaque région dense appartiennent à des classes différentes. Par contre, si nous considérons le deuxième sous-espace (Fig 37-b), nous trouvons deux régions denses dont les données de chacune appartiennent majoritairement à la même classe. Ainsi, nous pensons que les règles floues qui peuvent être construites à partir du deuxième sous-groupe d'attributs (X_3 et X_1) peuvent être plus efficaces que celles générées à partir du premier sous-groupe (X_1 et X_2). En d'autres termes, nous estimons que la qualité des règles floues générées en utilisant un regroupement d'attributs supervisé peut être significativement améliorée.

Donc, l'objectif de notre deuxième approche consiste à privilégier les groupes d'attributs qui sont aussi associés avec l'attribut *Classe*. À cette fin, nous proposons deux approches de regroupement d'attributs supervisé. Chaque approche prend en considération l'information fournie par l'attribut *Classe* de manière différente. Dans la première approche que nous proposons et que nous notons "SAR avec attribut Classe", l'attribut *Classe* est incorporé dans la phase de génération d'itemsets fréquents, c'est à dire qu'il est considéré comme un attribut descriptif et est traité comme les autres attributs lors de la génération des itemsets fréquents. Dans la deuxième approche proposée, nommée "SAR avec facteur d'homogénéité", le degré d'association β ne prend pas en compte seulement la densité des régions liées, mais aussi l'homogénéité de ces régions par rapport à l'attribut *Classe*.

Des descriptions détaillées de ces deux approches "SAR avec attribut Classe" et "SAR avec facteur d'homogénéité" sont fournies dans ce qui suit.

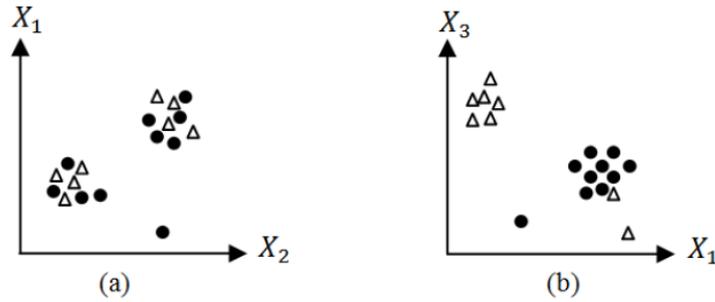


Figure 37 — Sous-espaces détectés avec regroupement d'attributs non-supervisé (a) ou supervisé (b)

4.2.1 Méthode 1 : SAR avec l'attribut Classe

Contrairement à la première méthode proposée (SIFRA), le but du regroupement d'attributs supervisé (SIFRA-SAR) consiste à prendre en compte l'information fournie par l'attribut *Classe* lors de la génération des groupes d'attributs liés. Dans cette partie, nous proposons une première approche SAR qui cherche à générer les itemsets fréquents en considérant tous les attributs y compris l'attribut *Classe* à prédire.

4.2.1.1 Principe

Le principe de cette méthode consiste à incorporer l'attribut *Classe* dans la phase de génération d'itemsets fréquents. Autrement dit, l'algorithme de génération d'itemsets fréquents (l'algorithme Apriori par exemple) est appliqué sur les attributs descriptifs ainsi que sur l'attribut *Classe*. Nous avons vu dans la partie 3.3.1 qu'une partition des attributs numériques en des intervalles est requise. Nous rappelons que l'attribut *Classe* correspond à un attribut qualitatif qui contient des catégories (ou des labels). Comme proposé dans [Srikant and Agrawal, 1996], le codage disjonctif complet est utilisé pour mapper chaque catégorie de la classe à une valeur booléenne. Ainsi, l'algorithme de génération d'itemsets fréquents est appliqué sur les intervalles des attributs descriptifs, ainsi que sur les catégories de l'attribut *Classe*.

Le choix de la méthode de [Srikant and Agrawal, 1996] dans la partie 3.3.1 afin d'appliquer l'algorithme Apriori (ou un autre algorithme de génération d'itemsets fréquents) sur les données numériques se justifie aussi dans cette partie puisque cette méthode peut prendre en compte des attributs aussi bien quantitatifs que qualitatifs. De ce fait, l'ajout de l'attribut *Classe* dans la phase de génération des itemsets fréquents, qui représente une variable qualitative, ne pose pas de problème bien qu'il soit de nature différente des attributs descriptifs.

Vu que la classe est incorporée dans la phase de regroupement d'attributs, nous nommons cette première approche de regroupement d'attributs supervisé "SAR avec Attribut Classe". Nous fournissons une description détaillée des différentes étapes de cette méthode

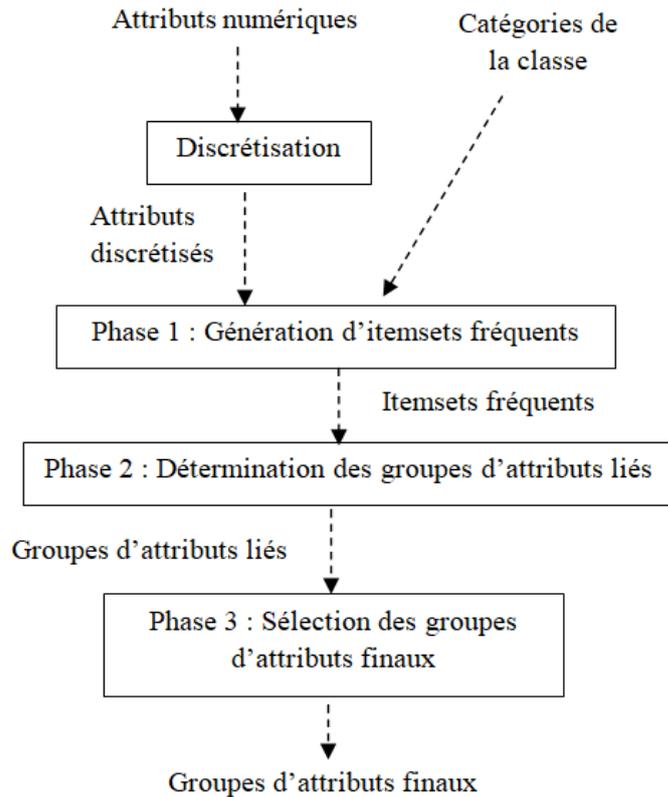


Figure 38 — Diagramme décrivant les étapes de la méthode "SAR avec attribut Classe"

dans le paragraphe suivant.

4.2.1.2 Description de la méthode

Comme la méthode de regroupement d'attributs de SIFRA, la méthode "SAR avec Attribut Classe" contient trois phases :

- Phase 1 : Génération des itemsets fréquents
- Phase 2 : Détermination des attributs liés
- Phase 3 : Sélection des groupes d'attributs finaux

La figure 38 résume les différentes étapes de cette méthode. La première phase de génération d'itemsets fréquents utilise l'algorithme Apriori (ou un autre algorithme de génération d'itemsets fréquents) sur les intervalles des attributs discrétisés et sur les catégories de l'attribut *Classe*. Dans la deuxième phase (phase de détermination des attributs liés), la même procédure décrite dans 3.3.2 est utilisée pour déterminer les groupes d'attributs liés. Néanmoins, le résultat de cette phase est un peu différent ; cette fois-ci, deux types de groupes d'attributs liés sont obtenus :

1. des groupes d'attributs qui contiennent l'attribut *Classe* (l'ensemble S_1) : ces groupes représentent les associations qui sont détectées entre des éléments de l'ensemble {attributs descriptifs, *Classe*}.
2. des groupes d'attributs qui ne contiennent pas l'attribut *Classe* (l'ensemble S_2) : dans ces groupes d'attributs, seules les associations entre les attributs descriptifs sont détectées.

Enfin, dans la troisième phase (phase de sélection des groupes d'attributs finaux), les deux critères de sélection proposés dans [3.3.3](#) sont utilisés mais en ajoutant un troisième critère plus important. Ce critère consiste à privilégier les groupes d'attributs liés qui contiennent l'attribut *Classe* (l'ensemble S_1). En effet, ces groupes d'attributs représentent des associations importantes entre des attributs descriptifs avec l'attribut *Classe*. Ces associations (dites "supervisées") peuvent contribuer à l'amélioration du pouvoir prédictif du modèle de classification.

Les étapes de la procédure de sélection des groupes d'attributs finaux sont décrites comme suit :

1. La procédure de sélection analyse en premier lieu les groupes d'attributs qui contiennent la classe (l'ensemble S_1). Les étapes de l'algorithme présentées dans la figure [39](#) sont appliquées sur l'ensemble des groupes d'attributs liés obtenus lors de la deuxième phase ; un premier ensemble de groupes d'attributs G' est alors formé. L'ensemble G' contient les groupes d'attributs finaux qui sont liés avec l'attribut *Classe*. Nous rappelons que la procédure de sélection garantit qu'aucun chevauchement n'existe entre les groupes d'attributs générés.
2. Les groupes d'attributs liés qui ne contiennent pas la classe (l'ensemble S_2) sont analysés en deuxième lieu. En effet, les attributs qui ne sont pas associés avec la classe peuvent quand même posséder des associations entre eux. Ces associations ne sont pas négligées, mais elles sont prises en compte en deuxième lieu en utilisant les deux critères de sélection classiques de l'approche SIFRA. Cependant, il faut d'abord éliminer de cet ensemble S_2 tous les groupes qui sont en intersection avec les éléments de l'ensemble G' . Puis, les étapes de l'algorithme illustré dans la Fig [31](#) sont appliquées sur les éléments restants de S_2 afin d'obtenir le second ensemble de groupes d'attributs G'' . L'ensemble G'' contient donc des groupes d'attributs qui ne se chevauchent pas et n'ayant aucun attribut en commun avec G' . La figure [40](#) résume ces différentes étapes.
3. Finalement, la réunion de G' et G'' forme l'ensemble des groupes d'attributs finaux.

Pour conclure, la différence majeure entre la méthode SIFRA et la méthode "SAR avec Attribut Classe" se résume en deux points principaux :

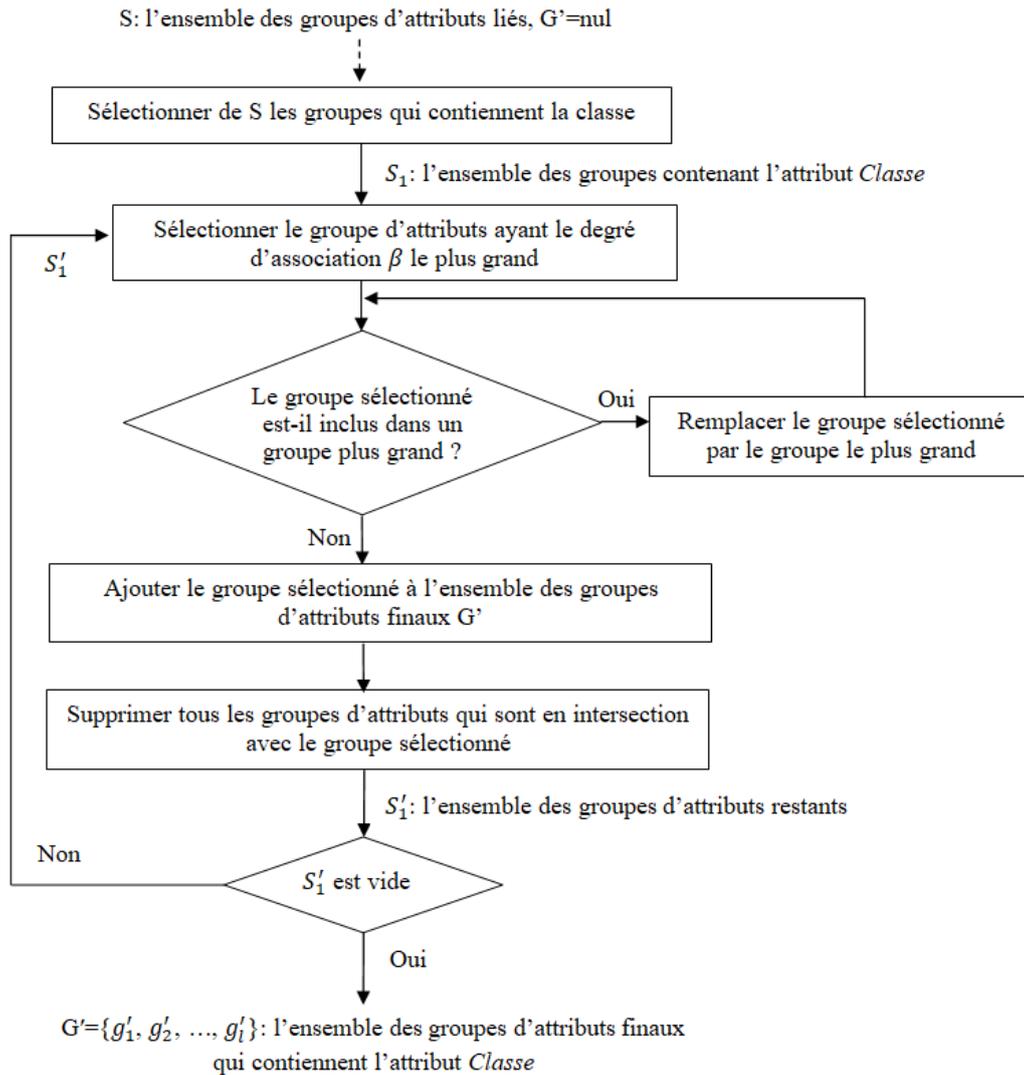


Figure 39 — Étapes de sélection des groupes d'attributs finaux contenant l'attribut *Classe*

- La première phase (génération d'itemsets fréquents) prend en entrée les attributs discrétisés mais aussi les labels de la classe.
- La dernière phase (sélection des groupes d'attributs finaux) prend en compte un nouveau critère qui consiste à privilégier les groupes d'attributs associés avec la classe.

4.2.2 Méthode 2 : SAR avec facteur d'homogénéité

Afin de prendre en considération l'information liée à la classe, nous avons présenté précédemment une première méthode qui consiste à incorporer l'attribut *Classe* dans l'étape de génération des itemsets fréquents. Dans cette section, nous proposons une deuxième méthode de regroupement d'attributs supervisé. Dans cette nouvelle méthode, la phase de génération d'itemsets fréquents reste inchangée (la même que SIFRA); c'est à dire,

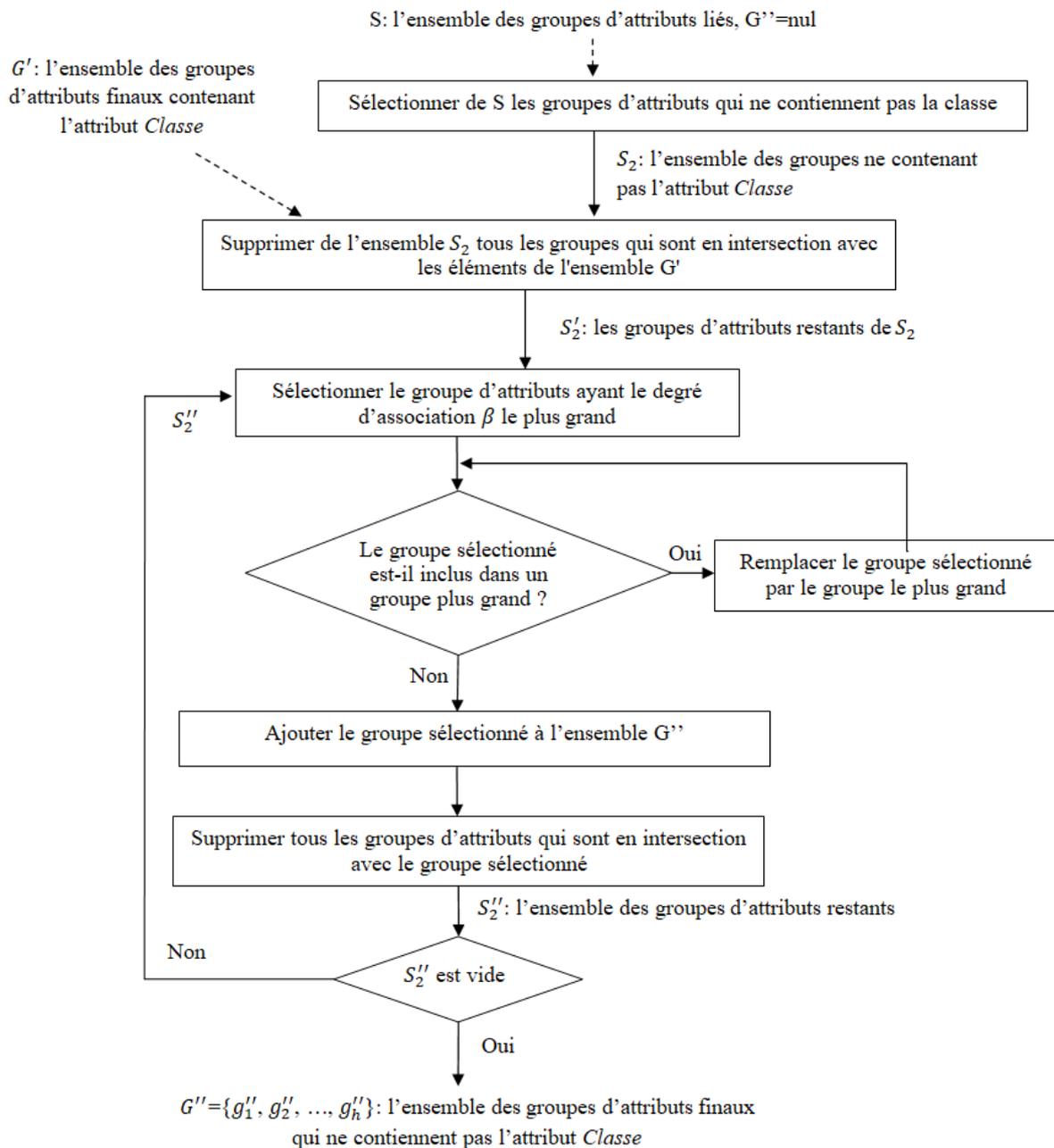


Figure 40 — Étapes de sélection des groupes d'attributs finaux qui ne contiennent pas l'attribut *Classe*

l'attribut *Classe* n'est pas pris en compte lors de la génération des itemsets fréquents. Par contre, l'information liée à la classe est utilisée dans la deuxième phase de SIFRA (phase de détermination des attributs liés) dans le calcul des degrés d'association β . Ainsi, l'approche consiste à renforcer les poids des régions denses qui possèdent une homogénéité importante des données relativement à l'attribut *Classe*.

4.2.2.1 Principe

Dans notre première méthode SIFRA, la définition du degré d'association β prend en compte le nombre des régions liées ainsi que leur densité. Dans cette section, nous proposons de considérer un troisième facteur, celui de l'homogénéité des régions liées. Dans chaque région liée, l'homogénéité des données est calculée par rapport à l'attribut *Classe*. On parle d'une région homogène à 100% lorsque toutes ses données appartiennent à la même classe. Plus l'homogénéité des données par rapport à l'attribut *Classe* augmente, plus le poids de cette région sera important.

4.2.2.2 Les mesures d'homogénéité

Plusieurs mesures ont été définies dans la littérature afin de calculer le degré d'homogénéité d'un ensemble de données par rapport aux valeurs d'un attribut. La mesure d'entropie [Shannon, 1948] et la mesure de Gini [Breiman et al., 1984] sont les plus connues. Dans le cadre de cette thèse, nous nous focalisons sur la mesure d'entropie pour calculer les degrés d'homogénéité. Une étude d'autres mesures sera envisagée dans de prochains travaux.

Soit X une variable aléatoire discrète possédant n valeurs possibles. $P = (p_1, p_2, \dots, p_n)$ est la distribution des probabilités d'occurrence de chaque valeur de X ($\sum_{i=1}^n p_i = 1$). L'entropie de la distribution P , notée $H(P)$, est égale à :

$$H(P) = - \sum_{i=1}^n p_i \times \log_2(p_i) \quad (50)$$

L'entropie possède quelques propriétés :

- $H \geq 0$
- $H = 0$ si et seulement si toutes les p_i sont nulles sauf une seule. C'est à dire, toutes les données appartiennent à une seule valeur de X .
- H est maximale et égale à $\log_2(n)$ lorsque toutes les p_i sont égales. Cela veut dire que la distribution est hétérogène entre les valeurs de X .
- Une valeur faible de l'entropie désigne une haute homogénéité des données et une valeur importante de l'entropie désigne une homogénéité faible.

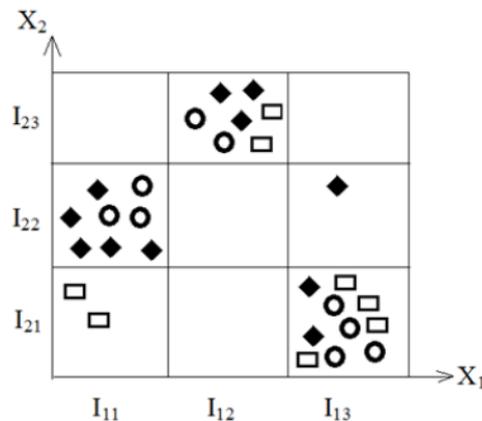


Figure 41 — Distribution des données étiquetées dans la grille d'association

4.2.2.3 Description de la méthode

Nous utilisons la mesure d'entropie pour calculer l'homogénéité d'une région liée, et ce par rapport aux catégories de l'attribut *Classe*. Nous prenons par exemple le cas de la région liée U_{12} (formée par les intervalles I_{11} et I_{22}) présentée dans la figure 41. Sachant que l'attribut *Classe* contient trois catégories C_1 (les cercles blancs), C_2 (les rectangles blancs) et C_3 (les losanges noirs), nous notons que trois exemples de la région U_{12} appartiennent à la classe C_1 , cinq exemples appartiennent à la classe C_3 et aucun exemple n'appartient à la classe C_2 . L'entropie de la région U_{12} est alors calculée comme suit :

$$H(U_{12}) = -3/8 \times \log_2(3/8) - 5/8 \times \log_2(5/8) - 0 = 0.954 \quad (51)$$

Nous avons mentionné dans les propriétés de la mesure d'entropie que celle-ci est toujours positive et que sa valeur maximale est égale à $\log_2(n)$ qui est dans ce cas égale à : $\log_2(3) = 1.5849$. Il faut connaître la valeur maximale de l'entropie pour savoir si la valeur obtenue reflète une haute ou bien une basse homogénéité. D'autre part, les valeurs des degrés d'association β et des poids des régions liées w sont toujours comprises entre 0 et 1. Donc, pour pouvoir utiliser la mesure d'homogénéité dans le calcul du degré d'association β , il faut la normaliser de manière à ce qu'elle soit elle-même comprise entre 0 et 1.

Parmi les méthodes de normalisation existantes, nous utilisons la normalisation définie dans [Stéphane, 2012]. L'entropie normalisée est ainsi définie comme suit :

$$H_{normalized} = \frac{H - H_{min}}{H_{max} - H_{min}} \quad (52)$$

où $H_{min} = 0$ et $H_{max} = \log_2(n)$

Dans ce travail, l'entropie est calculée sur la distribution des données dans les différentes catégories de la classe. Donc, la valeur de n est égale au nombre de labels de la classe qui est égal à C . L'entropie normalisée s'écrit alors comme suit :

$$H_{normalized} = \frac{H - 0}{\log_2(C) - 0} = \frac{H}{\log_2(C)} \quad (53)$$

D'autre part, comme mentionné dans les propriétés de l'entropie, une valeur importante de l'entropie H reflète une faible homogénéité et une valeur faible de l'entropie reflète une homogénéité élevée. Pour cela, nous définissons l'homogénéité d'une région liée U comme étant :

$$Hom(U) = 1 - H_{normalized} = 1 - \frac{H(U)}{\log_2(C)} \quad (54)$$

Nous avons décrit la façon de calculer l'homogénéité d'une région liée. Il nous reste donc à expliquer l'utilisation de cette mesure d'homogénéité dans le calcul du degré d'association β . Nous rappelons que le degré d'association β , comme défini dans notre première approche SIFRA, considère pour chaque région liée son poids w (qui reflète la densité des données). Dans cette nouvelle approche "SAR avec facteur d'homogénéité", une nouvelle définition du degré d'association β est proposée : celle-ci prend en compte aussi les homogénéités des différentes régions liées. Ainsi, le degré d'association d'un groupe d'attributs (par exemple X_1 et X_2) considère les deux paramètres suivants :

1. La somme des densités des différentes régions liées : ce paramètre représente le pourcentage des données couvertes par les régions liées. Ce paramètre, noté $\beta 1$, est égale à l'ancienne définition de β (équation [47](#)) :

$$\beta 1_{X_1 X_2} = \sum_{\substack{i=1..k_1 \\ j=1..k_2}} w(U_{ij}) \times r_{ij} \quad (55)$$

avec :

- k_1, k_2 : sont les tailles des partitions respectives des attributs X_1 et X_2
- r_{ij} est le coefficient qui représente la région U_{ij}

$$r_{ij} = \begin{cases} 1 & \text{si } U_{ij} \text{ est une région liée} \\ 0 & \text{sinon} \end{cases} \quad (56)$$

- $w(U_{ij})$ est le poids de la région U_{ij}

$$w(U_{ij}) = \text{support}(Int_{1i} \cup Int_{2j}) \quad (57)$$

2. Les homogénéités des différentes régions liées : nous avons présenté précédemment la définition de l'homogénéité de chaque région liée. Il faut maintenant procéder à l'agrégation de ces différentes homogénéités pour calculer l'homogénéité globale d'un groupe d'attributs, notée $\beta 2$. Nous proposons d'utiliser deux méthodes d'agrégation possibles :

- (a) β_2 est la moyenne des homogénéités des différentes régions liées du groupe d'attributs correspondant :

$$\beta_{2_{X_1 X_2}} = \frac{1}{nb_{RL}} \times \left(\sum_{\substack{i=1..k_1 \\ j=1..k_2}} Hom(U_{ij}) \times r_{ij} \right) \quad (58)$$

avec nb_{RL} est le nombre de régions liées, calculé comme suit :

$$nb_{RL} = \sum_{\substack{i=1..k_1 \\ j=1..k_2}} r_{ij} \quad (59)$$

- (b) β_2 est égal à la valeur maximale des homogénéités des différentes régions liées :

$$\beta_{2_{X_1 X_2}} = \max_{i,j} (Hom(U_{ij})) \quad (60)$$

où $i \in \{1, k_1\}$ et $j \in \{1, k_2\}$

Si aucune région liée n'existe pour un groupe d'attributs ($nb_{LR} = 0$), alors ni β_1 ni β_2 ne sont calculés ; la détermination du degré d'association n'est nécessaire que pour les groupes qui contiennent au moins une région liée.

Finalement, pour un groupe formé par les attributs X_1 et X_2 , la nouvelle définition du degré d'association est :

$$\beta_{X_1 X_2} = \frac{\beta_1 + \beta_2}{2} \quad (61)$$

La nouvelle définition de β correspond donc à la moyenne des deux facteurs β_1 et β_2 . Une moyenne pondérée peut toutefois être utilisée dans de prochains travaux afin de donner plus de poids à l'un des deux facteurs par rapport à l'autre.

Toutes les étapes et formules décrites dans cette partie se généralisent facilement au cas de plusieurs attributs.

Il est à noter que le facteur d'homogénéité β_2 peut avoir des valeurs très petites par rapport au facteur qui représente les densités β_1 . Dans ce cas, la valeur du degré β pourrait être faible et aucun des groupes d'attributs ne pourrait satisfaire le seuil d'association β_{min} qui est fixé par l'utilisateur. Par exemple, l'utilisateur pourrait choisir la valeur 0.7 du seuil β_{min} et considérer que les groupes ayant des degrés d'association au-delà de ce seuil sont des groupes d'attributs liés. Cependant, les valeurs du facteur d'homogénéité β_2 peuvent parfois ne pas dépasser 0.1 ou 0.2, ce qui affaiblit par la suite les valeurs du degré β . De ce fait, le choix d'un seuil approprié qui soit du même ordre de grandeur que celui des degrés d'association β n'est pas évident et ne présente pas une tâche facile pour l'utilisateur.

Pour cette raison, nous proposons un autre moyen pour définir le seuil β_{min} de manière à ce qu'il soit de la même grandeur que les valeurs obtenues pour les degrés d'association β . Cette méthode consiste à choisir un coefficient de pondération α au lieu de choisir

directement la valeur du seuil β_{min} . Ce coefficient est compris entre 0 et 1 et sera pondéré par la valeur maximale des degrés d'association obtenus.

$$\beta_{min} = \alpha \times \max_{i=1..G} \beta_i \quad (62)$$

avec β_i est le degré d'association du groupe d'attributs g_i et G représente le nombre de groupes d'attributs obtenus.

Le choix d'une grande valeur de α (par exemple 0.8) signifie que l'utilisateur cherche à détecter les groupes d'attributs avec les degrés d'association les plus importants. Alors que le choix d'une faible valeur de α signifie que même les groupes d'attributs avec des faibles associations sont acceptés.

Conclusion

Dans ce chapitre, nous avons proposé plusieurs versions "supervisées" de la méthode de regroupement d'attributs. L'idée principale défendue dans ce travail consiste à incorporer l'information liée à la classe dans la procédure de regroupement d'attributs, en considérant cette information de manières différentes et dans des niveaux différents de la méthode de regroupement. En effet, comme les groupes d'attributs formés sont ultérieurement utilisés pour générer les règles floues de classification, il nous a semblé intéressant d'utiliser cette information supervisée pour choisir les groupes les plus appropriés.

Dans ce contexte, nous avons développé trois extensions supervisées de la méthode SIFRA. La première réside au niveau de la phase de génération d'itemsets fréquents et consiste à utiliser une méthode de discrétisation supervisée et multi-variée pour la partition des attributs numériques. Ayant un effet considérable sur le résultat des itemsets fréquents générés, la méthode de discrétisation permet de choisir les points de coupure appropriés aux données considérées. Dans le contexte de l'apprentissage supervisé et du domaine des Règles d'Association, la discrétisation supervisée multi-variée prend en compte d'une part la classe des données et d'autre part les inter-dépendances entre les attributs pour construire les intervalles des différents attributs. L'effet de cette méthode sur les résultats de classification et sur la capacité de prédiction du système seront évalués dans le chapitre suivant. Les deux autres extensions de SIFRA proposées dans ce chapitre concernent l'utilisation de l'information liée à la classe lors de la génération des groupes d'attributs liés. Deux approches de regroupement d'attributs supervisé ont été alors proposées : la première incorpore l'attribut *Classe* dans la phase de génération des itemsets fréquents ; la deuxième prend en compte l'homogénéité des régions denses dans le calcul des degrés d'association. Ainsi, des groupes d'attributs qui sont liés avec l'attribut *Classe* sont obtenus dans la première méthode, alors que des groupes d'attributs dont les régions denses possèdent une homogénéité importante sont formés dans la deuxième approche. Nous fournissons, dans le chapitre suivant, des tests expérimentaux réalisés en utilisant les différentes versions de SIFRA sur différentes bases de données réelles.

Une perspective intéressante de ce travail consiste en la combinaison des deux approches de regroupement d'attributs supervisé. En d'autres termes, les deux méthodes "SAR avec Attribut Classe" et "SAR avec facteur d'homogénéité" seront combinées pour former une troisième méthode de regroupement doublement supervisée DSAR (Double Supervised Attributes Regrouping). Cette méthode prendra en compte l'attribut *Classe* dans la phase de génération des itemsets fréquents d'une part, et considérera l'homogénéité des régions denses dans le calcul du degré d'association d'autre part. Une autre perspective possible consiste en la génération d'une méthode de regroupement d'attributs triplement supervisée, cette fois-ci. Cette méthode, qu'on peut nommer TSAR (Triple Supervised Attributes Regrouping) prendra en considération les deux méthodes SAR ("SAR avec Attribut Classe" et "SAR avec facteur d'homogénéité") et ajoutera de plus la discrétisation supervisée.

Introduction

Nous avons développé les différentes méthodes proposées dans le cadre de cette thèse dans un système informatique, nommé SIFRA. Ce système implémente une méthode d'apprentissage supervisée par génération de règles floues ; il englobe les trois phases de notre méthode d'apprentissage : la phase de regroupement d'attributs, la phase d'apprentissage et la phase de classification. Les différentes versions et extensions de SIFRA proposées dans les chapitres 3 et 4 sont aussi intégrées dans la plateforme développée.

Nous présentons, dans le présent chapitre, les différentes études expérimentales et comparatives valorisant les apports de nos contributions en les testant sur différentes bases de données de l'UCI Repository¹.

Étant donné que notre méthode SIFRA est une méthode ensembliste qui cherche à associer les attributs liés dans des sous-groupes d'attributs, nous commençons nos études expérimentales par l'analyse des associations qui peuvent être détectées par SIFRA en les comparant aux associations trouvées par d'autres méthodes de regroupement d'attributs telles que la méthode SIFCO.

Dans un deuxième temps, une analyse du coût temporel d'exécution de notre méthode SIFRA est effectuée dans ce chapitre. Ainsi, des améliorations du coût d'exécution de la méthode SIFRA sont proposées, à savoir l'utilisation des algorithmes de génération d'itemsets fermés ou maximaux.

Dans un troisième temps, nous étudions l'effet des différents paramètres d'entrée de la méthode SIFRA, à savoir le type de la partition floue utilisée et les valeurs des seuils $minsupp$ et β_{min} . Nous menons aussi des tests expérimentaux avec différentes méthodes de discrétisation utilisées dans la phase de regroupement d'attributs pour discrétiser les attributs numériques. L'effet de ces méthodes de discrétisation sur les résultats de classification et sur les groupes d'attributs obtenus est analysé dans ce chapitre.

Finalement, nous menons différentes expérimentations pour évaluer la performance de classification de notre méthode SIFRA et de ses extensions SIFRA-SAR. Des comparaisons avec d'autres méthodes d'ensemble et d'autres méthodes d'apprentissage à base de règles

1. <https://archive.ics.uci.edu/ml/datasets.html>

floues sont ainsi présentées. L'évaluation des différentes méthodes se base sur deux critères principaux : le taux de bonne classification et le nombre de règles générées.

5.1 Implémentation

La méthode d'apprentissage supervisée à base de règles floues que nous avons proposée dans cette thèse a été implémentée en utilisant le langage de programmation Java sous l'environnement Eclipse.

Pour la réalisation de notre système SIFRA, nous avons utilisé la bibliothèque de la plateforme WEKA². Les structures de données de WEKA sont aussi adoptées dans notre développement. D'autre part, nous nous sommes basés sur le système SIFCO [Soua, 2009] qui implémente aussi une méthode d'apprentissage supervisée par regroupement d'attributs liés. Nous avons utilisé une partie du code déjà prêt de ce système (la phase d'apprentissage et la phase de classification) et nous y avons intégré le code de nos méthodes proposées (dans la phase de regroupement d'attributs). Pour l'implémentation de cette phase, nous avons également eu recours à une autre bibliothèque de fouille de données gratuite, la bibliothèque SPMF³ qui fournit plusieurs algorithmes de génération d'itemsets fréquents, fermés et maximaux. Une brève présentation de ces différentes plateformes utilisées pour la réalisation de notre système d'apprentissage est fournie en Annexe B.

Pour évaluer la capacité de généralisation de notre système SIFRA, il est nécessaire d'avoir deux ensembles de données : un ensemble pour la génération du modèle d'apprentissage et un autre pour le test de ce modèle généré. L'utilisation du même ensemble de données pour l'apprentissage et le test risque de générer une erreur biaisée, généralement très optimiste, conduisant à sous-estimer le risque de prévision réel [Arlot, 2018]. Dans notre travail, nous avons adopté la technique de validation croisée (*Cross Validation*) pour évaluer la validité de notre modèle d'apprentissage [Kohavi et al., 1995]. Le principe de la validation croisée consiste à diviser l'échantillon de données en D sous-ensembles ayant des effectifs approximativement égaux. La procédure de génération du modèle d'apprentissage est répétée D fois en utilisant à chaque fois $(D - 1)$ sous-ensembles comme échantillon d'apprentissage et en laissant le sous-ensemble restant pour la phase de test. La performance de chacun des D modèles générés est alors estimée sur le sous-ensemble restant et le taux de bonne classification de ce modèle est ainsi calculé. La performance du système global est calculée en moyennant les D taux de bonne classification obtenus. Nous avons utilisé la validation croisée d'ordre 10 ($D = 10$) dans tous nos tests expérimentaux.

Lors de l'implémentation de notre système SIFRA, nous avons réalisé une interface graphique pour faciliter à l'utilisateur la manipulation de notre système. Nous avons aussi profité de l'interface graphique du système SIFCO [Soua, 2009] que nous avons amélioré, et nous y avons intégré toutes nos méthodes. La figure 42 présente l'interface graphique

2. <https://www.cs.waikato.ac.nz/ml/weka/>

3. <http://www.philippe-fourmier-viger.com/spmf/>

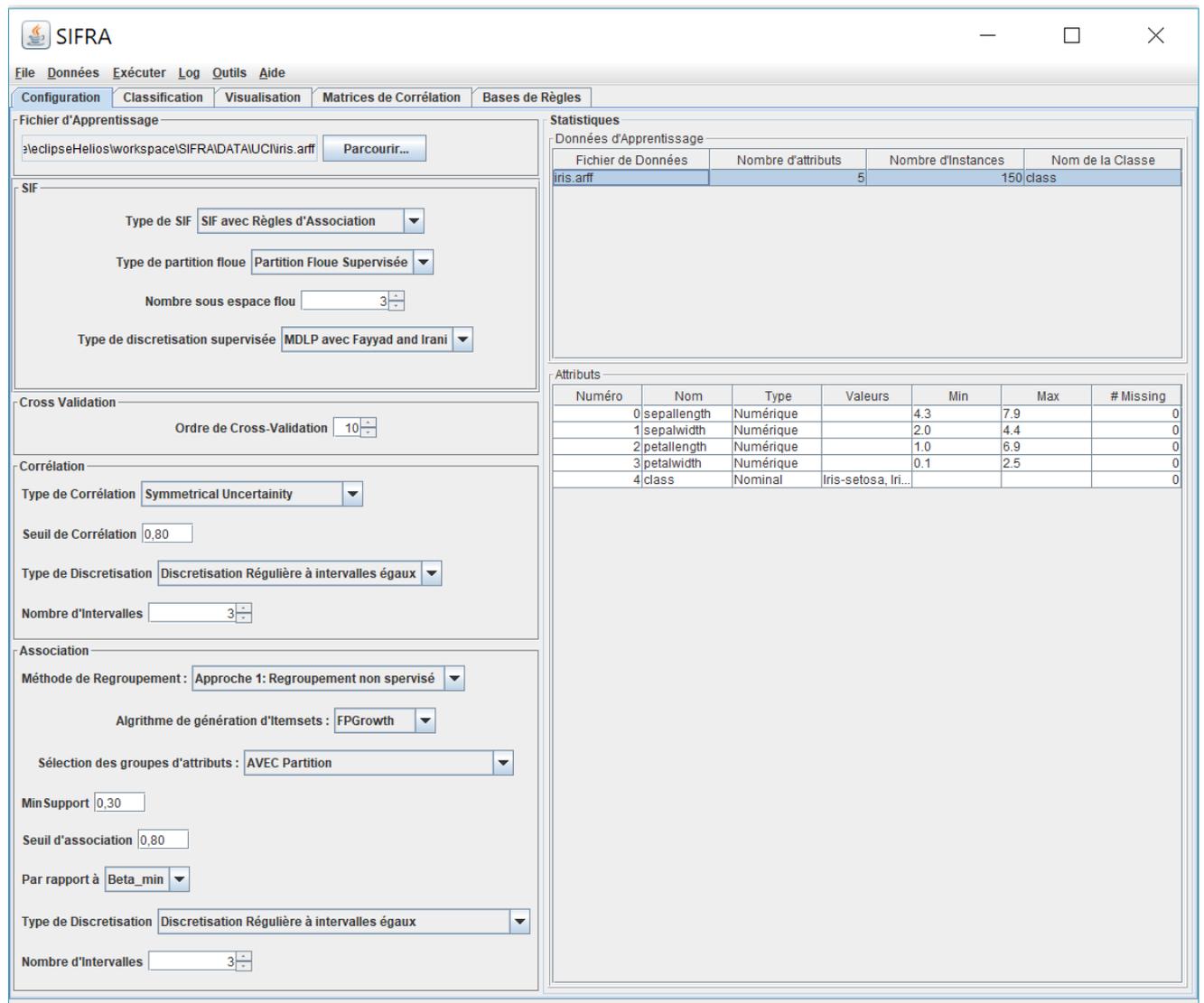


Figure 42 — Interface graphique de la plateforme SIFRA

de notre système SIFRA. Cette interface contient cinq modules de configuration (la partie gauche de la figure 42) :

- Module "Fichier d'apprentissage" : à travers cette partie graphique, l'utilisateur peut choisir le fichier de la base de données qu'il veut traiter. Les fichiers acceptés sont de type .arff.
- Module "SIF" : dans ce module, on choisit entre la méthode SIFCO (SIF avec recherche de Corrélation) ou SIFRA (SIF avec Règles d'Association). Puis, l'utilisateur peut choisir le type de la partition floue (simple ou supervisée). Si une partition floue simple est choisie, le nombre de sous-espaces flous peut être choisi à travers cette interface. Finalement, l'utilisateur peut sélectionner la méthode de discrétisation supervisée qu'il souhaite utiliser dans le cas d'une partition floue supervisée.
- Module "Cross Validation" : le nombre d'itérations de la méthode d'évaluation Cross-

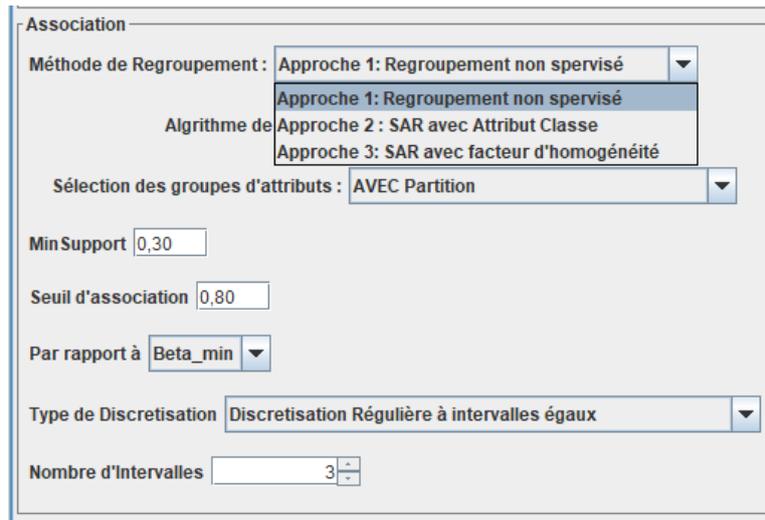


Figure 43 — Interface graphique : Choix de la méthode de regroupement d'attributs

validation peut être aussi défini à travers l'interface graphique.

- Module "Corrélation" : ce module concerne la méthode de regroupement d'attributs par recherche de corrélation linéaire proposée dans [Soua et al., 2013]. Ce module contient les différents paramètres à fixer par l'utilisateur pour cette méthode.
- Module "Association" : Ce module concerne notre méthode de regroupement d'attributs par RA. Tous nos approches et travaux figurent dans ce module.

Rappelons que la méthode SIFRA contient trois phases principales : la phase de regroupement d'attributs, la phase de génération de règles floues et la phase de classification (l'architecture générale de notre système SIFRA est schématisée en Annexe C). Les deux dernières phases sont prises directement du code de l'application SIFCO. Pour la première phase, phase de regroupement d'attributs, nous avons implémenté les différentes versions de cette phase, à savoir le regroupement d'attributs simple (chapitre 3) et le regroupement d'attributs supervisé (chapitre 4). La partie "Association" de l'interface graphique permet de choisir la méthode de regroupement d'attributs (voir figure 43).

L'extension de SIFRA à des algorithmes de génération d'itemsets fréquents plus performants que Apriori, et à des algorithmes de génération d'itemsets fermés et maximaux a été également considérée dans l'implémentation de SIFRA (Figure 44). Pour cela, nous avons utilisé le code de ces algorithmes fournis dans la bibliothèque SPMF. D'autre part, nous avons intégré plusieurs méthodes de discrétisation (régulières, supervisées et supervisées-multivariées) dans notre application SIFRA (voir Figure 45). Le code des méthodes régulières et supervisées a été pris de la plateforme WEKA. Cependant, nous avons développé nous-même le code de la méthode de discrétisation supervisée multi-variée.

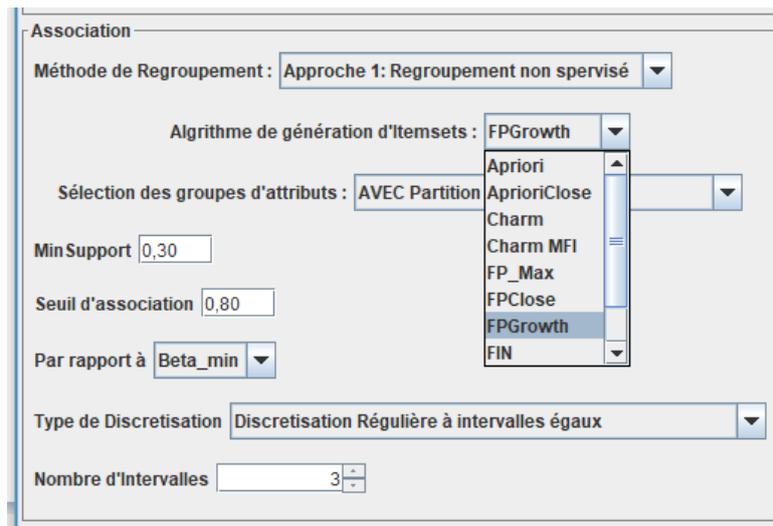


Figure 44 — Interface graphique : Choix de l'algorithme de génération des itemsets

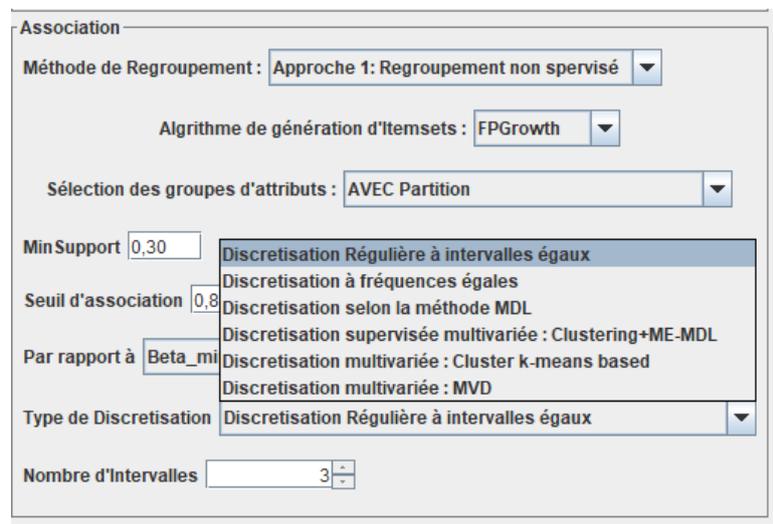


Figure 45 — Interface graphique : Choix de la méthode de discrétisation

5.2 Description des bases d'apprentissage

Nous menons cette étude expérimentale en utilisant différentes bases de données de l'UCI Repository⁴. Ces bases diffèrent par le nombre d'attributs, le nombre d'exemples et le nombre de classes. Nous avons choisi des bases de différentes grandeurs de dimension : variant de 4 jusqu'à 76 attributs (voir Tableau 7). Nous rappelons que dans le cadre de ce travail, celui des systèmes de classification à base de règles floues, toutes les variables sont quantitatives.

1. <https://archive.ics.uci.edu/ml/datasets.html>

Base de données	Nombre d'attributs	Nombre d'exemples	Nombre de classes
Iris	4	150	3
Balance-scale	4	625	3
Ecoli	7	336	8
Diabetes	8	768	2
Glass	9	214	7
Wine	13	178	3
Heart-Statlog	13	270	2
Vehicle	18	846	4
Unbalanced	32	856	2
Ionosphere	34	351	2
Sonar	60	208	2
mfeat-fourrier	76	2000	10

Tableau 7 — Description des bases de données

5.3 Analyse des associations

Nous nous focalisons dans cette section sur l'analyse des types d'associations qui peuvent être détectées par SIFRA ainsi qu'à l'étude des groupes d'attributs formés, et ce en comparaison à d'autres méthodes de regroupement d'attributs. Nous rappelons que notre méthode SIFRA cherche à construire les groupes d'attributs liés en utilisant le concept des règles d'association. Les associations entre les attributs numériques sont détectées en suivant trois étapes principales. D'abord, le principe de génération des itemsets fréquents est utilisé sur les attributs, discrétisés en des intervalles, pour analyser la co-occurrence des données dans des régions de l'espace et détecter par la suite les intervalles liés (ceux qui forment des régions denses). Ensuite, les liaisons trouvées entre les intervalles sont généralisées pour déduire les associations entre les attributs de départ. Pour cela, le facteur β est défini pour caractériser le degré d'association des attributs liés dans un même groupe. Finalement, les associations détectées ne sont pas toutes gardées (car elles présentent un nombre énorme de groupes associés), mais une sélection des groupes d'attributs les plus pertinents est élaborée en se basant sur deux critères principaux :

1. le groupe d'attributs qui possède le degré d'association β le plus grand est plus important que les autres
2. les groupes qui contiennent plus d'attributs sont préférés (les groupes les plus longs)

Nous attirons l'attention sur le fait que le choix de ces deux critères de sélection a un impact direct sur les groupes d'attributs générés. De ce fait, nous avons présenté dans la section 3.5 du chapitre 3 une analyse fine des critères de sélection adoptés. Dans cette analyse, nous avons discuté les raisons pour lesquelles nous avons choisi ces deux critères de sélection ainsi que notre choix de former une partition des attributs et d'éviter le

chevauchement entre les groupes. Afin de renforcer cette analyse, nous avons élaboré une étude expérimentale, présentée dans l'annexe [D](#), pour analyser l'effet de ces deux critères et du choix de la partition sur les résultats de classification et sur les groupes d'attributs obtenus. Dans cette étude, nous avons d'abord testé le fait d'enlever le deuxième critère qui correspond à la longueur des groupes d'attributs. La comparaison de ce test avec la version basique de SIFRA (en utilisant les deux critères dans l'ordre) a montré que la prise en compte des groupes d'attributs les plus longs peut significativement améliorer la performance de classification. D'autre part, nous avons testé le fait de commencer par le critère qui correspond à la longueur des groupes en premier lieu, puis d'analyser le degré d'association β en second lieu. L'analyse de cette méthode a montré qu'elle donne des résultats moins performants que la méthode SIFRA. Nous avons aussi analysé, dans l'annexe [D](#), la pertinence de notre choix qui consiste à former une partition des groupes d'attributs et d'éviter les intersections entre eux. Cette étude a été élaborée en utilisant tous les groupes d'attributs sans générer une phase de sélection. Les résultats expérimentaux de cette méthode ont montré qu'un nombre de règles très important peut être générée par cette méthode sans vraiment aboutir à de meilleurs taux de classification, et cela dans la majorité des bases que nous avons utilisées.

Pour toutes ces raisons, nous utilisons dans tous les tests expérimentaux élaborés de ce chapitre les deux critères de sélection proposés dans la section [3.3.3](#) du chapitre [3](#) pour former les groupes d'attributs finaux. Nous adoptons aussi le choix de former une partition des groupes d'attributs finaux.

Dans cette section, nous menons une comparaison de notre méthode SIFRA à d'autres méthodes d'ensemble qui se basent aussi sur le regroupement d'attributs, à savoir la méthode *Random Subspace* [\[Skurichina and Duin, 2002\]](#) et la méthode SIFCO [\[Soua et al., 2013\]](#). Nous rappelons que la méthode *Random Subspace* décompose l'ensemble des attributs descriptifs en des groupes d'attributs aléatoirement construits, tandis que la méthode SIFCO se base sur une recherche de corrélation linéaire entre les attributs pour former les groupes d'attributs liés. Nous visons à analyser les types d'associations et les groupes d'attributs détectés par notre méthode de regroupement d'attributs et à les comparer avec ceux détectés par la méthode SIFCO.

Dans le chapitre [3](#), nous avons émis l'hypothèse que la méthode de regroupement d'attributs proposée dans cette thèse pourrait être plus intéressante que SIFCO surtout dans le cas des bases de données qui ne contiennent pas de corrélations linéaires entre les attributs. Afin de confirmer cette hypothèse, nous proposons dans ce chapitre d'analyser le type de données utilisées dans l'expérimentation et nous distinguons deux types de données différents : les bases de données avec des attributs linéairement corrélés et les bases de données sans corrélation linéaire. Nous avons utilisé le logiciel RStudio⁴ pour identifier si une base de données contient des corrélations linéaires. RStudio est une application qui permet de calculer des statistiques et de générer des graphiques de qualité en se basant sur le langage et l'environnement de programmation R. À l'aide du logiciel RStudio, nous

4. <https://www.rstudio.com/products/RStudio/>

pouvons générer la matrice de corrélation d'une base de données et la représenter sous forme graphique (voir Figure 46). Les degrés de corrélation entre chaque paire d'attributs sont représentés par des cercles bleus : plus le cercle est sombre et grand, plus la corrélation est importante, par contre, un cercle fin et clair correspond à une faible corrélation. La figure 46 montre des matrices de corrélation graphiques pour les différentes bases de données utilisées dans cette partie ; nous considérons dans cette étude huit bases de données : Iris, Ecoli, Diabetes, Glass, Wine, Heart-statelog, Vehicle et Sonar.

D'après la figure 46.a, nous notons qu'il existe 3 corrélations linéaires importantes entre certains attributs de la base de données Iris. Les figures 46.b et 46.c montrent des corrélations linéaires importantes dans les bases de données Wine et Vehicle. La matrice de corrélation de Sonar (Fig. 46.d) contient également des cercles foncés qui correspondent à des corrélations linéaires importantes. Pour les bases Glass et Ecoli (Fig. 46.e et 46.f), nous remarquons qu'il n'y a qu'une seule corrélation linéaire importante pour chaque base. Ce nombre de corrélations est faible par rapport au nombre de liens possibles et peut être négligé. Pour la base de données Diabetes, la figure 46.g montre une seule corrélation peu importante entre deux attributs. Enfin, nous ne remarquons aucune corrélation importante pour Heart-statlog (Fig. 46.h). En conclusion, cette étude distingue deux groupes de bases de données :

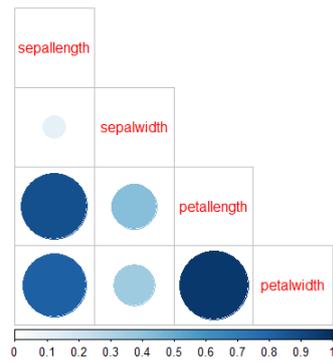
- Le premier groupe contient les bases de données Iris, Wine, Vehicle et Sonar qui ont des corrélations linéaires importantes.
- Le second groupe contient les bases de données Glass, Diabetes, Heart-statelog et Ecoli qui peuvent être considérées comme des bases sans corrélations linéaires.

5.3.1 Performance de SIFRA par rapport à d'autres méthodes de regroupement d'attributs

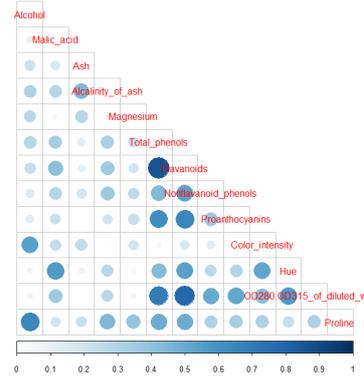
Nous fournissons une comparaison de notre méthode SIFRA à d'autres méthodes d'ensemble, à savoir la méthode *Random Subspace* [Skurichina and Duin, 2002] (version développée dans WEKA) et la méthode SIFCO [Soua et al., 2013]. Nous limitons dans ce paragraphe nos tests de comparaison aux méthodes d'ensemble qui divisent l'ensemble des attributs en sous-ensembles. Une comparaison avec d'autres méthodes d'ensemble comme le boosting et le bagging est fournie plus tard dans ce chapitre.

Nous utilisons la méthode *Random Subspace* pour construire les sous-ensembles de données tandis que la grille floue simple [Ishibuchi et al., 1992] est utilisée comme méthode d'apprentissage dans chaque sous-ensemble. Pour la recherche de corrélation dans la méthode SIFCO, nous avons considéré dans cette étude expérimentale trois mesures différentes : la mesure de corrélation linéaire de "Bravais Pearson", le coefficient de Cramer et le facteur SU.

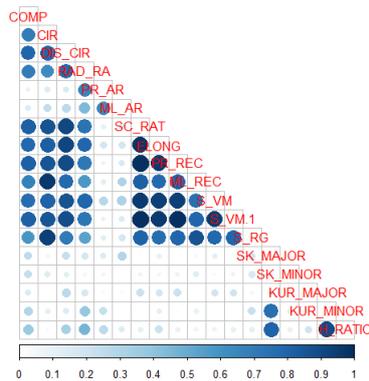
Dans la phase de regroupement d'attributs, certains paramètres doivent être spécifiés pour les deux méthodes SIFCO et SIFRA. Le seuil d'association β_{min} (dans SIFRA) et



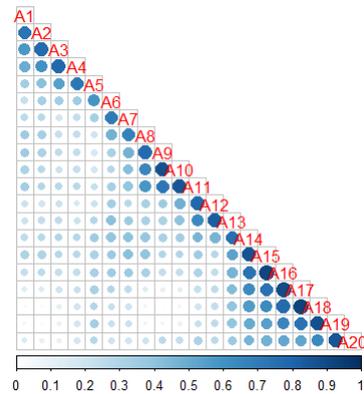
(a) Iris



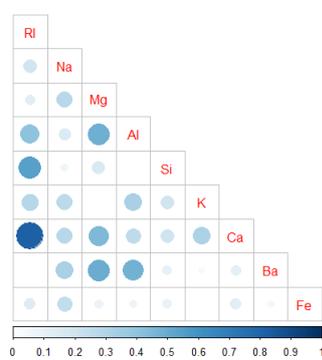
(b) Wine



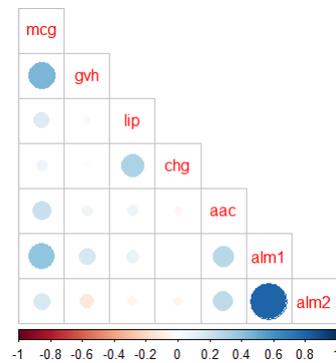
(c) Vehicle



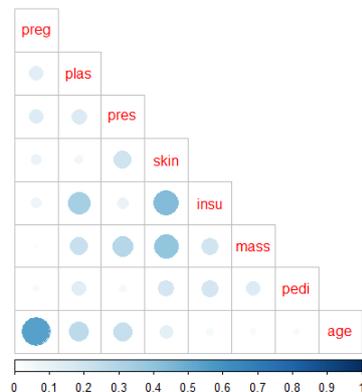
(d) Sonar



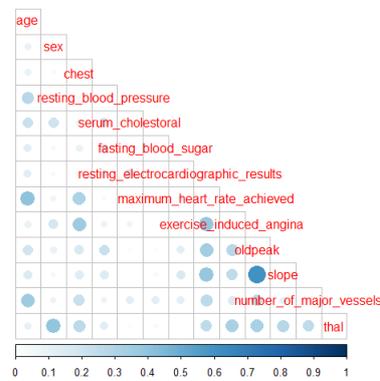
(e) Glass



(f) Ecoli



(g) Diabète



(h) Heart-Statelog

Figure 46 — Matrices de corrélation des bases de données

le seuil de corrélation (dans SIFCO) relèvent tous les deux le degré d'association des attributs qui sont réunis dans un même groupe. Étant donné que le but de la méthode de regroupement d'attributs est de détecter les associations les plus importantes, nous avons choisi de fixer de grandes valeurs pour ces deux seuils. Puisque les valeurs de ces seuils sont compris entre 0 et 1; nous avons testé les différentes valeurs 0.6, 0.7, 0.8 et 0.9. Le seuil *minsupp* est un autre paramètre à choisir dans notre méthode SIFRA. Il a été mentionné dans [Agrawal et al., 1993] qu'une très faible valeur de *minsupp* engendre la génération d'un nombre très important d'itemsets fréquents alors qu'une haute valeur de *minsupp* peut causer la perte de plusieurs associations. De plus, les valeurs de support dépendent fortement de la taille de l'échantillon (voir équation 1). Pour cela, nous avons testé plusieurs valeurs de *minsupp* allant de 0.1 jusqu'à 0.4. Nous avons toutefois utilisé des valeurs plus grandes de *minsupp* (0.5 et 0.6) pour les grandes bases de données (Sonar et Vehicle). Pour l'algorithme de génération d'itemsets fréquents, nous avons utilisé dans cette partie l'algorithme FP-Growth puisqu'il est moins coûteux que Apriori et qu'il génère tous les itemsets fréquents sans perdre aucune information. D'autres algorithmes de génération d'itemsets seront testés et étudiés dans la section 5.4. Finalement, l'application de l'algorithme FP-Growth (ou autre algorithme de génération d'itemsets) nécessite une phase de discrétisation des attributs numériques. Nous avons utilisé pour cette section la discrétisation régulière à 3 intervalles comme étant la méthode la plus simple. Les autres méthodes de discrétisation seront étudiées dans 5.5.

Dans la phase d'apprentissage, nous avons testés deux types de partitions floues : la partition homogène et la partition supervisée. Il a été démontré dans [Soua et al., 2013] que la partition floue supervisée améliore considérablement les taux de bonne classification pour la méthode SIFCO. Pour cette raison, nous adoptons dans ce paragraphe la partition floue supervisée et plus précisément la méthode MDLP [Fayyad and Irani, 1993]. Une analyse plus fine de l'effet de cette partition sur la méthode SIFRA est fournie dans le paragraphe 5.5.

Dans le tableau 8, nous présentons les taux de bonne classification suivis entre parenthèses du nombre de règles générées. Pour la méthode *Random Subspace*, WEKA ne donne pas le nombre de règles générées. Ainsi, seuls les taux de classification sont présentés dans le tableau 8. Les meilleurs taux de classification sont indiqués en gras. Nous rappelons que les quatre premières bases de données appartiennent au groupe des données qui possèdent plusieurs corrélations linéaires et que les quatre dernières bases de données ne contiennent aucune corrélation linéaire.

D'après le tableau 8, SIFRA et SIFCO dépassent considérablement les résultats de la méthode *Random Subspace*. En effet, la méthode *Random Subspace* génère aléatoirement les groupes d'attributs tandis que SIFCO et SIFRA cherchent des associations intéressantes entre les attributs. En comparant les trois versions de SIFCO (les trois mesures d'association utilisées dans SIFCO), nous constatons que les mêmes résultats sont obtenus par les trois méthodes dans plusieurs bases de données. Une petite amélioration est constatée par la mesure SU dans le cas de la base Iris (98% contre 97.33% pour Cramer

Base de données	Random subspace	SIFCO (V de Cramer)	SIFCO (Mesure SU)	SIFCO (Pearson)	SIFRA
Iris	68.66	97.33 (14)	98.00 (14)	96.00 (18)	97.33 (14)
Wine	93.25	98.87 (38)	98.87 (38)	98.87 (38)	98.87 (38)
Vehicle	54.96	60.28 (102)	55.43 (62)	60.99 (126)	67.61 (273)
Sonar	–	66.34 (81)	66.34 (81)	70.19 (68)	66.34 (76)
Glass	52.80	60.75 (21)	60.75 (21)	60.75 (21)	62.14 (13)
Diabetes	63.93	71.48 (21)	71.48 (21)	71.48 (21)	75.26 (35)
Heart-Statlog	76.66	68.88 (18)	68.88 (18)	70.74 (18)	77.40 (14)
Ecoli	48.51	53.57 (16)	55.95 (18)	55.95 (18)	63.98 (20)

Tableau 8 — Comparaison de la performance de SIFRA avec d'autres méthodes de regroupement d'attributs

et 96% pour Pearson) et une amélioration importante est obtenue par la mesure Pearson pour la base Sonar (70.19% contre 66.34% pour les deux autres mesures). Une petite amélioration est également obtenue par la mesure de Pearson avec les bases Vehicle et Heart-statelog. En conclusion générale, avec SIFCO, Bravais-Pearson semble être meilleur que les mesures de Cramer et SU.

Comparée à SIFCO, SIFRA donne presque toujours de meilleurs résultats, sauf pour les bases de données Sonar et Iris : SIFCO conduit à un meilleur taux de classification que SIFRA dans Sonar (70.19% contre 66.34%) et Iris (98% contre 97.33%). Les mêmes résultats sont obtenus par SIFCO et SIFRA dans le cas de la base de données Wine. Pour la base de données Vehicle, nous observons une amélioration significative du taux de bonne classification avec SIFRA. Nous obtenons une amélioration de près de 11% : 67.61% obtenu par SIFRA contre 60.99% obtenu par SIFCO. Cependant, le nombre de règles obtenues par SIFRA est plus grand. Rappelons que le nombre de règles dépend du nombre d'intervalles issus de la partition floue et du nombre d'attributs associés dans chaque groupe. Le nombre important de règles obtenues par SIFRA peut s'expliquer par la détection d'un plus grand nombre de sous-groupes contenant plus d'attributs associés (voir tableau 9).

D'après le tableau 8, nous remarquons également que les meilleurs taux de bonne classification sont obtenus par SIFRA dans les quatre dernières bases de données (les bases qui ne contiennent pas ou peu de corrélations linéaires). SIFRA améliore les taux de classification de 2.3% dans le cas de la base Glass, près de 5.2% dans le cas de la base diabète et plus de 9% dans la base de données Heart-statlog. Une nette amélioration de 14.35% est également obtenue avec Ecoli. Nous trouvons également des résultats satisfaisants pour le nombre de règles : presque le même nombre de règles que SIFCO pour les bases de données Heart-statlog et Ecoli, et un nombre de règles légèrement plus élevé pour la base de données Diabetes (qui reste quand même acceptable). Enfin, un nombre de règles réduit a été obtenu par notre méthode SIFRA dans le cas de la base Glass.

Ainsi, avec le deuxième groupe de données (données sans corrélations linéaires), SIFRA

surpasse SIFCO dans tous les cas. Par contre, pour le premier groupe qui correspond aux données linéairement corrélées (les quatre premières bases de données du tableau 8), seul le cas de Vehicle montre de bons résultats avec la méthode SIFRA. SIFCO surpasse SIFRA dans le cas d'Iris et Sonar et donne le même résultat avec Wine. Cela confirme notre hypothèse selon laquelle SIFRA est plus intéressante que SIFCO lorsqu'elle est appliquée à des bases de données sans corrélation linéaire. En revanche, SIFRA montre également un taux de classification très satisfaisant avec Vehicle, bien que cette base de données contienne un nombre élevé de corrélations. Ceci peut s'expliquer par le fait que Vehicle pourrait aussi contenir d'autres associations intéressantes (non linéaires) qui sont plus importantes que les corrélations linéaires et qui peuvent contribuer efficacement à la phase de classification. Afin de confirmer ce résultat et de mieux comprendre la contribution de SIFRA, nous proposons d'analyser les groupes d'attributs obtenus par les deux méthodes SIFCO et SIFRA et les différents types d'associations trouvés. C'est l'objectif de la section suivante.

Comme la mesure "Bravais-Pearson" donne des résultats meilleurs ou similaires que les deux autres mesures d'associations (le coefficient de Cramer et le facteur SU), nous ne considérons que le coefficient linéaire de Pearson dans SIFCO pour les comparaisons ultérieures.

5.3.2 Étude des groupes d'attributs obtenus

Dans le tableau 9, nous présentons les groupes d'attributs détectés par SIFCO (en utilisant le coefficient de Pearson) et SIFRA. Les attributs sont présentés avec des symboles numériques afin de simplifier la notation. Le symbole 1 fait référence à l'attribut X_1 dans la base de données correspondante. Les groupes résultants sont obtenus avec les mêmes paramètres que ceux utilisés dans le tableau 8.

Le tableau 9 montre que SIFCO et SIFRA donnent les mêmes groupes d'attributs pour Wine, c'est pour cela qu'on trouve les mêmes taux de classification dans le tableau 8. Pour la base de données Iris, le groupe {3, 4} a été détecté par les deux méthodes, mais SIFCO a associé l'attribut 1 à ce groupe. Dans le cas de la base Vehicle, les deux méthodes ont construit des groupes d'attributs différents, mais certaines associations ont été trouvées par les deux méthodes (comme l'association entre les attributs 9, 11 et 12). En ce qui concerne la base de données Sonar, nous remarquons que SIFCO et SIFRA ont donné des groupes d'attributs totalement différents. Ainsi, nous remarquons que, pour les quatre premières bases de données (contenant des corrélations linéaires élevées), SIFRA et SIFCO ont découvert des groupes d'attributs différents. Néanmoins, SIFRA détecte également certaines des corrélations linéaires découvertes par SIFCO. Dans les quatre dernières bases de données (sans corrélations linéaires), SIFCO ne trouve aucune association entre les attributs : chaque attribut est considéré comme indépendant. Ceci est prévisible puisque SIFCO ne recherche que des corrélations linéaires. En revanche, SIFRA découvre de nombreuses associations entre les attributs, ce qui conduit à de meilleurs taux

Base de données	Groupes d'attributs détectés par SIFCO	Groupes d'attributs détectés par SIFRA
Iris	[2] [1, 3, 4]	[1] [2] [3, 4]
Wine	[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]	[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]
Vehicle	[1] [2, 10, 13] [3, 7, 8, 9, 11, 12] [4] [5] [6] [14] [15] [16] [17] [18]	[1, 3, 5, 6, 8] [4, 18] [7, 10] [13] [2, 9, 11, 12, 14] [15, 16] [17]
Sonar	[1] [2] [3]... [14] [15, 16] [17, 18] [19] [20, 21] [22] ... [60]	[1, 57] [2, 3, 4, 5, 51, 52, 60] [6] [7] [8] [9] ... [46, 55] [47, 50] [56, 58, 59]
Glass	[1] [2] [3] [4] [5] [6] [7] [8] [9]	[1, 2, 5, 6, 8] [7, 9] [3] [4]
Diabètes	[1] [2] [3] [4] [5] [6] [7] [8]	[1, 2, 5, 7] [3, 6] [4, 8]
Heart-statlog	[1] [2] [3]... [12] [13]	[1] [2, 7] [6, 13] [3] [5] [4, 9] [10, 11] [8, 12]
Ecoli	[1] [2] [3] [4] [5] [6] [7]	[1, 2, 3, 4, 6] [5, 7]

Tableau 9 — Comparaison des groupes d'attributs, obtenus par SIFRA et SIFCO

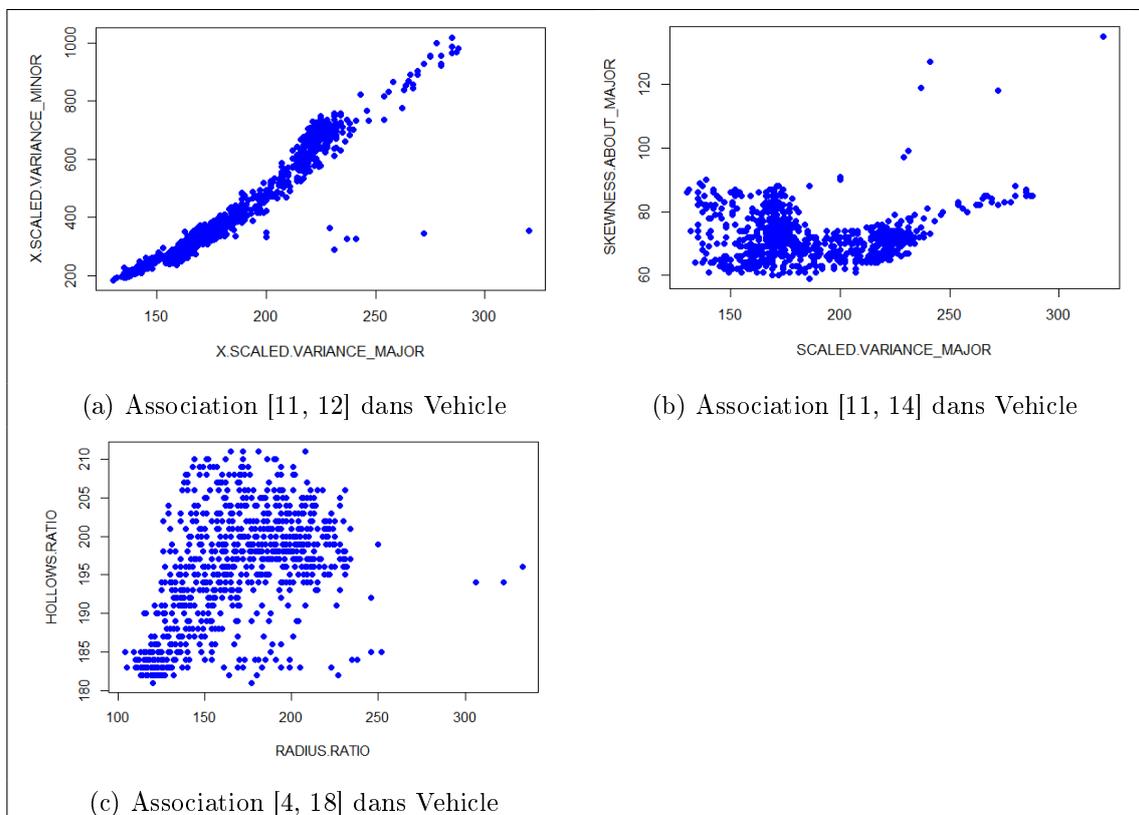


Figure 47 — Quelques types d'associations détectées par SIFRA

de classification. En conclusion, les résultats expérimentaux présentés dans le tableau 9 confirment l'hypothèse selon laquelle SIFRA est plus intéressante que SIFCO lorsqu'elle est appliquée à des bases de données sans corrélation linéaire. Le tableau 9 confirme que SIFCO n'est pas en mesure de construire des groupes d'attributs pour ce type de données. De ce fait, la phase de regroupement des attributs n'est pas utile dans SIFCO pour ce type de données. Néanmoins, SIFRA est capable de détecter différents types d'associations (linéaires ou pas). Cela est confirmé par une étude empirique fine des formes d'association découvertes par SIFRA et SIFCO. Cette étude est détaillée dans l'annexe E. Elle consiste à analyser les distributions des données entre chaque paire d'attributs, détectés comme liés, à travers des représentations graphiques fournies par RStudio (exemples de représentations graphiques dans la figure 47).

Cette étude a par exemple permis de montrer que SIFRA est capable de détecter des associations linéaires (comme le graphique 47.a) ou des associations non linéaires et non monotones (exemple dans 47.b). Elle peut également détecter de fortes concentrations de données dans une région de l'espace (47.c).

Ainsi, les résultats expérimentaux confirment que SIFRA peut détecter différents types de dépendances entre les attributs, y compris les corrélations linéaires et non linéaires. En fait, SIFRA ne cherche pas une forme spécifique de dépendance, mais elle se focalise sur la concentration des données dans certaines régions de l'espace.

5.4 Analyse du coût de la méthode SIFRA

Le coût de la méthode SIFRA peut être élevé dans certaines bases de données qui contiennent un nombre important d'attributs, et ceci est dû au coût important de l'algorithme Apriori dans ces bases. Dans ce contexte, nous avons proposé dans le chapitre 3 quelques solutions intéressantes pour réduire le coût d'exécution de la méthode SIFRA. L'utilisation d'algorithmes de génération d'itemsets fréquents basés sur des structures de données non-coûteuses comme FP-Growth, FIN et PrePost+ présente l'une des solutions envisagées. D'autres solutions plus rapides ont également été proposées telles que l'utilisation des algorithmes de génération d'itemsets fermés et des algorithmes de génération d'itemsets maximaux. Nous exposons, dans cette section, les tests expérimentaux menés avec ces solutions sur différentes bases de données. Nous choisissons des bases de données avec différents nombres d'attributs et nombres d'observations. Les nombres d'attributs traités varient entre 4 et 76. Nous rappelons que le traitement d'une dizaine d'attributs ou plus par les SCBRF provoque une explosion du nombre de règles générées [Ishibuchi et al., 1992, Ishibuchi et al., 1993, Ishibuchi and Yamamoto, 2005]. Notre but consiste à pouvoir traiter un nombre d'attributs moyennement important (entre 10 et 100), en utilisant la méthode de regroupement d'attributs proposée dans le chapitre 3, pour générer un classifieur flou performant dans un temps réduit.

Comme mentionné dans les paragraphes 1.2.2.3 et 3.6, plusieurs algorithmes de géné-

ration d'itemsets fréquents moins coûteux et plus rapides que Apriori ont été proposés dans la littérature (FP-Growth, FIN et PrePost+, ...). Nous choisissons de tester l'algorithme FP-Growth car c'est l'un des algorithmes les plus connus dans la littérature et il a prouvé une meilleure efficacité par rapport à Apriori dans plusieurs travaux. Nous rappelons que le choix d'un algorithme de génération des itemsets fréquents plus efficace et moins coûteux que Apriori résout le problème de la complexité de la première phase de SIFRA (la phase de génération des itemsets fréquents), mais le nombre important d'itemsets fréquents générés rend la deuxième phase de SIFRA (la phase de détermination des attributs liés) plus coûteuse. Comme discuté dans le chapitre 3, nous proposons de remédier à cela en utilisant des algorithmes de génération d'itemsets fermés ou maximaux qui fournissent un ensemble réduit d'itemsets. Dans ce contexte, nous choisissons l'algorithme FPclose pour la génération des itemsets fermés d'une part et l'algorithme FPmax pour la génération des itemsets maximaux d'autre part.

Des tests expérimentaux avec les trois algorithmes choisis (FP-Growth, FPclose et FPmax) ont été réalisés en comparaison avec ceux obtenus par la version classique de SIFRA (où Apriori a été utilisé pour la génération des itemsets fréquents). Le temps de réponse et la précision de classification sont les deux critères utilisés dans cette étude. Nous présentons en détails les différents tests menés et les résultats expérimentaux obtenus dans l'annexe E et nous exposons dans cette partie une synthèse de ces résultats.

Afin de pouvoir comparer et analyser les temps d'exécution des différentes solutions, les tests expérimentaux doivent être élaborés sous les mêmes conditions. Pour cela, nous avons utilisé les mêmes paramètres d'entrée pour chaque base de données. Il est à noter que certains paramètres, comme les seuils $minsupp$ et β_{min} , ont un effet considérable sur les résultats expérimentaux. En fait, le choix de ces deux seuils affecte le résultat de l'ensemble des itemsets fréquents générés, et influence par la suite le résultat de classification ainsi que le temps de réponse de la méthode SIFRA. Une analyse fine de l'effet des seuils $minsupp$ et β_{min} sur le temps de réponse de la méthode SIFRA est fournie dans l'annexe G.

En résumé, dans cette partie, nous avons testé cinq méthodes différentes de génération d'itemsets dans le contexte de SIFRA :

1. La première utilise l'algorithme Apriori et génère donc l'ensemble de tous les itemsets fréquents
2. La deuxième utilise l'algorithme FP-Growth qui génère aussi tous les itemsets fréquents mais qui est censé être plus rapide et efficace que Apriori
3. La troisième méthode utilise l'algorithme FPclose afin de générer seulement l'ensemble des itemsets fermés
4. La quatrième utilise aussi l'algorithme FPclose mais l'adapte de manière à ce qu'il puisse retrouver tous les itemsets fréquents et non-fermés. Cette adaptation permet

d'éviter la perte des régions liées qui peuvent être intéressantes pour la phase de classification (voir paragraphe "Génération d'itemsets fermés" dans [3.6](#))

5. La cinquième méthode utilise l'algorithme FPmax et génère alors l'ensemble des itemsets maximaux qui est plus réduit que tous les ensembles précédents.

L'étude expérimentale réalisée pour comparer les cinq versions de SIFRA et les résultats obtenus sont présentés dans l'annexe [F](#). Ils ont permis de déduire les résultats suivants :

- Concernant l'algorithme FP-Growth, son utilisation garantit l'obtention de tous les groupes d'attributs envisagés sans perdre aucune information intéressante, et donc il n'affecte pas la performance de classification. Par ailleurs, l'algorithme FP-Growth garantit la réduction du temps de réponse par rapport à Apriori et a aussi montré une réduction considérable du temps de réponse par rapport à FPclose dans certaines bases de données.
- Lorsque nous utilisons l'algorithme FPclose, nous obtenons généralement une réduction considérable du temps de réponse par rapport aux différents algorithmes. Cependant, une baisse de la performance de classification pourrait être obtenue à cause de la perte de quelques régions liées. Cet inconvénient a été surtout marqué dans les petites bases de données. En revanche, il n'a pas trop influencé la performance de classification dans les grandes bases de données. Ceci peut être expliqué par le fait que plusieurs associations peuvent être détectées lorsque le nombre d'attributs est important et que l'absence de quelques associations n'affecte pas trop le résultat de classification, par contre, dans les bases ayant un petit nombre d'attributs, un petit nombre d'associations est découvert par la méthode de regroupement d'attributs et l'élimination de quelques régions liées peut supprimer toutes les associations trouvées dans ces bases.
- L'utilisation de FPclose avec l'adaptation de son code pour qu'il retrouve toutes les régions liées ratées ne semble pas intéressante surtout dans le cas des bases de données ayant un grand nombre d'attributs. En effet, le traitement supplémentaire qui cherche les régions ratées nécessite un temps de réponse parfois beaucoup plus important que le temps mis par Apriori.
- En utilisant l'algorithme FPmax, le temps de réponse est généralement réduit par rapport à celui obtenu par Apriori. Cependant, aucun groupe d'attributs n'a été détecté dans la plupart des bases de données : chaque attribut a été considéré comme indépendant et traité seul par un classifieur. En effet, les itemsets de longueur importante possèdent généralement des supports faibles ; de même, les groupes d'attributs les plus longs possèdent généralement les degrés d'association les plus faibles qui ne peuvent généralement pas excéder le seuil d'association β_{min} . L'utilisation de l'algorithme FPmax ne semble donc pas être adaptée à notre méthode SIFRA puisqu'il ne

satisfait pas l'objectif principal de la phase de regroupement d'attributs qui consiste à détecter des associations intéressantes entre les attributs.

Pour conclure, nous notons que les deux choix les plus intéressants et adaptés à notre méthode SIFRA sont les algorithmes FP-Growth ou FPclose (sans adaptation). FPclose possède l'avantage de réduire considérablement le temps de réponse dans la plupart des bases de données. Cependant, il peut provoquer la perte de quelques informations utiles et diminuer par la suite la performance de classification de notre méthode SIFRA dans certaines bases de données. D'un autre côté, l'algorithme FP-Growth permet de détecter toutes les associations intéressantes sans perdre aucune information et garantit de plus la réduction du temps de réponse par rapport à Apriori et par rapport à FPclose dans certains cas. Pour cela, nous choisissons d'utiliser l'algorithme FP-Growth dans le reste de ce chapitre afin de mener des tests expérimentaux rapides tout en gardant une bonne performance et précision de notre système.

5.5 Analyse des effets des paramètres sur la méthode SIFRA

La méthode SIFRA requiert le choix de certains paramètres d'entrée. Le choix de ces paramètres peut influencer les résultats des classifieurs flous ; il peut notamment influencer le résultat des groupes d'attributs générés, les taux de bonne classification, le nombre de règles générées ou encore le temps de réponse. Les valeurs de ces paramètres sont choisies et fixées par l'utilisateur. Certains de ces paramètres sont requis pour la phase d'apprentissage (génération des règles floues) ; ces paramètres sont :

- la méthode de la partition floue : supervisée ou simple
- la taille de la partition floue k (si une partition simple est choisie)

D'autres paramètres sont requis pour la phase de regroupement d'attributs ; ces paramètres sont :

- le seuil $minsupp$
- le seuil β_{min}
- l'algorithme de génération des itemsets fréquents
- la méthode de discrétisation utilisée pour l'application de l'algorithme de génération des itemsets fréquents

Cette partie est consacrée à l'étude de tous ces paramètres à l'exception de l'algorithme de génération des itemsets fréquents. Nous avons en effet analysé, dans le paragraphe

précédent, l'effet du choix de ces algorithmes sur le temps de réponse et sur la performance de la méthode SIFRA. Suite à cette étude, nous avons choisi d'utiliser l'algorithme FP-Growth pour la génération des itemsets fréquents dans cette section.

5.5.1 Effet de la méthode de partition floue

Nous commençons cette étude par l'analyse de l'effet de la méthode de partition floue sur les résultats de SIFRA ; les résultats sont traités en terme de taux de bonne classification et du nombre de règles générées. Réduire le nombre de règles dans les SCBRFs est l'objectif principal de notre travail. Cependant, le souci majeur des méthodes de classification consiste à obtenir la meilleure précision (le meilleur taux de bonne classification). Nous cherchons alors dans ce travail à trouver un compromis entre ces deux paramètres : c'est à dire pouvoir réduire le nombre de règles générées par rapport aux SCBRFs classiques sans dégrader la performance de classification.

Deux méthodes de partition floue ont été testées dans ce travail : la partition floue simple qui utilise la discrétisation régulière *EWI* et la partition floue supervisée qui utilise la discrétisation supervisée *MDLP* [Sousa et al., 2013]. La méthode proposée dans [Sousa et al., 2013] a montré une amélioration considérable des taux de classification dans les SCBRF.

Les résultats expérimentaux de cette étude sont représentés en détail dans l'annexe H. Le tableau 19 de l'annexe H contient toutes les valeurs des taux de bonne classification et des nombres de règles générées par SIFRA en utilisant les deux méthodes de partition différentes. Pour la partition simple, plusieurs valeurs de k (la taille de la partition) ont été testées. Pour résumer ces résultats, nous présentons dans la figure 48 un histogramme qui sert à comparer les deux méthodes de partition floue en termes du taux de bonne classification et du nombre de règles générées. Dans le cas de la partition floue simple, une seule valeur de k , celle qui fournit le meilleur taux de bonne classification, est considérée.

La figure 48 montre que la partition floue supervisée fournit des résultats meilleurs que ceux obtenus par la partition simple dans la plupart des bases de données. Elle améliore considérablement le taux de bonne classification dans la majorité des bases (Wine, Vehicle, Glass, Diabetes, Heart-statlog, Balance-scale et Ionosphere) et diminue aussi significativement le nombre de règles générées dans toutes les bases (sauf Vehicle et mfeat-fourrier). Cette méthode de partition floue permet alors d'avoir un compromis entre ces deux facteurs. Pour cette raison, nous utilisons la partition floue supervisée dans le reste des expérimentations présentées dans ce rapport.

5.5.2 Effet des seuils $minsupp$ et β_{min}

Nous avons analysé l'effet des seuils $minsupp$ et β_{min} sur le temps de réponse de la méthode SIFRA en annexe G. Nous complétons, dans cette partie, l'étude des effets de ces deux paramètres sur la performance de SIFRA en termes de taux de bonne classification

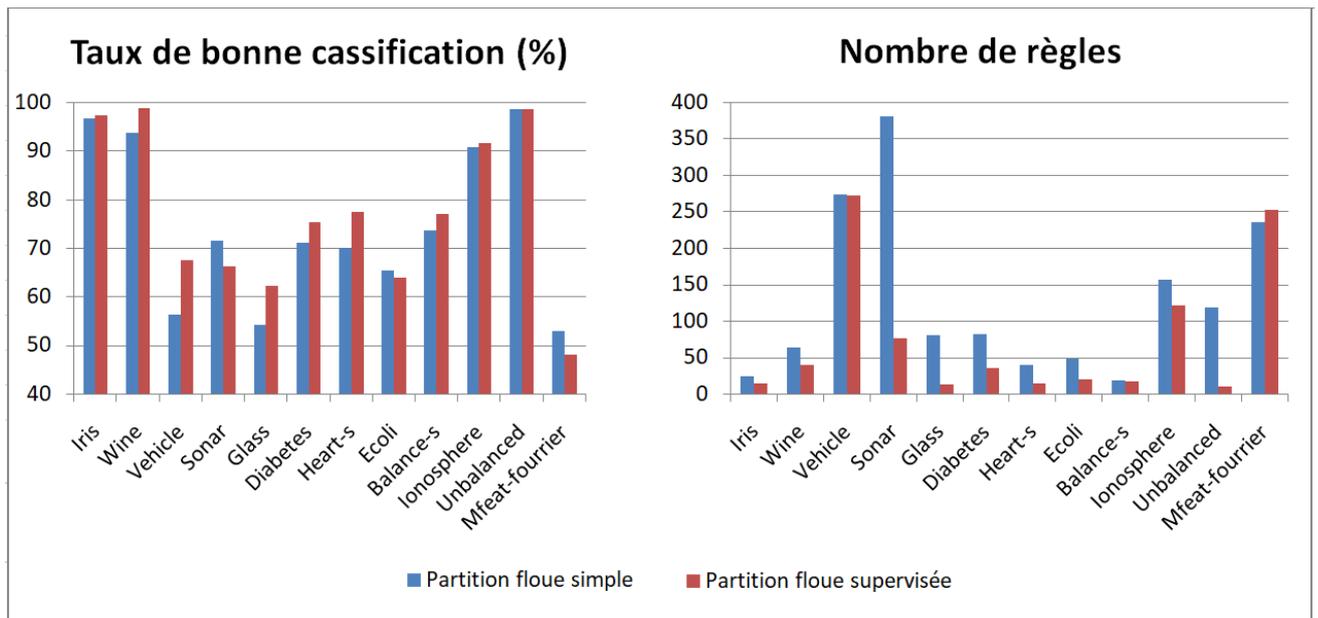


Figure 48 — Effet de la partition floue sur la méthode SIFRA

et du nombre de règles générées.

Nous avons noté précédemment que la valeur du seuil $minsupp$ affecte l'ensemble des itemsets générés. La diminution de ce seuil engendre un ensemble plus grand d'itemsets fréquents (la même chose pour les itemsets fermés). De ce fait, l'ensemble des groupes d'attributs sera ainsi modifié par la modification du seuil $minsupp$. L'ensemble des groupes d'attributs liés est aussi influencé par la valeur du seuil β_{min} . La diminution de la valeur de β_{min} engendre la génération d'un ensemble plus grand de groupes d'attributs liés. Il est à noter que le choix des groupes d'attributs liés a un impact important sur les résultats de classification. Pour cela, nous proposons d'analyser dans cette partie l'effet des seuils $minsupp$ et β_{min} sur les résultats de classification de notre méthode SIFRA, notamment sur le taux de bonne classification et le nombre de règles générées.

Nous avons mené des tests expérimentaux en utilisant différentes valeurs des seuils $minsupp$ et β_{min} . Dans cette étude, nous avons considéré six bases de données avec différents nombres d'attributs et d'observations. L'étude expérimentale et les résultats obtenus sont présentés dans l'annexe [H](#). Suite à cette étude, nous avons constaté que le choix des valeurs appropriées des seuils $minsupp$ et β_{min} dépend des données. Le comportement de ces seuils est illustré par la figure [49](#) qui représente les valeurs de $minsupp$ et β_{min} donnant les meilleurs taux de bonne classification pour les différentes bases testées. Cette figure montre que la valeur appropriée de $minsupp$ diffère d'une base à une autre. Quant au seuil β_{min} , la figure [49](#) montre que la valeur 0.7 donne presque toujours le meilleur taux de classification (à l'exception de la base Ionosphere).

Il serait donc intéressant de proposer une méthode de paramétrage automatique qui détermine, pour chaque base, les meilleurs seuils d'une manière automatique. Une solution possible consisterait à utiliser un Algorithme Génétique (AG) et à considérer les deux

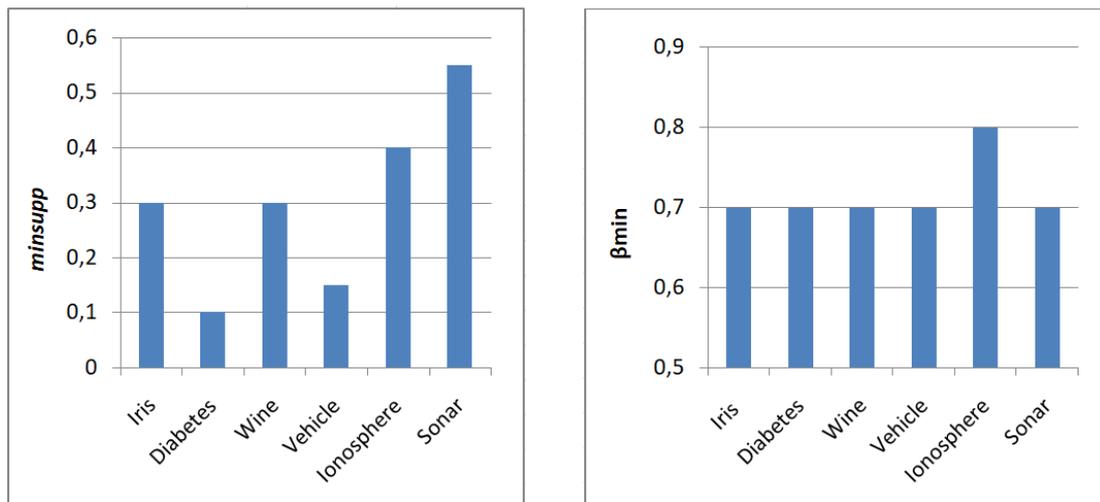


Figure 49 — Valeurs de $minsupp$ et β_{min} donnant le meilleur taux de classification

facteurs (nombre de règles et taux de bonne classification) dans la fonction *fitness* de l'AG. L'étude de cette solution est envisagée comme perspective de ce travail. Pour le reste de ce rapport, nous choisissons les valeurs de paramètres ayant conduit aux meilleurs résultats obtenus pour chaque base de données.

5.5.3 Effet de la méthode de discrétisation pour le regroupement d'attributs

Dans la phase de regroupement d'attributs de SIFRA, une première étape de discrétisation est requise pour partitionner les attributs numériques en des intervalles et pouvoir appliquer par la suite un algorithme de génération des itemsets fréquents. Dans le chapitre 4, nous avons noté que la méthode de discrétisation choisie peut avoir un effet majeur sur les résultats des groupes d'attributs générés et nous avons étudié quelques méthodes de discrétisation qui peuvent être adaptées à notre contexte. Parmi les méthodes étudiées, la discrétisation supervisée-multivariée se distingue par le fait qu'elle prend en compte l'interdépendance entre les attributs d'une part et l'information liée à la classe d'autre part. Nous rappelons que notre méthode de regroupement d'attributs vise à dégager les liaisons entre les attributs, donc prendre en compte les inter-dépendances entre ces attributs lors de leur discrétisation pourrait être très intéressant. De plus, puisque les groupes d'attributs générés seront utilisés pour construire des règles de classification floues, il serait donc intéressant de considérer l'information liée à la classe lors de la phase de discrétisation. Pour cela, nous avons choisi d'utiliser une méthode de discrétisation à la fois supervisée et multivariée dans la phase de regroupement d'attributs de SIFRA.

D'un autre côté, nous rappelons que la deuxième phase de SIFRA (la phase d'apprentissage ou de génération des règles floues) requiert aussi une étape de discrétisation pour le partitionnement flou des variables d'entrée en des sous-espaces flous (voir section 1.3.1.3 du chapitre 1). Nous avons adopté la méthode de discrétisation supervisée MDLP

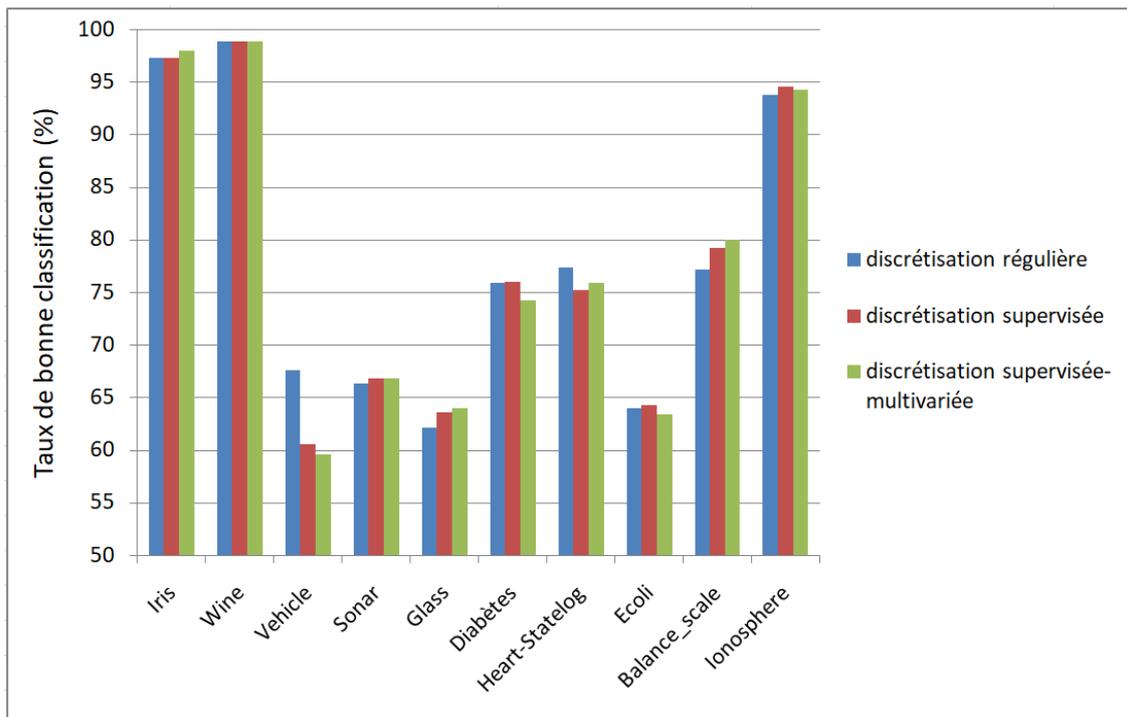


Figure 50 — Effet des méthodes de discrétisation sur la performance de la méthode SIFRA

[Fayyad and Irani, 1993] dans cette phase puisqu'elle a montré des résultats meilleurs que ceux obtenus par la discrétisation régulière (voir paragraphe "Effet de la méthode de partition floue" et annexe H). Il serait intéressant de tester l'utilisation de la même méthode de discrétisation dans les deux phases : la phase de regroupement d'attributs et la phase d'apprentissage. La discrétisation supervisée de MDLP [Fayyad and Irani, 1993] est adoptée dans ce cas dans les deux phases.

Nous menons alors des tests expérimentaux de SIFRA avec trois méthodes de discrétisation différentes :

- la discrétisation régulière : la méthode EWI avec 3 intervalles
- la discrétisation supervisée : la méthode MDLP de [Fayyad and Irani, 1993]
- la discrétisation supervisée-multivariée : la méthode nommée "Clustering+MDLP" de [Gupta et al., 2010]

Les valeurs précises des taux de bonne classification et les groupes d'attributs liés obtenus dans différentes bases de données sont présentés en Annexe I. Afin de bien illustrer l'effet de ces méthodes et de pouvoir les comparer, nous présentons sous forme graphique (Figure 50) les taux de bonne classification obtenus par SIFRA en utilisant les trois méthodes de discrétisation.

D'après cette figure, nous remarquons que la discrétisation supervisée-multivariée réalise de meilleurs résultats par rapport aux deux autres méthodes de discrétisation dans

le cas des bases de données Iris, Glass et Balance-scale. Cependant, comparée à la discrétisation régulière, nous notons que la discrétisation supervisée-multivariée a dégradé la performance de classification dans 4 bases de données.

Avec les bases Diabetes, Ecoli et Ionosphere, nous trouvons que les meilleurs taux de bonne classification sont obtenus avec la discrétisation supervisée. Il est aussi à noter que cette méthode de discrétisation réalise de meilleures performances que la discrétisation régulière dans presque tous les cas (sauf avec les bases Vehicle et Heart-statlog). Cela confirme le fait qu'utiliser la même méthode de discrétisation dans la phase de regroupement d'attributs et la phase d'apprentissage de SIFRA pourrait améliorer les résultats de classification.

Finalement, nous remarquons que la discrétisation régulière donne parfois des résultats de classification plus intéressants que ceux obtenus par les deux autres méthodes de discrétisation (comme le cas des bases Vehicle et Heart-statlog).

Pour conclure, la discrétisation supervisée-multivariée pourrait être intéressante pour certaines bases de données (comme Iris, Glass et Balance-scale). Cependant, il s'avère que la discrétisation supervisée présente aussi un choix intéressant dans plusieurs autres bases de données. Ceci peut être expliqué par le fait que la même méthode de discrétisation est utilisée dans la phase d'apprentissage de SIFRA. Néanmoins, dans les bases Vehicle et Heart-statlog, la discrétisation régulière présente des résultats très satisfaisants par rapport aux deux autres méthodes. À travers ces résultats, nous déduisons que le choix de la méthode de discrétisation appropriée s'avère être dépendant des données. Élargir les bases de tests et analyser la nature des données testées peut être envisageable comme perspective à cette étude.

L'utilisation de la méthode de discrétisation supervisée dans SIFRA montre des résultats satisfaisants dans certaines bases de données. Il pourrait aussi être intéressant d'utiliser cette méthode de discrétisation dans le cas de l'extension supervisée de SIFRA, la méthode SIFRA-SAR. Comme la méthode SIFRA-SAR utilise une approche de regroupement d'attributs supervisée, la prise en compte de l'attribut *Classe* dans la phase de discrétisation pourrait aussi être intéressant. Dans ce contexte, nous proposons d'utiliser la méthode de discrétisation supervisée MDLP dans les deux versions de l'approche SIFRA-SAR : "SAR avec Attribut Classe" et "SAR avec facteur d'homogénéité". Nous proposons aussi de tester la méthode de discrétisation supervisée-multivariée dans ces deux méthodes SAR. Les résultats de tous ces tests sont fournis en Annexe [I](#). À travers cette étude, nous notons que les résultats obtenus sur les méthodes SIFRA-SAR ont les mêmes tendances que ceux obtenus avec SIFRA : l'utilisation des méthodes de discrétisation supervisée ou supervisée-multivariée dans le contexte des approches SIFRA-SAR semble intéressante dans certaines bases de données. Cependant, la discrétisation régulière paraît plus adaptée à nos méthodes SIFRA-SAR dans plusieurs cas.

5.6 Analyse de la performance des méthodes proposées

Cette section est dédiée à l'analyse de la performance de nos méthodes proposées en les comparant à d'autres méthodes d'ensemble et d'autres méthodes de classification floue. D'abord, nous commençons notre étude par l'analyse et la comparaison des différentes méthodes proposées dans cette thèse : SIFRA, "SAR avec l'attribut Classe" et "SAR avec facteur d'homogénéité". Puis, nous fournissons une étude comparative de nos méthodes avec d'autres méthodes d'ensemble qui opèrent par répartition des données d'entrée entre plusieurs classifieurs et par combinaison des modèles générés par les différents classifieurs. Enfin, nous menons une dernière comparaison de nos méthodes avec d'autres méthodes de classification floues.

Pour valider la performance de nos méthodes, nous considérons dans cette section deux critères principaux : le taux de bonne classification et le nombre de règles générées. Le but est de trouver un compromis entre un taux de classification important et un nombre de règles réduit.

Nous rappelons également que l'application des méthodes SIFRA et SIFRA-SAR nécessite de fixer certains paramètres d'entrée. L'analyse de ces paramètres dans les paragraphes précédents nous a mené à fixer les choix suivants :

- La partition floue utilisée dans ces tests est la partition floue supervisée. L'étude expérimentale présentée dans la section [5.5](#) a prouvé que ce type de partition aide à améliorer la performance de SIFRA dans presque tous les tests réalisés.
- Nous choisissons d'utiliser l'algorithme FP-Growth pour la génération des itemsets fréquents. Ce choix permet de réduire la complexité et le coût de la méthode SIFRA sans réduire la performance de classification.
- Concernant la discrétisation des attributs dans la phase de regroupement, nous adoptons la discrétisation régulière à 3 intervalles vue qu'elle est la plus simple et vue qu'elle a conduit à des résultats satisfaisants dans plusieurs bases de données dans l'étude menée dans la section [5.5](#).
- Pour les seuils $minsupp$ et β_{min} , nous avons remarqué d'après l'analyse présentée dans [5.5](#) que le choix des valeurs appropriées dépend fortement des données. Pour cela, nous testons plusieurs valeurs de ces seuils et nous choisissons les meilleurs résultats obtenus pour chaque base de données.

5.6.1 Comparaison entre SIFRA et SIFRA-SAR

Dans cette section, nous présentons une comparaison de nos méthodes proposées : SIFRA, "SAR avec l'attribut Classe" et "SAR avec facteur d'homogénéité". Nous rappelons que la méthode SIFRA utilise une approche de regroupement d'attributs qui cherche les

associations intéressantes entre les attributs sans considérer leur liaisons avec la classe. Alors que les deux approches SAR cherche les associations intéressantes entre les attributs en analysant l'information liée à la classe de deux manières différentes : la première inclut l'attribut *Classe* dans la phase de génération des itemsets fréquents et la deuxième privilège les régions denses qui possèdent une homogénéité importante par rapport à l'attribut *Classe*.

Nous considérons également dans la méthode "SAR avec facteur d'homogénéité" les deux méthodes d'agrégation proposées : *Max* (pour dénoter l'homogénéité maximale parmi les homogénéités des différentes régions liées) et *Moy* (qui signifie la moyenne des homogénéités des différentes régions liées).

Le tableau [10](#) montre les taux de bonne classification obtenus par les différentes méthodes, suivies entre parenthèses du nombre de règles générées.

D'après ce tableau, nous remarquons que des résultats similaires sont obtenus par toutes les approches avec la base Iris. Ceci est dû au fait que les mêmes groupes d'attributs sont trouvés par les différentes méthodes. Cependant, les méthodes de regroupement d'attributs supervisées donnent de meilleurs résultats que SIFRA dans presque toutes les autres bases de données, à l'exception de Vehicle. Elles améliorent de façon significative les taux de bonne classification pour les cinq dernières bases de données.

Ainsi, "SAR avec l'attribut Classe" améliore les taux de classification de 3% par rapport à SIFRA dans le cas de la base de données Glass et de 0.85% dans la base Diabetes. La méthode "SAR avec facteur d'homogénéité" donne de meilleurs résultats que SIFRA et que "SAR avec l'attribut Classe" pour les bases de données Heart-statlog, Ecoli et Balance-scale. Une amélioration considérable de 27% est obtenue dans le cas de la base Ecoli. Avec les bases de données Sonar et Wine, la méthode "SAR avec facteur d'homogénéité" donne des résultats légèrement meilleurs que SIFRA et que "SAR avec l'attribut Classe". Pour la base de données Vehicle, la version "Moy" de la méthode "SAR avec facteur d'homogénéité" donnent une précision légèrement inférieure à SIFRA (66.19% contre 67.61%). Le nombre de règles dans ce cas est inférieur à celui obtenu par SIFRA.

En résumé, les méthodes SAR semblent être meilleures que SIFRA dans presque tous les cas, à l'exception de la base de données Vehicle. Cela confirme que le fait de privilégier les groupes d'attributs qui sont associés à la classe peut améliorer les taux de bonne classification. Enfin, pour comprendre pourquoi SIFRA surpassent les méthodes SAR pour la base Vehicle, une analyse détaillée de la nature des données et de leurs liaisons avec l'attribut *Classe* devrait être effectuée dans un travail futur.

Concernant le nombre de règles générées, il varie légèrement entre les différentes approches pour toutes les bases de données, à l'exception de la base de données Diabetes. En fait, le nombre de règles générées dépend du nombre d'attributs dans chaque groupe. Ainsi, le grand nombre de règles obtenues par la méthode "SAR avec facteur d'homogénéité" dans le cas de la base Diabetes peut s'expliquer par le fait que des groupes contenant un grand nombre d'attributs liés ont été détectés.

	SIFRA	SAR avec l'attribut Classe	SAR avec facteur d'homogénéité	
			Moy	Max
Iris	97.33 (14)	97.33 (14)	97.33 (14)	97.33 (14)
Wine	98.87 (38)	98.87 (38)	99.43 (39)	98.87 (38)
Vehicle	67.61 (273)	65.13 (131)	66.19 (148)	65.95 (156)
Sonar	66.34 (78)	66.34 (76)	66.82 (78)	66.82 (78)
Glass	62.14 (13)	64.48 (24)	63.08 (31)	63.55 (8)
Diabetes	75.26 (35)	76.95 (40)	76.43 (79)	76.56 (97)
Heart-statlog	77.40 (14)	79.25 (13)	80.00 (17)	80.00 (17)
Ecoli	63.98 (20)	66.07 (18)	81.25 (47)	81.25 (47)
Balance-scale	77.12 (17)	77.28 (17)	78.40 (30)	77.92 (30)

Tableau 10 — Comparaison entre les méthodes proposées : SIFRA et SAR

Nous rappelons que dans la méthode "SAR avec facteur d'homogénéité", le degré d'association d'un groupe d'attributs β prend en considération deux facteurs : β_1 qui représente la densité des données dans les régions liées et β_2 qui représente le degré d'homogénéité des différentes régions liées du groupe d'attributs. Ainsi, pour calculer le degré d'homogénéité β_2 , nous avons proposé dans le chapitre 4 d'utiliser deux méthodes d'agrégation différentes :

1. β_2 est calculé comme étant la moyenne des homogénéités des différentes régions liées (voir équation 58), désignée par Moy
2. β_2 est égal à la valeur maximale des homogénéités des différentes régions liées (voir équation 60), désignée par Max

D'après le tableau 10, nous remarquons que ces deux méthodes (la moyenne et le maximum) donnent les mêmes résultats dans 4 bases de données et une différence mineure dans les autres bases de données. La faible différence obtenue dans la plupart des cas s'explique par le fait que le facteur d'homogénéité β_2 a généralement une valeur faible par rapport au coefficient β_1 (quelque soit la fonction d'agrégation utilisée). Ainsi, modifier la méthode du calcul du facteur d'homogénéité β_2 a généralement un impact mineur sur le degré d'association β . Pour améliorer la valeur du facteur d'homogénéité, il serait intéressant d'utiliser dans son agrégation avec le facteur β_1 une méthode de moyenne pondérée pour calculer le degré d'association β . Cela permettrait de donner plus de poids au facteur d'homogénéité β_2 .

Comme les deux méthodes (Moy et Max) ont donné des résultats similaires, nous avons décidé de présenter uniquement les meilleurs taux de classification obtenus par les deux méthodes dans les essais expérimentaux suivants.

5.6.2 Comparaison des méthodes proposées avec d'autres méthodes d'ensemble

Puisque nos méthodes SIFRA et SIFRA-SAR font partie des méthodes d'ensemble floues, nous les avons comparées à d'autres méthodes d'ensemble classiques et récentes. Les résultats de nos méthodes ont été comparés aux travaux de [Cordon et al., 2008], [Cao et al., 2011] et [Nguyen et al., 2014]. Dans les travaux de [Cordon et al., 2008], les approches *Random Subspace* et *Bagging* ont été combinées pour construire un modèle de règles floues en utilisant un algorithme génétique pour sélectionner les meilleurs classifieurs. Les auteurs dans [Cao et al., 2011] utilisent un algorithme d'optimisation multi-objectif (Multi-Objective Evolutionary Algorithm : MOEA) pour sélectionner un ensemble compact des classifieurs générés. Leurs objectifs consistent à réduire la taille des classifieurs combinés et garantir la diversité (en éliminant les classifieurs similaires). Enfin, les auteurs dans [Nguyen et al., 2014] intègrent un SCBRF dans une méthode d'ensemble afin de générer des règles floues à partir des données pré-traitées : un algorithme Stacking [Ting and Witten, 1999] a été appliqué sur les données d'entrée pour générer des données pré-traitées à partir desquelles seront construites les règles floues.

Le tableau [11] montre les taux de bonne classification obtenus par ces méthodes. Les valeurs manquantes (qui n'ont pas été testées dans les documents originaux) sont marquées par "NA" (Not Available).

Par ailleurs, nous avons testé la méthode d'ensemble bien connue AdaBoost [Freund and Schapire, 1996] avec l'algorithme d'apprentissage par grille floue simple [Ishibuchi et al., 1992]. Nous avons utilisé l'algorithme AdaBoost pour construire les sous-ensembles de données et l'approche de la grille floue simple pour l'apprentissage dans chaque sous-ensemble. Les taux de bonne classification obtenus sont présentés dans le tableau [11]. Dans ce tableau, les deux méthodes que nous avons proposées, "SAR avec l'attribut Classe" et "SAR avec facteur d'homogénéité", sont notées respectivement "SAR 1" et "SAR 2".

D'après ce tableau, nous pouvons observer que les trois méthodes que nous avons proposées dans le cadre de cette thèse surpassent les autres méthodes pour la plupart des bases de données. Elles améliorent les taux de bonne classification dans les bases de données Iris, Wine, Vehicle, Diabetes, Ecoli et Balance-scale. Pour la base de données Heart-statlog, AdaBoost et notre méthode "SAR avec facteur d'homogénéité" ont abouti au même meilleur taux (80%). Une très légère amélioration par rapport à notre méthode "SAR avec Attribut Classe" a été détectée par la méthode de [Cordon et al., 2008] dans le cas de la base Glass. Enfin, nos trois méthodes donnent des résultats moins bons pour une seule base de données à savoir la base Sonar : nos méthodes donnent lieu à une performance de presque 66% dans cette base alors que les méthodes de Nguyen et al. [Nguyen et al., 2014] et Cordon et al. [Cordon et al., 2008] aboutissent à des taux de 79.33% et 80.2%, respectivement. Il serait donc envisageable d'analyser dans de prochains travaux les données de la base Sonar et les groupes d'attributs obtenus ainsi que les règles

	AdaBoost	Cordon et al. 2008	Cao et al. 2011	Nguyen et al. 2014	SIFRA	SAR 1	SAR 2
Iris	95.33	NA	95.60	96.70	97.33	97.33	97.33
Wine	91.57	NA	NA	NA	98.87	98.87	99.43
Vehicle	39.95	64.10	NA	NA	67.61	65.13	66.19
Sonar	71.63	80.20	NA	79.33	66.34	66.34	66.82
Glass	44.85	64.60	48.43	NA	62.14	64.48	63.55
Diabetes	74.35	75.60	75.29	75.40	75.26	76.95	76.56
Heart-statlog	80.00	NA	77.78	NA	77.40	79.25	80.00
Ecoli	67.55	NA	NA	NA	63.98	66.07	81.25
Balance-scale	68.80	NA	NA	NA	77.12	77.28	78.40

Tableau 11 — Comparaison des méthodes proposées avec d'autres méthodes d'ensemble

floues générées par les différentes méthodes afin de comprendre ce résultat.

En conclusion, les résultats obtenus par nos méthodes sont satisfaisants par rapport aux autres méthodes d'ensemble dans la plupart des bases testées, à l'exception de la base Sonar. Cela peut s'expliquer par le fait que nos méthodes cherchent non seulement à tirer profit de la combinaison de différents classifieurs, mais aussi à construire les sous-ensembles de données les plus appropriés, surtout dans le cas de nos méthodes SAR. En effet, dans ces méthodes, les attributs combinés dans le même sous-ensemble sont d'une part associés l'un à l'autre et d'autre part liés à l'attribut *Classe*. Ainsi, les règles de classification floues sont construites pour les sous-espaces ayant une forte association avec la classe.

5.6.3 Comparaison des méthodes proposées avec d'autres SCBRFs

Nous avons précédemment comparé les méthodes proposées SIFRA et SAR avec des méthodes d'ensemble et des méthodes de regroupement d'attributs (dans les tableaux [8](#) et [11](#)). Dans cette section, nous étendons notre étude comparative à des méthodes de classifications floues qui ne se basent pas sur le concept des ensembles d'apprentissage. D'abord, puisque nous utilisons la méthode d'apprentissage et la méthode de classification d'Ishibuchi et al. [[Ishibuchi et al., 1992](#), [Ishibuchi and Yamamoto, 2005](#)], nous avons choisi de comparer nos méthodes avec ce système basé sur une grille floue simple. Ensuite, nous nous comparons à des méthodes de classification à base de règles floues qui utilisent des approches de sélection de règles ou de sélection d'attributs pour réduire la complexité des SCBRF. Enfin, nous comparons nos résultats à des méthodes de classification floues qui utilisent des concepts de data mining, non pas pour regrouper les attributs comme notre cas, mais pour construire des règles associatives floues.

5.6.3.1 Comparaison avec la méthode d'Ishibuchi

Notre système SIFRA et ses extensions SIFRA-SAR s'inspirent fortement des méthodes de génération de règles par grille floue [Ishibuchi and Yamamoto, 2005], [Ishibuchi et al., 1992], [Ishibuchi et al., 1993], [Ishibuchi et al., 1994]. L'objectif majeur de notre travail consiste à réduire la complexité de ces systèmes. Le but de cette partie est de confirmer, à travers des tests expérimentaux, que notre méthode permet de réduire la complexité de ces systèmes (en termes de nombre de règles générées) et d'améliorer de plus la performance de classification. Nous fournissons, dans le tableau [12], une comparaison de nos méthodes SIFRA, "SAR avec l'attribut Classe" (notée *SAR 1*) et "SAR avec facteur d'homogénéité" (notée *SAR 2*) avec la méthode de classification d'Ishibuchi basée sur une grille floue simple (notée *SIF*). Les taux de bonne classification et le nombre de règles générées de cette dernière méthode sont obtenus avec le logiciel WEKA. Les bases de données utilisées dans ces tests sont ordonnées dans le tableau [12] selon un ordre croissant de leur nombre d'attributs.

En analysant le tableau [12], nous notons d'abord que nos méthodes surpassent la méthode de la grille floue simple (en terme de taux de bonne classification) dans presque toutes les bases de données à l'exception de la base Balance-scale. Une amélioration considérable est surtout notée dans le cas de la base Heart-statlog lorsque notre méthode "SAR avec facteur d'homogénéité" (*SAR 2*) est utilisée (80% contre 69.25% obtenue par la méthode *SIF*). Une amélioration significative est aussi obtenue par *SAR 2* dans le cas des bases Ecoli (81.25% contre 76.19%) et Wine (99.43% contre 92.69%). Quant à la méthode "SAR avec Attribut Classe" (*SAR 1*), elle a permis d'améliorer les taux de bonne classification dans les bases de données Diabetes et Glass.

Par ailleurs, nous notons que la génération des règles floues par la méthode d'Ishibuchi (*SIF*) est impossible dans le cas des bases avec un nombre d'attributs relativement élevé (> 18) : les bases Vehicle, Unbalanced, Ionosphere, Sonar et Mfeat-fourrier. Ce problème est dû au fait que le nombre de règles qui doivent être générées dans ces bases (égal à k^N) est très important et que leur génération nécessite un temps d'exécution très grand. Ce problème a été surmonté par nos méthodes SIFRA et SIFRA-SAR puisque le grand nombre d'attributs est divisé en des sous-ensembles de plus petit nombre ; le problème est donc divisé en des sous-problèmes de moindre complexité.

Enfin, par rapport à la grille floue simple, nos méthodes ont l'avantage de bénéficier des décisions de différents classifieurs au lieu d'une seule, ce qui peut considérablement améliorer les taux de classification.

Passons à l'analyse du nombre de règles générées par les différentes méthodes, le tableau [12] montre que nos méthodes génèrent toujours un nombre de règles réduit par rapport à la méthode *SIF*. Une réduction très intéressante paraît surtout dans le cas des bases de données Wine et Heart-statlog où la méthode "SAR avec facteur d'homogénéité" (*SAR 2*) a permis de générer 39 règles au lieu de 138 dans le cas de la base Wine et 17 règles au lieu 244 pour la base Heart-statlog. La méthode "SAR avec Attribut Classe"

	Taux de classification (%)				Nombre de règles			
	SIF	SIFRA	SAR 1	SAR 2	SIF	SIFRA	SAR 1	SAR 2
Iris	96.00	97.33	97.33	97.33	17	14	14	14
Balance-scale	89.60	77.12	77.28	78.40	68	17	17	30
Ecoli	76.19	63.98	66.07	81.25	77	20	18	47
Diabetes	72.91	75.26	76.95	76.56	116	35	40	97
Glass	60.75	62.14	64.48	63.55	39	13	24	8
Wine	92.69	98.87	98.87	99.43	138	38	38	39
Heart-statlog	69.25	77.40	79.25	80.00	244	14	13	17
Vehicle	imp	67.61	65.13	66.19	–	273	131	148
Unbalanced	imp	98.59	98.59	98.59	–	10	8	6
Ionosphère	imp	93.73	93.73	93.73	–	121	122	121
Sonar	imp	66.34	66.34	66.82	–	78	76	78
Mfeat-fourier	imp	48.15	48.15	48.70	–	253	247	285

Tableau 12 — Comparaison de SIFRA et SIFRA-SAR avec la méthode d’Ishibuchi

(*SAR 1*) permet aussi de réduire considérablement le nombre de règles générées par rapport à la méthode SIF dans les bases de données Balance-scale (17 contre 68 obtenues par SIF) et Diabetes (40 règles au lieu de 116). Nous mentionnons aussi que, dans le cas de la base Unbalanced, un nombre de règles très petit est obtenu par nos méthodes (10, 8 et 6) bien que le nombre d’attributs de cette base de données soit relativement important (32 attributs). Par contre, la méthode SIF ne réussit pas à générer les règles floues avec cette base, ni avec les cinq dernières bases ayant un nombre d’attributs respectif de 18, 32, 34, 60 et 76.

Suite à cette analyse, nous pouvons confirmer que la phase de regroupement d’attributs que nous avons développée dans cette thèse a permis de réduire considérablement la complexité des SCBRF classiques en termes du nombre de règles générées. Par ailleurs, l’idée de répartir l’information entre plusieurs apprenants et combiner les décisions locales a permis d’améliorer notablement la performance et la précision des SCBRF.

5.6.3.2 Comparaison avec des méthodes de sélection d’attributs et des méthodes de sélection de règles

Parmi les solutions les plus connues dans la littérature pour réduire le nombre de règles générées dans les SCBRF, nous trouvons les méthodes de sélection de règles d’une part et les méthodes de sélection d’attributs d’autres part (voir la section [1.3.3](#)). Notre méthode est basée sur une autre approche pour réduire la complexité des SCBRF, celle du regroupement d’attributs. Nous comparons dans cette section nos méthodes basées sur le regroupement d’attributs avec des travaux de sélection de règles et des travaux de sélection d’attributs appliqués dans les SCBRF. Les critères de comparaison sont toujours les taux de bonne classification et le nombre de règles générées.

	HGBML		SGERD		Mansouri et al.		SIFRA		SAR 1		SAR 2	
	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles
Iris	94.00	14.9	94.89	3.4	96.13	6.4	97.33	14	97.33	14	97.33	14
Ecoli	69.38	10.3	74.05	9.4	76.79	11.1	63.98	20	66.07	18	81.25	47
Diabetes	75.26	10.6	73.37	3.1	74.13	3.7	75.26	35	76.95	40	76.56	97
Glass	57.99	9.4	58.49	6.9	48.48	9	62.14	13	64.48	24	63.55	8
Wine	92.61	9.2	91.88	4.2	94.62	6.5	98.87	38	98.87	38	99.43	39
Heart-statlog	75.93	12.7	73.21	2.7	NA	NA	77.40	14	79.25	13	80.00	17
Vehicle	NA	NA	NA	NA	48.64	9.1	67.61	273	65.13	131	66.19	148
Ionosphere	NA	NA	NA	NA	86.72	6.1	93.73	121	93.73	122	93.73	121
Sonar	68.24	10.3	71.90	3.2	73.84	5.7	66.34	78	66.34	76	66.82	78

Tableau 13 — Comparaison de SIFRA et SIFRA-SAR avec des méthodes de sélection de règles et de sélection d'attributs

Nous considérons dans cette étude comparative deux méthodes de sélection de règles basées sur des algorithmes génétiques ; ces méthodes sont HGBML [Ishibuchi et al., 2005] et SGERD [Mansoori et al., 2008] [Alcala-Fdez et al., 2011]. Les résultats de ces deux méthodes appliquées sur différentes bases de données sont présentés dans le tableau 13.

Nous considérons également une méthode de sélection d'attributs qui cherchent les attributs les plus discriminants par rapport à la classe et qui utilise de plus un critère d'évaluation pour mesurer la capacité de discrimination de chaque règle floue générée [Mansoori and Shafiee, 2016].

Le tableau 13 présentent les taux de bonne classification et le nombre de règles générées par ces trois méthodes et ceux obtenus par nos méthodes SIFRA, "SAR avec l'attribut Classe" (notée *SAR 1*) et "SAR avec facteur d'homogénéité" (notée *SAR 2*). Les meilleures valeurs sont marquées en gras. Le symbole "NA" (Not Available) signifie que les valeurs correspondantes ne sont pas fournies dans les papiers d'origine. Les bases de données Balance-scale, Unbalanced et Mfeat-fourrier ne sont pas considérées dans cette étude car elles ne figurent pas dans les résultats expérimentaux de [Alcala-Fdez et al., 2011] et [Mansoori and Shafiee, 2016].

D'après le tableau 13, nous remarquons d'abord que les meilleurs taux de bonne classification sont obtenus par nos méthodes dans presque toutes les bases, à l'exception de la base Sonar. En particulier, notre méthode "SAR avec facteur d'homogénéité" (*SAR 2*) fournit des résultats très satisfaisants par rapport aux méthodes HGBML, SGERD et la méthode de [Mansoori and Shafiee, 2016] dans les bases de données Ecoli (81.25%), Heart-statlog (80%) et Wine (99.43%). Notre méthode "SAR avec Attribut Classe" (*SAR 1*) surpasse aussi considérablement les autres méthodes de sélection de règles et d'attributs dans la base de données Glass (64.48%).

En ce qui concerne le nombre de règles générées par nos méthodes, il est proche de celui obtenu par la méthode HGBML dans le cas des bases Iris, Glass et Heart-statlog. Concernant les autres bases de données, nos méthodes génèrent un nombre de règles plus important que celui fourni par HGBML, mais avec des taux de bonne classification en général plus intéressants. Ainsi, l'augmentation du nombre de règles pourrait parfois améliorer la performance de classification. Quant à la méthode SGERD et la méthode de [Mansoori and Shafiee, 2016], elles fournissent généralement des nombres de règles plus réduits que nos méthodes, mais souvent avec des taux de classification moins intéressants.

Pour conclure, les méthodes que nous avons proposées et qui se basent sur l'approche de regroupement d'attributs, montrent dans la plupart des bases testées une performance meilleure que les méthodes de sélection de règles et la méthode de sélection d'attributs. La partition des attributs en des sous-groupes d'attributs liés et surtout la prise en compte de l'information liée à la classe permet d'augmenter la performance des SCBRF. Néanmoins, le nombre de règles générées par nos méthodes peut parfois être important par rapport aux méthodes de sélection de règles, même s'il est beaucoup plus réduit que celui généré par les SIF classiques (sans utiliser la phase de regroupement d'attributs) (Tableau 12). Les méthodes de sélection de règles permettent de déterminer l'ensemble de règles les

plus discriminantes et de produire par la suite une base de règles réduite. Cependant, nous attirons l'attention sur le fait que la sélection des règles pertinentes vient en aval de la phase de génération de règles, donc le coût de la génération des règles reste inchangé. Pour cette raison, avec ces méthodes, le temps de réponse peut rester élevé (à cause de la génération de toutes les règles). Une idée intéressante consisterait à procéder par notre méthode de regroupement d'attributs pour générer un premier ensemble de règles performantes, et puis à appliquer une méthode de sélection de règles sur cet ensemble pour obtenir un ensemble plus compact de règles de classification floues sans être obligé de générer toutes les règles candidates au départ.

5.6.3.3 Comparaison avec des méthodes basées sur le concept des RA

Dans la littérature, plusieurs méthodes de classification floue ont utilisé le concept des Règles d'Association pour générer des règles de tailles différentes et ayant un pouvoir prédictif important. Ces règles sont appelées des règles de classification associatives. Puisque nous utilisons aussi le concept des RA dans nos méthodes de regroupement d'attributs, nous menons une comparaison avec les travaux les plus connus de ces méthodes de classification associative. Dans cette section, nous choisissons trois méthodes de classification basées sur les RA, les méthodes LAFAR [Hu et al., 2003], CFAR [Chen and Chen, 2008] et FARC-HD [Alcala-Fdez et al., 2011].

Le tableau [14] montre les taux de bonne classification et le nombre de règles générées par ces trois méthodes en comparaison avec nos méthodes SIFRA, "SAR avec Attribut Classe" (SAR 1) et "SAR avec facteur d'homogénéité" (SAR 2). Les valeurs des méthodes CFAR, LAFAR et FARC-HD présentées dans ce tableau sont prises des travaux de [Alcala-Fdez et al., 2011]. Seulement sept bases de données sont considérées dans ce tableau à cause de la non disponibilité des résultats des autres bases.

D'après le tableau [14], nous notons que nos méthodes fournissent des résultats beaucoup plus intéressants que la méthode LAFAR que ce soit pour le taux de bonne classification ou bien pour le nombre de règles. Une amélioration considérable de 12.2% est spécifiquement obtenue par notre méthode "SAR avec facteur d'homogénéité" dans le cas de la base de données Ecoli avec un nombre de règles aussi plus réduit (47 règles obtenues par notre méthode contre 68.4 règles par LAFAR). Notre méthode "SAR avec facteur d'homogénéité" fournit aussi un nombre de règles (8) beaucoup plus réduit que celui généré par LAFAR (36) avec la base de données Glass. En ce qui concerne la méthode CFAR, elle est aussi beaucoup moins performante que nos méthodes SIFRA et SAR dans presque toutes les bases de données. Seules les bases Heart-statlog et Sonar ont marqué la supériorité de la méthode CFAR par rapport à nos méthodes. Concernant le nombre de règles, la méthode CFAR a fourni un nombre de règles très réduit par rapport à nos méthodes dans les quatre premières bases de données. Cependant, la performance de classification de la méthode CFAR dans ces bases est faible. En outre, la méthode CFAR a aussi conduit à de grands nombre de règles dans les bases Wine (115.7) et Heart-statlog (45.3).

	LAFAR		CFAR		FARC-HD		SIFRA		SAR 1		SAR 2	
	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles	Taux	Nb. de règles
Iris	92.97	17.4	90.67	9.1	96.00	4	97.33	14	97.33	14	97.33	14
Ecoli	72.36	68.4	47.61	4	82.19	33.8	63.98	20	66.07	18	81.25	47
Diabetes	75.40	32.3	65.11	2	75.66	22.7	75.26	35	76.95	40	76.56	97
Glass	52.17	36	44.66	3.4	70.24	22.7	62.14	13	64.48	24	63.55	8
Wine	NA	NA	93.24	115.7	94.35	8.7	98.87	38	98.87	38	99.43	39
Heart-statlog	NA	NA	82.22	45.3	84.44	27	77.40	14	79.25	13	80.00	17
Sonar	NA	NA	72.48	37.3	80.19	18	66.34	78	66.34	76	66.82	78

Tableau 14 — Comparaison de SIFRA et SIFRA-SAR avec des méthodes basées sur le concept des RA

Nous considérons enfin le cas de la méthode FARC-HD, nous remarquons que cette méthode donne les meilleurs taux de bonne classification dans quatre bases de données (Ecoli, Glass, Heart-statlog et Sonar). En revanche, nos méthodes fournissent généralement un nombre de règles plus réduit dans ces bases. Par exemple, notre méthode "SAR avec Attribut Classe" génère 18 règles dans la base Ecoli et 13 règles dans la base Heart-statlog alors que FARC-HD fournit respectivement 33.8 et 27 règles pour ces deux bases. Pour les trois autres bases de données (Iris, Wine et Diabetes), nos méthodes (en particulier les méthodes SAR) surpassent la méthode FARC-HD en terme de taux de bonne classification.

En résumé, nos méthodes donnent une précision importante comparées aux autres méthodes, surtout nos méthodes SAR : elles surpassent la majorité des résultats par rapport à LAFAR, CFAR. Cependant, la méthode FARC-HD semble être meilleure que nos méthodes dans plusieurs bases de données. Cela peut s'expliquer par le fait que FARC-HD utilise un algorithme génétique pour ajuster les fonctions d'appartenance dans la partition floue et sélectionner les règles les plus précises. Comme déjà mentionné, une sélection des règles les plus pertinentes pourrait être appliquée dans notre cas afin de produire un ensemble plus compact et plus performant de règles floues.

Conclusion

Dans ce chapitre, nous avons mené différents tests expérimentaux pour évaluer la performance et l'efficacité des différentes méthodes que nous avons proposées dans le cadre de ce travail. Les méthodes ont été évaluées selon différents critères, notamment la précision de classification, le nombre de règles générées, les groupes d'attributs formés, les types des associations détectées entre les attributs ainsi que le temps de réponse du système. Les études expérimentales présentées dans ce chapitre analysent d'une part les résultats de regroupement d'attributs des différentes versions proposées de SIFRA, et comparent d'autre part ces méthodes avec d'autres méthodes d'ensemble et d'autres méthodes d'apprentissage flou de la littérature.

Les résultats expérimentaux obtenus ont prouvé une bonne performance de classification de nos méthodes par rapport à d'autres méthodes d'ensemble et de regroupement d'attributs. D'abord, notre méthode SIFRA a donné lieu à des résultats intéressants en terme de taux de bonne classification et des groupes d'attributs générés. Elle a détecté plusieurs types d'association entre les attributs y compris des corrélations linéaires et non-linéaires. En fait, la méthode SIFRA détecte les sous-espaces d'attributs qui comportent des régions denses en données. L'étude expérimentale menée a ainsi montré que la méthode SIFRA est beaucoup plus intéressante que la méthode *Random Subspace* qui regroupe les attributs d'une manière aléatoire sans suivre une heuristique de sélection spécifique ; et elle est aussi plus intéressante que la méthode SIFCO surtout pour les bases de données où il n'existe pas de corrélations linéaires entre les attributs. Dans un deuxième temps, les tests expérimentaux réalisés avec nos méthodes de regroupement d'attributs supervisées

SAR ont conduit à des résultats meilleurs que ceux obtenus par SIFRA. En effet, la prise en compte de l'information liée à la classe lors de la formation des groupes d'attributs liés permet d'améliorer la performance de classification des règles floues générées. Finalement, comparées à d'autres méthodes d'apprentissage flou, nos méthodes SIFRA et SAR ont aussi conduit à des résultats satisfaisants en terme de taux de bonne classification et du nombre de règles générées.

Il est à noter que ces résultats ne concernent que les bases testées ici, et sont donc à nuancer. Une première perspective de ce travail consiste à compléter ces tests expérimentaux avec d'autres bases de données issues de différents domaines réels. Il serait aussi intéressant d'examiner de près les résultats obtenus avec certaines bases de données comme la base Sonar où les performances ne sont pas bonnes par rapport à d'autres méthodes d'apprentissage.

Pour finir, nous envisageons, dans de prochain travaux, de mener des tests expérimentaux avec nos méthodes de regroupement d'attributs en les combinant avec d'autres méthodes d'apprentissage (autre que les systèmes d'inférence floue de [Ishibuchi et al., 1992]) tels que le système ANFIS [Jang, 1993]. En outre, d'autres méthodes d'inférence floue pourraient être utilisées dans de prochain travaux comme l'inférence par vote multiple [Ishibuchi et al., 1999].

Conclusion et perspectives

L'apprentissage automatique à partir d'exemples représente un champ de recherche ancien et riche qui est utilisé dans plusieurs disciplines telles que le domaine médical, la biologie, la robotique, etc. Un des outils les plus utilisés en apprentissage supervisé est l'apprentissage à base de règles floues. Par rapport aux autres méthodes d'apprentissage, ces systèmes (nommé SCBRF) présentent l'avantage de produire des modèles de classification intelligibles et facilement compréhensibles par l'utilisateur. Ils produisent des modèles composés de règles floues de type SI-ALORS et formés par des termes linguistiques. Un autre point fort de ces systèmes est qu'ils sont capables de traiter des données imprécises et imparfaites à travers l'utilisation de la notion des sous-ensembles flous. En revanche, ces méthodes souffrent principalement de l'inconvénient de la complexité des modèles générés lorsque les bases de données traitées possèdent un nombre important d'attributs descriptifs. L'intelligibilité des règles floues générées pourrait alors être affectée vu que leur nombre et leur taille pourraient être importants dans certains cas.

Nos travaux de recherche s'inscrivent dans le cadre des SCBRF et visent à réduire la complexité de ces systèmes dans les bases ayant un nombre d'attributs relativement important. Afin de contourner le problème du grand nombre d'attributs, nous nous sommes basés sur le concept des ensembles d'apprentissage artificiel et nous avons développé une méthode d'apprentissage supervisée basée sur l'approche de regroupement d'attributs. Notre méthode appartient aux méthodes d'ensemble qui reposent sur la décomposition du problème global en des sous-problèmes de complexité inférieure et sur la combinaison des décisions de plusieurs apprenants pour améliorer l'exécution du système global.

La méthode proposée dans le cadre de cette thèse consiste à créer des projections différentes des données d'apprentissage et à impliquer plusieurs classifieurs, chacun avec une projection différente des données, pour former un modèle de classification compact et performant. Pour cela, une phase de pré-traitement de données est proposée pour décomposer l'ensemble des attributs en des sous-groupes d'attributs liés. Bien que la méthode de regroupement d'attributs proposée dans le cadre de cette thèse ait été appliquée dans le cadre des SCBRF, elle pourrait être utilisée dans d'autres méthodes ou d'autres domaines d'apprentissage. Cette méthode peut donc être considérée comme une méthode de regroupement d'attributs ou de recherche d'associations entre des attributs numériques.

Nos contributions

Nos contributions résident précisément dans cette phase de pré-traitement. Nous avons étudié différentes méthodes de regroupement d'attributs et proposé dans cette thèse deux méthodes principales de regroupement d'attributs. Ces méthodes sont inspirées du travail de Soua et al. [Soua et al., 2013] dans la mesure où les attributs qui sont associés sont regroupés et traités ensemble.

La plupart des méthodes de la littérature qui cherchent à détecter des dépendances entre les attributs sont basées sur des mesures statistiques. Par exemple, la méthode SIFCO de [Soua et al., 2013] utilise une mesure de corrélation linéaire (la mesure Pearson) pour détecter les attributs liés. Dans le cadre de cette thèse, nous avons toutefois proposé de recourir à des concepts de data mining, notamment le concept des Règles d'Association et de génération des itemsets fréquents pour détecter des associations intéressantes entre des attributs numériques. Nos méthodes ne cherchent pas une forme spécifique de liaison (telle qu'une liaison linéaire ou parabolique,...), mais elles détectent la condensation des données dans des régions de l'espace. Cela rend nos méthodes capables de détecter différents types d'associations entre les attributs. De plus, elles sont capables de traiter des attributs aussi bien numériques que catégoriques.

La première méthode proposée dans cette thèse est nommée SIFRA. Elle utilise le concept de génération des itemsets fréquents pour détecter les co-occurrences de données dans des régions denses de l'espace. Les groupes d'attributs qui forment des sous-espaces riches en régions denses sont sélectionnés. Des tests expérimentaux menés dans le cadre de cette thèse ont prouvé que notre méthode SIFRA est plus performante que plusieurs autres méthodes de regroupement d'attributs comme *Random Subspace* et SIFCO. Ces tests ont aussi confirmé que la méthode SIFRA est capable de détecter différents types d'association entre les attributs, y compris les corrélations linéaires et non linéaires. En fait, SIFRA ne cherche pas une forme spécifique de dépendance, mais elle se focalise sur la concentration des données dans des régions de l'espace.

Les groupes d'attributs formés par SIFRA reflètent des associations intéressantes entre les attributs descriptifs, mais ils ne considèrent aucune information liée à la classe. Notre deuxième contribution consiste alors à proposer une méthode de regroupement d'attributs supervisé, nommée SIFRA-SAR. Elle cherche à réunir les attributs qui possèdent des associations intéressantes entre eux en considérant également les informations liées à la classe. Nous avons pris en compte ces informations de deux manières différentes. La première méthode que nous avons proposée est dénommée "SAR avec Attribut Classe", elle incorpore l'attribut *Classe* dans la procédure de génération des itemsets fréquents. Ainsi, des groupes d'attributs qui contiennent la classe sont alors formés. Quant à la deuxième méthode, dénommée "SAR avec facteur d'homogénéité", elle évalue l'homogénéité des régions denses et privilégie les groupes d'attributs qui contiennent des régions non seulement denses mais également homogènes.

Les études expérimentales ont montré la supériorité de ces méthodes de regroupement

d'attributs supervisé par rapport à la première méthode proposée SIFRA. Ce résultat a confirmé que le choix des groupes d'attributs liés a un effet considérable sur la performance de classification ; plus précisément, le choix des groupes d'attributs associés à la classe aide à améliorer l'efficacité des règles floues générées.

Afin d'évaluer la performance des méthodes que nous avons proposées dans le cadre de cette thèse, nous avons mené une étude comparative avec d'autres méthodes d'ensemble et d'autres méthodes de classification floues. Cette étude a confirmé notre hypothèse que la décomposition de l'ensemble des attributs de départ en des groupes d'attributs liés améliore, dans la majorité des cas, la performance de la méthode de classification. Elle réduit ainsi le coût d'exécution et la taille des bases de règles générées.

Notre étude comparative a touché aussi des méthodes de classification basées sur l'approche de sélection de règles et d'autres basées sur la sélection des attributs les plus significatifs. Cette étude a montré la supériorité de notre approche de regroupement d'attributs en termes de taux de bonne classification, pour la plupart des bases. En revanche, le nombre de règles générées par notre approche est plus important que celui obtenu par les méthodes de sélection de règles.

Perspectives

Tout au long de l'élaboration de nos travaux de recherche, nous avons souligné plusieurs points d'approfondissement et certaines perspectives intéressantes. Nous présentons ici quelques pistes de recherche en distinguant les perspectives à court terme de celles à plus long terme.

Perspectives à court terme

Parmi les études qui peuvent être envisagées à court terme, nous citons les perspectives suivantes.

- Le paramétrage automatique :

La méthode de regroupement d'attributs que nous avons proposée nécessite le choix de certains paramètres d'entrée dont les principaux sont : le seuil de support $minsupp$ et le seuil d'association β_{min} . Dans ce travail, nous avons étudié et testé plusieurs valeurs possibles de ces deux seuils. Ainsi, nous avons observé que le choix des meilleurs seuils dépend fortement des données utilisées. Afin de mieux comprendre ce résultat, nous envisageons d'étudier de près la relation entre ces seuils et les caractéristiques des données utilisées. Par ailleurs, une perspective intéressante consiste à proposer une méthode de paramétrage qui détermine d'une manière automatique les valeurs appropriées de ces seuils. Dans ce cadre, une méthode à base d'algorithme génétique pourrait être proposée [Wendt et al., 2010, De Almeida et al., 2009].

- L'utilisation d'autres méthodes pour la génération des règles floues :

Dans la phase de génération de règles, deux étapes sont requises : une partition floue de l'espace des entrées en des sous-espaces et la construction d'une règle floue pour chaque sous-espaces flou. Pour la partition floue, nous avons adopté la partition floue supervisée qui considère l'information liée à la classe lors de la partition des attributs numériques. Concernant la construction des règles floues, nous avons adopté la méthode heuristique d'Ishibuchi [Ishibuchi et al., 1992, Ishibuchi and Yamamoto, 2005]. Nous avons toutefois présenté dans le chapitre 1 d'autres heuristiques pour le calcul du degré de certitude CF des règles, notamment celles basées sur la mesure Confiance [Ishibuchi and Yamamoto, 2005]. Nous avons par ailleurs mentionné que certaines de ces heuristiques peuvent notablement améliorer la performance de classification (section 1.3.1.2). Il serait donc intéressant de tester ces heuristiques pour le calcul du degré de certitude comme perspective de notre travail.

- La méthode d'inférence par vote multiple :

Comme méthode d'inférence floue, nous avons adopté la méthode de la seule règle gagnante. Cette méthode considère la conclusion de la règle la plus représentative. Cependant, plusieurs règles peuvent contribuer à la classification d'une même donnée surtout que les règles générées sont de prémisses différentes ; chaque règle représente une partie de l'information et participe ainsi à la phase de classification. Pour cette raison, l'utilisation de la méthode du vote multiple présente une perspective intéressante de ce travail.

Perspectives à long terme

Nous évoquons ici quelques pistes de recherche qui pourraient, à long terme, étendre ce travail et améliorer les résultats obtenus dans le cadre de cette thèse.

- Application à d'autres méthodes d'apprentissage :

Les approches de regroupement d'attributs, basées sur le concept des RA, que nous avons proposées dans cette thèse ont été testées dans le cadre des systèmes de classification par grille floue. Cependant, il serait intéressant de tester ces méthodes avec d'autres systèmes de classification. Cette idée a été appliquée avec le système SUCRAGE dans le cadre d'un projet de maîtrise [Zaatour et al., 2016]. Dans ce travail, notre méthode de regroupement d'attributs a été utilisée avec le système de classification SUCRAGE qui est un système d'apprentissage à base de règles non-floues. L'étude expérimentale a montré que notre approche de regroupement d'attributs a permis d'améliorer les résultats de classification. De plus, l'utilisation de notre méthode a permis d'étendre la méthode SUCRAGE à des données qualitatives, chose qui n'était pas possible avec la méthode de regroupement d'attributs classique de SUCRAGE.

ANFIS [Jang et al., 1997, Karaboga and Kaya, 2018] est aussi un autre système d'apprentissage qui pourrait être combiné avec notre méthode de regroupement. En effet, ANFIS est un système d'inférence floue basé sur les réseaux de neurones adaptatifs qui cherchent à construire des règles floues avec des fonctions d'appartenance appropriées. Nous pensons que la combinaison d'une méthode de regroupement d'attributs qui cherche à former les groupes d'attributs liés avec une méthode d'apprentissage adaptative qui cherche à construire les fonctions d'appartenance appropriées peut donner lieu à un modèle de classification très performant.

L'intégration de la méthode de regroupement d'attributs dans d'autres méthodes d'apprentissage comme les colonies de fourmis [Dorigo and Gambardella, 1997] peut aussi être envisagée.

- Combinaison de la méthode de regroupement d'attributs avec une méthode de sélection de règles :

Parmi les travaux de la littérature qui cherchent à réduire la complexité des SCBRF, nous avons cité les méthodes de sélection de règles. Ces méthodes cherchent à construire un ensemble compact et performant de règles floues en éliminant les règles redondantes et non-significatives. Les méthodes de sélection de règles sont généralement appliquées en aval de la phase de génération des règles, ce qui ne résout pas le problème du coût de la phase de génération des règles, mais garanti une réduction considérable du nombre de règles final.

Comparées à certaines méthodes de sélection de règles, nos méthodes SIFRA et SIFRA-SAR ont prouvé une haute performance de classification dans la plupart des cas. Cependant, le nombre de règles générées par nos approches est généralement plus important que celui obtenu par les méthodes de sélection de règles.

Ce résultat nous incite, en perspective, à ajouter une dernière phase de sélection de règles dans notre système SIFRA (ou SIFRA-SAR). De ce fait, l'utilisation de la phase de regroupement d'attributs garantit la réduction du coût de la phase de génération des règles et fournit une première base de règles réduite. Ensuite, la phase de sélection de règles permettrait de sélectionner, à partir de ces règles candidates, les règles les plus pertinentes pour réduire le nombre final de règles floues tout en gardant une haute performance de classification du système.

- Combinaison de la méthode de regroupement d'attributs avec une méthode de sélection/extraction d'attributs :

L'approche de sélection ou d'extraction d'attributs est une autre approche très utilisée pour réduire la complexité des SCBRF. Ces méthodes sont utilisées pour contourner le problème de la haute dimension en éliminant les attributs redondants ou inutiles. Une idée intéressante qui pourrait être envisagée dans de prochains travaux consiste à combiner une approche de sélection/extraction d'attributs avec notre méthode de regroupement d'attributs. La combinaison de ces deux méthodes peut nous aider à élargir notre champ de travail et nous permettre de traiter des bases de données

plus grandes que celles testées dans ce travail. Ainsi, l'approche de sélection (ou d'extraction) d'attributs permettrait de réduire la dimension importante du problème traité, puis la démarche de regroupement d'attributs permettrait de réduire davantage la dimension ainsi que le nombre de règles générées et la taille de leurs prémisses.

- Bagging et regroupement d'attributs pour traiter de grandes bases :

Notre approche de regroupement d'attributs permet de contourner le problème du grand nombre d'attributs dans les SCBRF. Notre méthode appartient ainsi aux méthodes d'ensemble qui divisent les attributs descriptifs en des sous-groupes d'attributs de tailles réduites. Nous avons présenté dans le chapitre 2 une autre catégorie de méthodes d'ensemble qui divisent l'ensemble des observations (et non pas des attributs) en des sous-bases de tailles réduites. Ces méthodes permettent de traiter les bases qui possèdent un grand nombre d'observations. Une perspective intéressante de notre travail consiste en la combinaison d'une de ces méthodes (comme le Bagging ou Ada-Boost) avec notre méthode de regroupement d'attributs. Cela permettrait de traiter les bases de données qui possèdent un grand nombre d'attributs d'une part et un grand nombre d'observations d'autre part.

L'idée de combiner la méthode d'ensemble *Bagging* avec une méthode de regroupement d'attributs (la méthode *Random Subspace*) a été déjà proposée dans [Cordon et al., 2008]. Cependant, les auteurs ont affirmé que la sélection aléatoire des attributs (avec la méthode *Random Subspace*) n'est pas toujours significative et qu'une heuristique de sélection pourrait être plus intéressante. Dans ce contexte, nous pouvons étendre le travail de [Cordon et al., 2008] en utilisant notre méthode de regroupement d'attributs à la place de la méthode *Random Subspace*.

Bibliographie

- [Agrawal et al., 1999] Agrawal, R., Gehrke, J. E., Gunopulos, D., and Raghavan, P. (1999). Automatic subspace clustering of high dimensional data for data mining applications. US Patent 6,003,029.
- [Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2) :207–216.
- [Agrawal and Srikant, 1994] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *20th int. conf. very large data bases, VLDB.*, volume 1215, pages 487–499.
- [Alcalá et al., 2007] Alcalá, R., Gacto, M. J., Herrera, F., and Alcalá Fdez, J. (2007). A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(05) :539–557.
- [Alcalá-Fdez et al., 2009] Alcalá-Fdez, J., Alcalá, R., Gacto, M. J., and Herrera, F. (2009). Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160(7) :905–921.
- [Alcala-Fdez et al., 2011] Alcala-Fdez, J., Alcala, R., and Herrera, F. (2011). A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy Systems*, 19(5) :857–872.
- [Alhamzi et al., 2014] Alhamzi, A., Nasr, M., and Salama, S. (2014). A comparative study of association rules algorithms on large databases. *Egyptian Computer Science Journal*, 38(3).
- [Amouzadi and Mirzaei, 2010] Amouzadi, A. and Mirzaei, A. (2010). Hierarchical fuzzy rule-based classification system by evolutionary boosting algorithm. In *2010 5th International Symposium on Telecommunications*. IEEE.
- [Antonelli et al., 2014] Antonelli, M., Ducange, P., and Marcelloni, F. (2014). A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. *Information Sciences*, 283 :36–54.

- [Arlot, 2018] Arlot, S. (2018). *Apprentissage statistique et données massives*, chapter Validation croisée. Editions Technip.
- [Bay, 2001] Bay, S. D. (2001). Multivariate discretization for set mining. *Knowledge and Information Systems*, 3(4) :491–512.
- [Bellman, 1961] Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton University Press.
- [Benediktsson et al., 2003] Benediktsson, J., Pesaresi, M., and Arnason, K. (2003). Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9) :1940–1949.
- [Berka and Rauch, 2010] Berka, P. and Rauch, J. (2010). Machine learning and association rules. In *19th Int. Conf. On Computational Statistics COMPSTAT*.
- [Besse, 2009] Besse, P. (2009). Apprentissage statistique and data mining. Technical report, Institut de Mathématiques de Toulouse, Laboratoire de Statistique et Probabilités, Institut National des Sciences Appliquées de Toulouse.
- [Bharati and Kanchan, 2016] Bharati, K. and Kanchan, D. (2016). Comparative study of frequent itemset mining algorithms : Fp growth, fin, prepost + and study of efficiency in terms of memory consumption, scalability and runtime. *Int. Journal of Technical Research and Applications*, 4 :72–77.
- [Borgi, 1999] Borgi, A. (1999). *Apprentissage supervisé par génération de règles : le système SUCRAGE*. PhD thesis, Université Paris 6.
- [Borgi, 2008] Borgi, A. (2008). Un raisonnement approximatif pour l’apprentissage supervisé de règles. In *AAFD*, pages 85–106.
- [Borgi and Akdag, 2001] Borgi, A. and Akdag, H. (2001). Knowledge based supervised fuzzy classification : An application to image processing. *Annals of Mathematics and Artificial Intelligence, Special Issue : Representation of Uncertainty*, 32 :67–86.
- [Born and Schmidt-Thieme, 2004] Born, S. and Schmidt-Thieme, L. (2004). Optimal discretization of quantitative attributes for association rules. In *Classification, Clustering, and Data Mining Applications*, pages 287–296. Springer Berlin Heidelberg.
- [Bouchon-Meunier, 1995] Bouchon-Meunier, B. (1995). *La logique floue et ses applications*.
- [Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2) :123–140.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45 :5–32.

- [Breiman et al., 1984] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Chapman and Hall, New York, NY.
- [Brin et al., 1997] Brin, S., Motwani, R., and Silverstein, C. (1997). Beyond market baskets. *ACM SIGMOD Record*, 26(2) :265–276.
- [Burdick et al., 2001] Burdick, D., Calimlim, M., and Gehrke, J. (2001). Mafia : A maximal frequent itemset algorithm for transactional databases. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 443–452. IEEE.
- [Busarov et al., 2016] Busarov, V., Grafeeva, N., and Mikhailova, E. (2016). A comparative analysis of algorithms for mining frequent itemsets. In *Communications in Computer and Information Science*, pages 136–150. Springer International Publishing.
- [Cao et al., 2011] Cao, J., Wang, H., Kwong, S., and Li, K. (2011). Combining interpretable fuzzy rule-based classifiers via multi-objective hierarchical evolutionary algorithm. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE.
- [Carbonell et al., 1983] Carbonell, J. G., Michalski, R., and Mitchell, T. M. (1983). *Machine learning : An artificial intelligence approach.*, chapter An overview of machine learning. Tioga Publishing Company.
- [Catlett, 1991] Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *European working session on learning*, pages 164–178. Springer.
- [Chandrashekar and Sahin, 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1) :16–28.
- [Chao and Li, 2005] Chao, S. and Li, Y. (2005). Multivariate interdependent discretization for continuous attribute. In *Third International Conference on Information Technology and Applications (ICITA05)*, volume 1, pages 167–172. IEEE.
- [Chaudhari et al., 2014] Chaudhari, S., Patil, M., and Bambhori, J. (2014). Study and review of fuzzy inference systems for decision making and control. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, 14(147) :88–92.
- [Chen and Chen, 2008] Chen, Z. and Chen, G. (2008). Building an associative classifier based on fuzzy association rules. *International Journal of Computational Intelligence Systems*, 1(3) :262.
- [Chimerge, 1992] Chimerge, K. R. (1992). Discretization of numeric attributes. In Press, A., editor, *Tenth national conference on Artificial intelligence*, pages 123–128.
- [Cordón et al., 1999] Cordón, O., del Jesus, M. J., and Herrera, F. (1999). A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20(1) :21–45.

- [Cordon et al., 2008] Cordon, O., Quirin, A., and Sanchez, L. (2008). A first study on bagging fuzzy rule-based classification systems with multicriteria genetic selection of the component classifiers. In *2008 3rd International Workshop on Genetic and Evolving Systems*. IEEE.
- [Cornillon and Matzner-Lober, 2007] Cornillon, P.-A. and Matzner-Lober, É. (2007). *Régression : Théorie et applications*. Springer.
- [Cornuéjols et al., 2003] Cornuéjols, A., Miclet, L., and Kodrato, Y. (2003). *Apprentissage artificiel concepts et algorithmes*.
- [Cramer, 1999] Cramer, H. (1999). *Mathematical methods of statistics*. Princeton University Press.
- [Crémilleux, 1991] Crémilleux, B. (1991). *Induction automatique : aspects théoriques, le système ARBRE, Applications en médecine*. PhD thesis, Université Joseph Fourier-Grenoble I.
- [Cruz and Wishart, 2006] Cruz, J. A. and Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics*, 2 :117693510600200.
- [Cutler et al., 2007] Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY. *Ecology*, 88(11) :2783–2792.
- [D’Andrea and Lazzerini, 2013] D’Andrea, E. and Lazzerini, B. (2013). A hierarchical approach to multi-class fuzzy classifiers. *Expert Systems with Applications*, 40(9) :3828–3840.
- [De Almeida et al., 2009] De Almeida, G., Salles, J., et al. (2009). Optimal tuning parameters of the dynamic matrix predictive controller with constraints. *Latin American applied research*, 39(1) :33–40.
- [Dehzangi et al., 2007] Dehzangi, O., Zolghadri, M. J., Taheri, S., and Fakhrahmad, S. M. (2007). Efficient fuzzy rule generation : A new approach using data mining principles and rule weighting. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*. IEEE.
- [Del Jesus et al., 2004] Del Jesus, M. J., Hoffmann, F., Navascués, L. J., and Sánchez, L. (2004). Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3) :296–308.
- [Deng et al., 2012] Deng, Z., Wang, Z., and Jiang, J. (2012). A new algorithm for fast mining frequent itemsets using n-lists. *Science China Information Sciences*, 55(9) :2008–2030.
- [Deng and Lv, 2014] Deng, Z.-H. and Lv, S.-L. (2014). Fast mining frequent itemsets using nodesets. *Expert Systems with Applications*, 41(10) :4505–4512.

- [Deng and Lv, 2015] Deng, Z.-H. and Lv, S.-L. (2015). PrePost : An efficient n-lists-based algorithm for mining frequent itemsets via children–parent equivalence pruning. *Expert Systems with Applications*, 42(13) :5424–5432.
- [Dietterich, 2000] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15. Springer Berlin Heidelberg.
- [Dietterich and Bakiri, 1995] Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2 :263–286.
- [Ding and Peng, 2005] Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02) :185–205.
- [Dorigo and Gambardella, 1997] Dorigo, M. and Gambardella, L. (1997). Ant colony system : a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1) :53–66.
- [Fakhrahmad et al., 2007] Fakhrahmad, S. M., Zare, A., and Jahromi, M. Z. (2007). Constructing accurate fuzzy rule-based classification systems using apriori principles and rule-weighting. In *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, pages 547–556. Springer Berlin Heidelberg.
- [Farahbod, 2012] Farahbod, F. (2012). Comparison of different t-norm operators in classification problems. *International Journal of Fuzzy Logic Systems*, 2(3) :33–39.
- [Fayyad and Irani, 1993] Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous valued attributes for classification learning. In *In 13th Int. Joint Conf. on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann. San Francisco, CA.
- [Ferchichi et al., 2009] Ferchichi, S. E., Laabidi, K., and Zidi, S. (2009). Genetic algorithm and tabu search for feature selection. *Studies in Informatics and control*, 18(2) :181–187.
- [Feuilloy, 2009] Feuilloy, M. (2009). *Etude d’algorithmes d’apprentissage artificiel pour la prédiction de la syncope chez l’homme*. PhD thesis, Université d’Angers.
- [Fleuret, 2004] Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5 :1531–1555.
- [Freund, 1995] Freund, Y. (1995). Boosting a weak learning algorithm by majority. In *Information and computation*, pages 202–216.
- [Freund and Schapire, 1996] Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of International Conference on Machine Learning*, pages 148–156.

- [Galar et al., 2011] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., and Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems : Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8) :1761–1776.
- [García et al., 2010] García, M. N. M., Lucas, J. P., Batista, V. F. L., and Martín, M. J. P. (2010). Multivariate discretization for associative classification in a sparse data application domain. In *Lecture Notes in Computer Science*, pages 104–111. Springer Berlin Heidelberg.
- [Garg and Gulia, 2015] Garg, R. and Gulia, P. (2015). Comparative study of frequent itemset mining algorithms apriori and FP growth. *International Journal of Computer Applications*, 126(4) :8–12.
- [Ghahramani, 2004] Ghahramani, Z. (2004). *Advanced lectures on machine learning*, chapter Unsupervised learning, pages 72–112. Springer Berlin Heidelberg.
- [Goethals and Zaki, 2003] Goethals, B. and Zaki, M., editors (2003). *IEEE ICDM Workshop Frequent Itemsets Mining Implementations*, volume 80.
- [Gonzalez and Perez, 1998] Gonzalez, A. and Perez, R. (1998). Completeness and consistency conditions for learning fuzzy rules. *Fuzzy Sets and Systems*, 96(1) :37–51.
- [Gouda and Zaki, 2001] Gouda, K. and Zaki, M. J. (2001). Efficiently mining maximal frequent itemsets. In *icdm*, page 163. IEEE.
- [Grahne and Zhu, 2003] Grahne, G. and Zhu, J. (2003). High performance mining of maximal frequent itemsets. In *6th International Workshop on High Performance Data Mining*, volume 16, page 34.
- [Grahne and Zhu, 2005] Grahne, G. and Zhu, J. (2005). Fast algorithms for frequent itemset mining using fp-trees. *IEEE transactions on knowledge and data engineering*, 17(10) :1347–1362.
- [Gupta et al., 2010] Gupta, A., Mehrotra, K. G., and Mohan, C. (2010). A clustering-based discretization for supervised learning. *Statistics & Probability Letters*, 80(9-10) :816–824.
- [Guyon and Elisseev, 2003] Guyon, I. and Elisseev, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, pages 1157–1182.
- [Gyorodi et al., 2004] Gyorodi, C., Gyorodi, R., and Holban, S. (2004). A comparative study of association rules mining algorithms. In *Hungarian Joint Symposium on Applied Computational Intelligence*, Oradea.
- [Hadzic et al., 2011] Hadzic, F., Tan, H., and Dillon, T. S. (2011). Mining maximal and closed frequent subtrees. In *Mining of Data with Complex Structures*, pages 191–199. Springer Berlin Heidelberg.

- [Han et al., 1997] Han, E., Karypis, G., and Kumar, V. (1997). Min-apriori : An algorithm for finding association rules in data with continuous attributes. Technical Report TR-97-068, Department of Computer Science, University of Minnesota.
- [Han et al., 2012] Han, J., Pei, J., and Kamber, M. (2012). *Data mining : concepts and techniques. Third Edition.* Elsevier.
- [Han et al., 2000] Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2) :1–12.
- [Havens et al., 2012] Havens, T. C., Bezdek, J. C., Leckie, C., Hall, L. O., and Palaniswami, M. (2012). Fuzzy c-means algorithms for very large data. *IEEE Transactions on Fuzzy Systems*, 20(6) :1130–1146.
- [He and Tan, 2012] He, H. and Tan, Y. (2012). A two-stage genetic algorithm for automatic clustering. *Neurocomputing*, 81 :49–59.
- [Hefny et al., 2010] Hefny, H., Ghiduk, A., Wahab, A., and Elashiry, M. (2010). Effective method for extracting rules from fuzzy decision trees based on ambiguity and classifiability. *Universal Journal of Computer Science and Engineering Technology*, 1(1) :55–63.
- [Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8) :832–844.
- [Hoffmann, 2001] Hoffmann, F. (2001). Boosting a genetic fuzzy classifier. In *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*. IEEE.
- [Hong et al., 2001] Hong, T.-P., Kuo, C.-S., and Chi, S.-C. (2001). Trade-off between computation time and number of rules for fuzzy mining from quantitative data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 09(05) :587–604.
- [Hong and Lee, 1996] Hong, T.-P. and Lee, C.-Y. (1996). Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems*, 84(1) :33–47.
- [Hou, 1996] Hou, W.-C. (1996). Extraction and applications of statistical relationships in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, 8(6) :939–945.
- [Hu et al., 2003] Hu, Y.-C., Chen, R.-S., and Tzeng, G.-H. (2003). Finding fuzzy classification rules using data mining techniques. *Pattern Recognition Letters*, 24(1-3) :509–519.
- [Hyvarinen and Oja, 2000] Hyvarinen, A. and Oja, E. (2000). Independent component analysis : algorithms and applications. *Neural Networks*, 13(4-5) :411–430.

- [Ishibuchi et al., 1997] Ishibuchi, H., Murata, T., and Turksen, I. (1997). Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems*, 89(2) :135–150.
- [Ishibuchi and Nakashima, 2001] Ishibuchi, H. and Nakashima, T. (2001). Effect of rule weights in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy systems*, 9(4) :504–515.
- [Ishibuchi et al., 1999] Ishibuchi, H., Nakashima, T., and Morisawa, T. (1999). Voting in fuzzy rule-based systems for pattern classification problems. *Fuzzy Sets and Systems*, 103(2) :223–238.
- [Ishibuchi et al., 1995a] Ishibuchi, H., Nakashima, T., and Murata, T. (1995a). A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems. In *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*. IEEE.
- [Ishibuchi et al., 2001a] Ishibuchi, H., Nakashima, T., and Murata, T. (2001a). Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences*, 136(1-4) :109–133.
- [Ishibuchi et al., 1992] Ishibuchi, H., Nozaki, K., and Tanaka, H. (1992). Distributed representation of fuzzy rules and its application to pattern classification. *Fuzzy Sets and Systems*, 52(1) :21–32.
- [Ishibuchi et al., 1993] Ishibuchi, H., Nozaki, K., and Tanaka, H. (1993). Efficient fuzzy partition of pattern space for classification problems. *Fuzzy Sets and Systems*, 59(3) :295–304.
- [Ishibuchi et al., 1994] Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H. (1994). Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets and Systems*, 65(2-3) :237–253.
- [Ishibuchi et al., 1995b] Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H. (1995b). Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 3(3) :260–270.
- [Ishibuchi and Yamamoto, 2002] Ishibuchi, H. and Yamamoto, T. (2002). Effect of fuzzy discretization in fuzzy rule-based systems for classification problems with continuous attributes. *Archives of Control Sciences*, 12(4) :351–378.
- [Ishibuchi and Yamamoto, 2004] Ishibuchi, H. and Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, 141(1) :59–88.

- [Ishibuchi and Yamamoto, 2005] Ishibuchi, H. and Yamamoto, T. (2005). Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 13(4) :428–435.
- [Ishibuchi et al., 2001b] Ishibuchi, H., Yamamoto, T., and Nakashima, T. (2001b). Fuzzy data mining : effect of fuzzy discretization. In *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE Comput. Soc.
- [Ishibuchi et al., 2005] Ishibuchi, H., Yamamoto, T., and Nakashima, T. (2005). Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 35(2) :359–365.
- [Jang, 1993] Jang, J.-S. (1993). ANFIS : adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3) :665–685.
- [Jang et al., 1997] Jang, J. S., Sun, C., and Mizutani, E. (1997). Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence. *USA : Prentice Hall*, pages 333–393.
- [Jiang and Wang, 2011] Jiang, Y. and Wang, J. (2011). An improved association rules algorithm based on frequent item sets. *Procedia Engineering*, 15 :3335–3340.
- [Karaboga and Kaya, 2018] Karaboga, D. and Kaya, E. (2018). Adaptive network based fuzzy inference system (anfis) training approaches : a comprehensive survey. *Artificial Intelligence Review*, pages 1–31.
- [Kenny, 1979] Kenny, D. A. (1979). *Correlation and causation*. New York : Wiley.
- [Khalid et al., 2014] Khalid, S., Khalil, T., and Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*. IEEE.
- [Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.
- [Kohavi and Quinlan, 2002] Kohavi, R. and Quinlan, J. R. (2002). Data mining tasks and methods : Classification : decision-tree discovery. In *Handbook of data mining and knowledge discovery*, pages 267–276. Oxford University Press, Inc.
- [Kolen and Pollack, 1991] Kolen, J. F. and Pollack, J. B. (1991). Back propagation is sensitive to initial conditions. *Advances in neural information processing systems*, pages 860–867.
- [Kononenko et al., 1996] Kononenko, I., Robnik-Sikonja, M., and Pompe, U. (1996). Relief for estimation and discretization of attributes in classification, regression, and ilp problems. *Artificial intelligence : methodology, systems, applications*, pages 31–40.

- [Kotsiantis and Kanellopoulos, 2006] Kotsiantis, S. and Kanellopoulos, D. (2006). Discretization techniques : A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1) :47–58.
- [Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6).
- [Krawczyk et al., 2015] Krawczyk, B., Woźniak, M., and Herrera, F. (2015). On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. *Pattern Recognition*, 48(12) :3969–3982.
- [Kvalseth, 1987] Kvalseth, T. O. (1987). Entropy and correlation : Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3) :517–519.
- [Ladha and Deepa, 2011] Ladha, L. and Deepa, T. (2011). Feature selection methods and algorithms. *International journal on computer science and engineering*, 3(5) :1787–1797.
- [Lazar et al., 2012] Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., and Nowe, A. (2012). A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4) :1106–1119.
- [Lazkano and Sierra, 2003] Lazkano, E. and Sierra, B. (2003). Bayes-nearest : a new hybrid classifier combining bayesian network and distance based algorithms. In *Portuguese Conference on Artificial Intelligence*, pages 171–183, Berlin, Heidelberg. Springer.
- [Lee and Verleysen, 2007] Lee, J. A. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer Science and Business Media.
- [Leondes, 2000] Leondes, C. T. (2000). *Knowledge-Based Systems, Four-Volume Set : Techniques and Applications*. Elsevier.
- [Li and Shang, 2013] Li, X. and Shang, J. (2013). A novel apriori algorithm based on cross linker. In *Lecture Notes in Electrical Engineering*, pages 449–456. Springer London.
- [Li et al., 2017] Li, Y., Xu, J., Yuan, Y.-H., and Chen, L. (2017). A new closed frequent itemset mining algorithm based on gpu and improved vertical structure. *Concurrency and Computation : Practice and Experience*, 29(6) :e3904.
- [Liebertrau, 1983] Liebertrau, A. M. (1983). *Measures of association*, volume 32. Sage.
- [Lorigo and Govindaraju, 2006] Lorigo, L. M. and Govindaraju, V. (2006). Oine arabic handwriting recognition : a survey. *IEEE transactions on pattern analysis and machine intelligence*, 28(5) :712–724.

- [Maclin and Shavlik, 1995] Maclin, R. and Shavlik, J. W. (1995). Combining the predictions of multiple classifiers : Using competitive learning to initialize neural networks. In *IJCAI*, pages 524–531.
- [MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics and Probability*, pages 281–297, Berkeley, CA : University of California Press.
- [Mansoori and Shafiee, 2016] Mansoori, E. G. and Shafiee, K. S. (2016). On fuzzy feature selection in designing fuzzy classifiers for high-dimensional data. *Evolving Systems*, 7(4) :255–265.
- [Mansoori et al., 2008] Mansoori, E. G., Zoldhadri, M. J., and Katebi, S. D. (2008). Sgerd : A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on Fuzzy Systems*, 16(4) :1061–1071.
- [Mazieres, 2016] Mazieres, A. (2016). *Cartographie de l'apprentissage artificiel et de ses algorithmes*. PhD thesis, Université Paris Diderot - Ecole Doctorale Frontières du Vivant.
- [Mehta et al., 2005] Mehta, S., Parthasarathy, S., and Yang, H. (2005). Toward unsupervised correlation preserving discretization. *IEEE Transactions on Knowledge and Data Engineering*, 17(9) :1174–1185.
- [Mena et al., 2012] Mena, L. J., Orozco, E. E., Felix, V. G., Ostos, R., Melgarejo, J., and Maestre, G. E. (2012). Machine learning approach to extract diagnostic and prognostic thresholds : Application in prognosis of cardiovascular mortality. *Computational and Mathematical Methods in Medicine*, 2012 :1–6.
- [Menger, 1942] Menger, K. (1942). Statistical metrics. In *Proceedings of the National Academy of Sciences of the U.S.A.*, volume 28, pages 535–537.
- [Mitchell, 1997] Mitchell, T. M. (1997). *Machine learning*, volume 45. Burr Ridge, IL : McGraw Hill.
- [Mitra and Hayashi, 2000] Mitra, S. and Hayashi, Y. (2000). Neuro-fuzzy rule generation : survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3) :748–768.
- [Monti and Cooper, 1999] Monti, S. and Cooper, G. F. (1999). A latent variable model for multivariate discretization. In *AISTATS*.
- [Moreno et al., 2007] Moreno, M. N., Segrera, S., Lopez, V. F., and Polo, M. J. (2007). Improving the quality of association rules by preprocessing numerical data. In *Il Congreso Espanola Informatics*, pages 223–230.
- [Moshe, 2006] Moshe, D. (2006). *Human Sensitivity to Mutual Information*. PhD thesis, Graduate School-New Brunswick Rutgers, The State University of New Jersey.

- [Muhlenbach and Rakotomalala, 2002] Muhlenbach, F. and Rakotomalala, R. (2002). Multivariate supervised discretization, a neighborhood graph approach. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 314–321. IEEE Comput. Soc.
- [MutaherBa-Alwi, 2013] MutaherBa-Alwi, F. (2013). Knowledge acquisition tool for learning membership function and fuzzy classification rules from numerical data. *International Journal of Computer Applications*, 64(13) :24–30.
- [Naim et al., 2011] Naim, P., Wullemmin, P. H., Leray, P., Pourret, O., and Becker, A. (2011). *Réseaux bayésiens*.
- [Nguyen et al., 2013] Nguyen, C. H., Pedrycz, W., Duong, T. L., and Tran, T. S. (2013). A genetic design of linguistic terms for fuzzy rule based classifiers. *International Journal of Approximate Reasoning*, 54(1) :1–21.
- [Nguyen et al., 2014] Nguyen, T. T., Liew, A. W.-C., To, C., Pham, X. C., and Nguyen, M. P. (2014). Fuzzy if-then rules classifier on ensemble data. In *Communications in Computer and Information Science*, pages 362–370. Springer Berlin Heidelberg.
- [Nguyen-Tuong and Peters, 2011] Nguyen-Tuong, D. and Peters, J. (2011). Model learning for robot control : a survey. *Cognitive Processing*, 12(4) :319–340.
- [Nojima and Ishibuchi, 2006] Nojima, Y. and Ishibuchi, H. (2006). Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion. In *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS06)*. IEEE.
- [Nozaki et al., 1994] Nozaki, K., Ishibuchi, H., and Tanaka, H. (1994). Selecting fuzzy rules with forgetting in fuzzy classification systems. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*. IEEE.
- [Nozaki et al., 1996] Nozaki, K., Ishibuchi, H., and Tanaka, H. (1996). Adaptive fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 4(3) :238–250.
- [Ou and Murphey, 2007] Ou, G. and Murphey, Y. L. (2007). Multi-class pattern classification using neural networks. *Pattern Recognition*, 40(1) :4–18.
- [Pal and Chakraborty, 2001] Pal, N. and Chakraborty, S. (2001). Fuzzy rule extraction from ID3-type decision trees for real data. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 31(5) :745–754.
- [Parsons et al., 2004] Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data : a review. *Acm Sigkdd Explorations Newsletter*, 6(1) :90–105.
- [Pasquier, 2000a] Pasquier, N. (2000a). *Data Mining : algorithmes d’extraction et de réduction des règles d’association dans les bases de données*. Theses, Université Blaise Pascal - Clermont-Ferrand II.

- [Pasquier, 2000b] Pasquier, N. (2000b). Extraction de bases pour les règles d'association à partir des itemsets fermés fréquents. In *INFORSID'2000 Congress*, pages 56–77.
- [Pasquier et al., 1999] Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory, ICDT '99*, pages 398–416, London, UK, UK. Springer-Verlag.
- [Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical magazine*, (2) :559–572.
- [Pearson, 1926] Pearson, K. (1926). On the coefficient of radical likeness. *Biometrika*, 18 :105–117.
- [Pei et al., 2007] Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., and Yang, D. (2007). Hmine : Fast and space-preserving frequent pattern mining in large databases. *IIE Transactions*, 39(6) :593–605.
- [Pei et al., 2000] Pei, J., Han, J., Mao, R., et al. (2000). Closet : An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD workshop on research issues in data mining and knowledge discovery*, volume 4, pages 21–30.
- [Peng et al., 2005] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8) :1226–1238.
- [Polikar, 2006] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3) :21–45.
- [Prudhomme and Lallich, 2007] Prudhomme, E. and Lallich, S. (2007). Ensemble prÃ©dicteur fondé sur les cartes auto-organisatrices adapté aux données volumineuses. In *EGC*, pages 473–484.
- [Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1) :81–106.
- [Quinlan, 1993] Quinlan, J. R. (1993). *C4. 5 : programs for machine learning*. Morgan kaufmann.
- [Rifkin and Klautau, 2004] Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5 :101–141.
- [Rokach, 2009] Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks : A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53(12) :4046–4072.
- [Rokach and Maimon, 2015] Rokach, L. and Maimon, O. (2015). *Data mining with decision trees : theory and applications (2nd Edition)*, volume 81. World scientific.

- [Russel and Norvig, 2002] Russel, S. J. and Norvig, P. (2002). *Artificial intelligence : a modern approach*.
- [Sagi and Rokach, 2018] Sagi, O. and Rokach, L. (2018). Ensemble learning : A survey. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 8(4) :e1249.
- [Saporta, 2006] Saporta, G. (2006). *Probabilités, analyse des données et statistique*.
- [Savesere et al., 1995] Savesere, A., Omiecinski, E., and Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. In *Proceedings of 20th International Conference on VLDB*, pages 432–444.
- [Schweizer and Sklar, 1960] Schweizer, B. and Sklar, A. (1960). Statistical metric spaces. *Pacific Journal of Mathematics*, 10(1) :313–334.
- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication, part i, part ii. *Bell System Technical Journal*, 27 :623–656.
- [Skurichina and Duin, 2002] Skurichina, M. and Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2) :121–135.
- [Songram, 2012] Songram, P. (2012). Mining closed itemsets : A review. *International Journal of Advancements in Computing Technology*, 4(5) :181–188.
- [Soua, 2009] Soua, B. (2009). Regroupement d’attributs et partition floue supervisée dans les systèmes d’inférence floue. Master’s thesis, Université de Manouba.
- [Soua et al., 2013] Soua, B., Borgi, A., and Tagina, M. (2013). An ensemble method for fuzzy rule-based classification systems. *Knowledge and Information Systems*, 36(2) :385–410.
- [Srikant and Agrawal, 1996] Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD 96*. ACM Press.
- [Stéphane, 2012] Stéphane, T. (2012). *Data Mining et statistique décisionnelle : L’intelligence des données*. Editions Technip.
- [Tan et al., 2005] Tan, P. N., Steinbach, M., and Kumar, V. (2005). Association analysis : basic concepts and algorithms. *Introduction to Data mining*, pages 327–414.
- [Ting and Witten, 1999] Ting, K. M. and Witten, I. H. (1999). Issues in stacked generalization. *Journal of artificial intelligence research*, 10 :271–289.
- [Toussaint, 1980] Toussaint, G. T. (1980). The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4) :261–268.

- [Valentini and Masulli, 2002] Valentini, G. and Masulli, F. (2002). Ensembles of learning machines. In *13th Italian workshop on neural nets*, pages 3–19, Berlin. Springer.
- [Walia et al., 2015] Walia, N., Singh, H., and Sharma, A. (2015). Anfis : Adaptive neuro-fuzzy inference system-a survey. *International Journal of Computer Applications*, 123(13).
- [Wang et al., 2015] Wang, X., Liu, X., Pedrycz, W., and Zhang, L. (2015). Fuzzy rule based decision trees. *Pattern Recognition*, 48(1) :50–59.
- [Wendt et al., 2010] Wendt, K., Cortés, A., and Margalef, T. (2010). Knowledge-guided genetic algorithm for input parameter optimisation in environmental modelling. *Procedia Computer Science*, 1(1) :1367–1375.
- [Yang et al., 2009] Yang, Y., Wang, G. I., and Wu, X. (2009). *Data mining and knowledge discovery handbook*, chapter Discretization methods, pages 101–116. Springer, Boston, MA.
- [Yao, 2003] Yao, Y. Y. (2003). Information-theoretic measures for knowledge discovery and data mining. In *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, pages 115–136. Springer Berlin Heidelberg.
- [Ye and Ji, 2009] Ye, J. and Ji, S. (2009). Discriminant analysis for dimensionality reduction : An overview of recent developments. In *Biometrics*, pages 1–19. John Wiley & Sons, Inc.
- [Yu and Liu, 2003] Yu, I. and Liu, H. (2003). Feature selection for high-dimensional data : a fast correlation-based filter solution. In *Proceedings of the 20th International Conference on Machine Learning*, volume 3, pages 856–863.
- [Zaatour et al., 2016] Zaatour, R., Borgi, A., and Slima, I. B. (2016). Attributes regrouping by association rules in SUCRAGE. In *2016 11th International Conference on Intelligent Systems : Theories and Applications (SITA)*. IEEE.
- [Zadeh, 1965] Zadeh, L. (1965). Fuzzy set. *Information and Control*, 8 :338–353.
- [Zadeh, 1978] Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100 :9–34.
- [Zadeh, 1975] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information sciences*, 8(3) :199–249.
- [Zaki and Hsiao, 2002] Zaki, M. J. and Hsiao, C.-J. (2002). Charm : An efficient algorithm for closed itemset mining. In *Proceedings of the 2002 SIAM international conference on data mining*, pages 457–473. SIAM.

- [Zaki et al., 1997] Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997). New algorithms for fast discovery of association rules. In *Proceedings of the 3rd Int. Conf. on Knowledge Discovery and Data Mining, KDD'97*, pages 283–286.
- [Zhao et al., 2012] Zhao, J., Yu, P. L., Shi, L., and Li, S. (2012). Separable linear discriminant analysis. *Computational Statistics & Data Analysis*, 56(12) :4290–4300.
- [Zhou et al., 2013] Zhou, S., He, S., and Wang, W. (2013). Improved apriori for continuous attributes. *International Journal of Information Science and Intelligent System*, 2 :37–43.
- [Zhu and Goldberg, 2009] Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. 3 :1–130.

A

La logique floue

La logique floue a été introduite par Zadeh en 1965, elle consiste à représenter, formaliser et traiter les informations de nature imparfaite [Zadeh, 1965]. La logique floue introduit un aspect graduel à la logique classique et vise à imiter la perception et le raisonnement de l'être humain. Le raisonnement humain est un raisonnement approximatif qui prend en compte la nature imparfaite des données du monde réel. Cette imperfection des données ou des connaissances peut être de plusieurs natures. Bouchon-Meunier énumère trois natures principales de l'imperfection des connaissances dans son ouvrage [Bouchon-Meunier, 1995] :

- Les imprécisions : elles peuvent être dues soit à une mauvaise connaissance des valeurs numériques suite à l'insuffisance des instruments de mesure par exemple, soit à l'utilisation des termes de langage naturel pour une expression spontanée de la connaissance (grand, proche, jeune).
- Les incertitudes : les données qu'on traite peuvent passer par une source intermédiaire qui est susceptible de commettre des erreurs lors de la transmission de la connaissance ; il peut y avoir un doute sur la validité de l'information («je crois que...») ou carrément une modification de l'information.
- Les incomplétudes : on peut également être confronté au cas où des données sont absentes ou incomplètes.

La théorie du flou regroupe diverses théories dont nous citons : la théorie des sous-ensembles flous qui est une généralisation des sous-ensembles classiques [Zadeh, 1965] ; la théorie des quantités floues qui généralise celle des nombres et intervalles [Zadeh, 1975] ; la théorie des possibilités qui étend celle des probabilités [Zadeh, 1978] ; ...

Les sous-ensembles flous

Un sous-ensemble flou est une généralisation du sous-ensemble classique dans le sens où on introduit un aspect graduel (nuance) de l'appartenance d'un élément au sous-ensemble. Généralement, dans le monde réel, les sous-ensembles n'ont pas de limites précises ; par

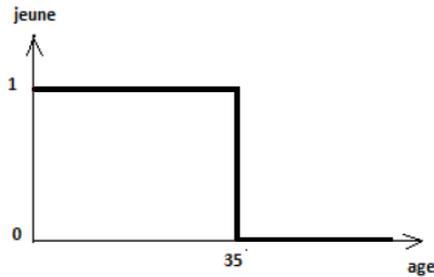


Figure 51 — Sous-ensemble classique



Figure 52 — Sous-ensemble flou

exemple on utilise les termes : petit, jeune, beau, ... qui ne peuvent pas être modélisés par des intervalles aux limites précises. Cette notion de sous-ensemble flou vise à éviter le passage brusque d'une classe à une autre et permettre l'appartenance partielle à chacune avec un degré spécifique [Bouchon-Meunier, 1995]. Soit un domaine de valeurs X , les sous-ensembles classiques et flous sont définis comme suit [Bouchon-Meunier, 1995] :

- Un sous-ensemble A classique est défini par une fonction caractéristique f_A qui prend la valeur 0 pour les éléments n'appartenant pas à A et la valeur 1 pour les éléments qui appartiennent à A (voir exemple dans [51]).

$$f_A : X \rightarrow \{0, 1\} \quad (63)$$

- Un sous-ensemble flou A est caractérisé par une fonction d'appartenance μ_A qui associe à chaque élément x de X un degré compris entre 0 et 1 (voir exemple dans [52]).

$$\mu_A : X \rightarrow [0, 1] \quad (64)$$

Les opérateurs flous

L'introduction des sous-ensembles flous nécessite l'utilisation d'opérateurs spécifiques. Ces opérateurs sont équivalents à ceux définis pour les sous-ensembles classiques : l'inclusion, l'intersection, l'union et le complément (Figure [53]).

- On dit qu'un sous-ensemble A est inclus dans un sous-ensemble B de X si tout élément x de X qui appartient, même de façon modérée, à A appartient aussi à B avec un degré plus grand ou égal :

$$\forall x \in X \quad \mu_A(x) \leq \mu_B(x) \quad (65)$$

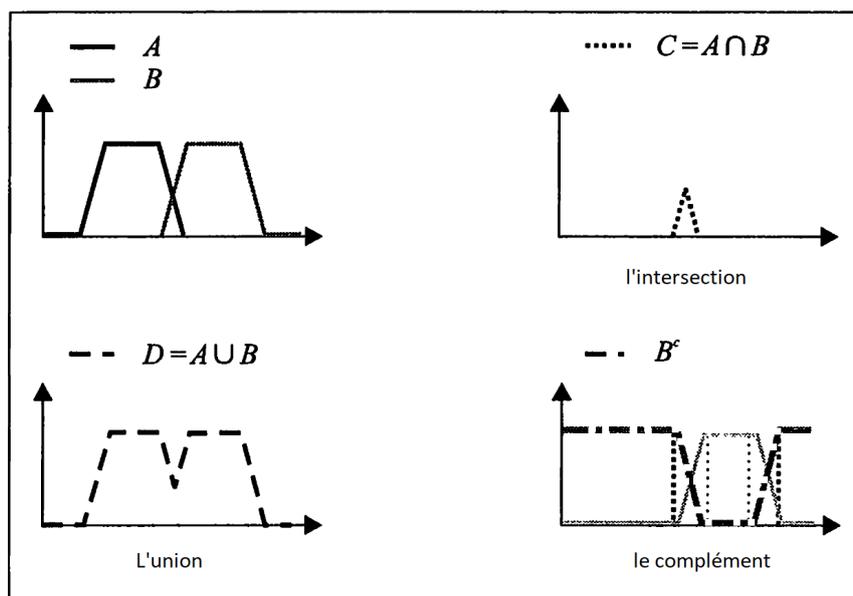


Figure 53 — Exemples d'intersection, d'union et de complément de sous-ensembles flous

- L'intersection de deux sous-ensembles A et B de X est le sous-ensemble C constitué des éléments de X qui vérifient :

$$\forall x \in X \quad \mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (66)$$

- L'union de deux sous-ensembles A et B de X est le sous-ensemble D constitué des éléments de X tels que :

$$\forall x \in X \quad \mu_D(x) = \max(\mu_A(x), \mu_B(x)) \quad (67)$$

- Le complément B^C d'un sous-ensemble B de X est défini comme suit :

$$\forall x \in X \quad \mu_{B^C}(x) = 1 - \mu_B(x) \quad (68)$$

Nous avons défini ci-dessus l'intersection, l'union et le complément d'un sous-ensemble flou par les opérateurs \min , \max et la complémentation à 1, respectivement. Néanmoins, il existe d'autres opérateurs pour définir ces notions. Ces opérateurs sont les normes triangulaires (t-norme) pour l'intersection, les conormes triangulaires pour l'union (t-conorme) et les négations pour le complément. Le tableau [15](#) présente les principales t-normes et t-conormes.

Nom	T-norme	T-conorme	Négation
Zadeh	$\min(\mu_A, \mu_B)$	$\max(\mu_A, \mu_B)$	$1 - \mu_A$
Probabiliste	$\mu_A \times \mu_B$	$\mu_A + \mu_B - \mu_A \times \mu_B$	$1 - \mu_A$
Lukasiewicz	$\max(\mu_A + \mu_B - 1, 0)$	$\min(\mu_A + \mu_B, 1)$	$1 - \mu_A$
Hamacher ($\gamma > 0$)	$\frac{\mu_A \mu_B}{\gamma + (1 - \gamma)(\mu_A + \mu_B - \mu_A \mu_B)}$	$\frac{\mu_A + \mu_B - \mu_A \mu_B - (1 - \gamma)\mu_A \mu_B}{1 - (1 - \gamma)\mu_A \mu_B}$	$1 - \mu_A$
Drastique	μ_A si $\mu_B = 1$ μ_B si $\mu_A = 1$ 0 sinon	μ_A si $\mu_B = 0$ μ_B si $\mu_A = 0$ 1 sinon	$1 - \mu_A$

Tableau 15 — Principales T-normes et T-conormes

B

Présentation des plateformes WEKA, SPMF et SIFCO

Nous donnons dans cette partie une brève description des différentes plateformes utilisées pour la réalisation de notre application SIFRA : les plateformes WEKA, SPMF et SIFCO.

La plateforme WEKA

WEKA (Waikato Environnement for Knowledge Analysis) est une plateforme de fouille de données qui contient un ensemble d'algorithmes d'apprentissage écrits en Java. Cette plateforme, gratuite et libre, représente l'un des outils de Data mining les plus utilisés par les chercheurs. Elle contient des outils de pré-traitement de données, de classification supervisée et non supervisée, de régression, de découverte des règles d'association ainsi que des outils de visualisation.

Les algorithmes présents dans WEKA peuvent être directement utilisés à travers l'interface graphique ou bien appelés dans un code Java à travers l'API fournie gratuitement.

Pour le développement de notre système SIFRA, nous avons eu recours à l'API de WEKA et nous avons intégré, dans notre propre code, le code de quelques algorithmes implémentés dans WEKA, notamment l'algorithme Apriori et les algorithmes de discrétisation régulière (EWI) et supervisée (MDLP).

Par ailleurs, quelques projets qui visent à étendre le logiciel WEKA à des versions destinées à des domaines spécifiques sont également fournis dans le site officiel de WEKA, comme par exemple l'extension de WEKA à des algorithmes flous (*FuzzyWeka*), la version conçue pour les biologistes (*BioWeka*), le Framework dédié à l'extraction de flux de données (*MOA*), ...

FuzzyWeka fournit une implémentation du système de classification à base de règles floues de Ishibuchi et al. [\[Ishibuchi et al., 1992\]](#), qui est basé sur la grille floue simple. Nous avons utilisé le code et la documentation gratuite du logiciel FuzzyWeka pour l'im-

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%   (a) Creator: R.A. Fisher
%   (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%   (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Figure 54 — En-tête d'un fichier .arff

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

Figure 55 — Données d'un fichier .arff

plémentation de notre système SIFRA.

Les données d'apprentissage utilisées comme entrée de notre système SIFRA sont stockées dans un fichier ayant une structure spécifique. Nous avons utilisé la même structure de données traitée par WEKA afin de pouvoir intégrer les algorithmes de WEKA d'une manière simple. En effet, WEKA traite des données décrites dans des fichiers enregistrés sous le format *ARFF*. Un fichier .arff est un fichier texte qui décrit une liste d'instances qui partagent un ensemble d'attributs descriptifs. Un fichier .arff est composé de deux sections différentes : une première section pour la description des informations sur la base de données, telles que le nom de la base, les noms des attributs et leurs types (voir exemple décrit dans la figure [54](#)) et une deuxième section pour les données (voir exemple de la figure [55](#)).

La bibliothèque SPMF

La plateforme WEKA fournit seulement quelques algorithmes de génération d'itemsets fréquents, notamment Apriori et FP-Growth. Cependant, nous envisageons d'utiliser d'autres algorithmes de génération d'itemsets comme les algorithmes de génération d'itemsets fermés et maximaux, notamment FPclose et FPmax. Pour cela, nous avons eu recours

à une autre bibliothèque, spécialisée dans la découverte d'itemsets dans les bases de données. Cette bibliothèque, nommée SPMF, est aussi gratuite et fournit un code libre écrit en Java.

La bibliothèque SPMF offre l'implémentation de 150 algorithmes de Data mining, notamment des algorithmes de génération de règles d'association, de découverte d'itemsets fréquents, d'itemsets fermés, séquentiels, périodiques, etc.

Le code source des algorithmes fournis dans cette bibliothèque peut être facilement intégré dans d'autres logiciels Java. En outre, une documentation bien détaillée des différents algorithmes et de la manière de les utiliser est fournie dans le site officiel de cette bibliothèque.

Le système SIFCO

Pour l'implémentation de notre méthode SIFRA, nous avons aussi utilisé le système SIFCO implémenté par Soua [Soua et al., 2013] en langage Java. La phase de regroupement d'attributs par recherche de corrélation linéaire, la phase de génération de règles par partition floue simple et aussi partition floue supervisée, ainsi que la phase de classification par la méthode de la seule règle gagnante ont tous été implémentés dans SIFCO. Nous avons utilisé ce code et l'avons étendu pour mettre en place notre système SIFRA.

Le système SIFCO possède aussi une interface graphique qui permet à l'utilisateur de fixer les différents paramètres d'entrée et lui fournir un récapitulatif détaillé des résultats de classification (la base de règles générée, les taux de bonne classification, la matrice de corrélation générée, ...). Nous avons aussi utilisé cette interface et l'avons amélioré pour qu'elle englobe toutes les méthodes présentés dans ce travail et pour qu'elle permet de facilement choisir les différentes versions et paramètres de notre méthode SIFRA.

L'architecture générale de notre système SIFRA est présentée dans la figure [56](#). Les trois phases de SIFRA sont développées dans trois modules séparés :

- Module de regroupement d'attributs
- Module de génération de règles (qui représente la phase d'apprentissage)
- Module d'inférence floue (qui représente la phase de classification)

Les différentes contributions de cette thèse résident dans le premier module "module de regroupement d'attributs". L'interface graphique développée permet à l'utilisateur de choisir la méthode de regroupement d'attributs (SIFRA, SAR avec attribut Classe ou SAR avec facteur d'homogénéité), l'algorithme de génération des itemsets (Apriori, FP-Growth, FPclose, FPmax), la méthode de discrétisation pour le regroupement d'attributs (régulière, supervisée ou supervisée-multivariée) et le choix de la partition floue (supervisée ou simple). L'utilisateur peut aussi, à travers l'interface, fixer les valeurs des paramètres d'entrée (les seuils $minsupp$ et β_{min}).

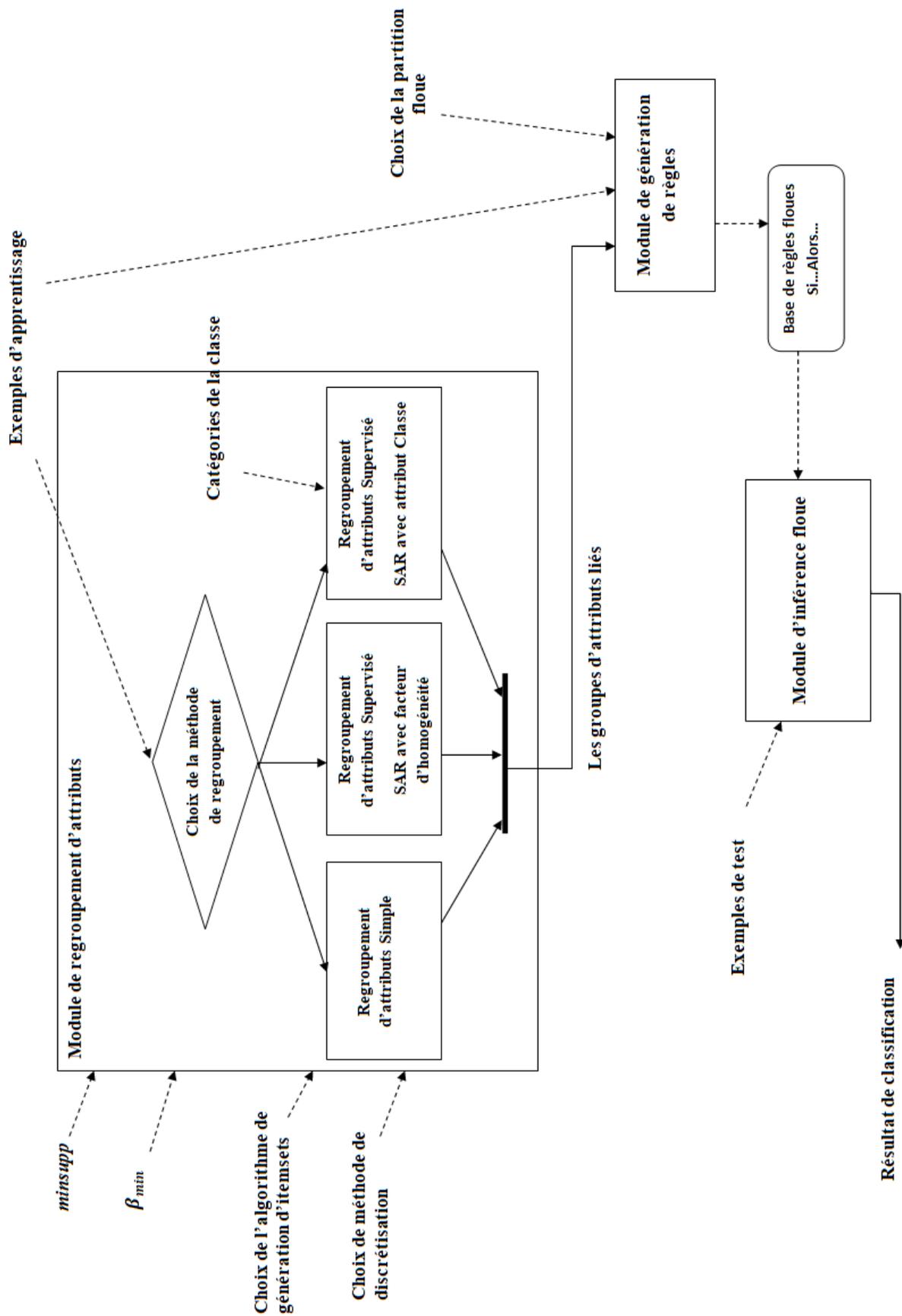


Figure 56 — Architecture générale du système SIFRA

Étude des critères de sélection des groupes d'attributs finaux

Nous rappelons que la méthode de regroupement d'attributs de SIFRA possède trois parties principales :

- Génération des itemsets fréquents
- Détermination des attributs liés
- Sélection des groupes d'attributs finaux.

Nous nous focalisons sur la troisième phase, la phase de sélection des groupes d'attributs finaux et nous présentons ici une étude expérimentale des critères de sélection des groupes d'attributs finaux. Nous avons choisi, dans l'ordre, les deux critères de sélection suivants :

1. le degré d'association β
2. les groupes qui contiennent le plus d'attributs

De plus, nous avons choisi de sélectionner les groupes d'attributs finaux de façon à former une partition de l'ensemble des attributs de départ. Pour cela, chaque fois qu'on sélectionne le meilleur groupe d'attributs, la suppression de tous les groupes en intersection avec le meilleur groupe est effectuée.

Le choix de ces deux critères, ainsi que la partition, ont été discutés dans la section [3.5](#). Afin de mieux argumenter cette discussion et de renforcer nos choix, nous menons dans cette partie des tests expérimentaux qui comparent les étapes et critères de sélection que nous avons choisis dans le chapitre [3](#) avec d'autres méthodes de sélection possibles. Le premier point qui a été évoqué dans la section [3.5](#) consiste à discuter si le fait de former une partition des groupes d'attributs est indispensable. Pour voir si ce choix est pertinent, nous testons la méthode de sélection avec les mêmes deux critères de sélection mais sans former nécessairement une partition. En d'autres termes, l'étape de suppression

des groupes d'attributs qui sont en intersection avec le groupe sélectionné est éliminée. Seuls les groupes d'attributs qui sont strictement inclus dans le groupe sélectionné sont supprimés car ces groupes possèdent une association redondante qui a été déjà prise en compte. Ainsi, les groupes d'attributs finaux retenus peuvent se chevaucher entre eux mais ne peuvent pas être inclus entre eux. La deuxième question qui a été discutée dans la partie [3.5](#) est la pertinence du choix des groupes d'attributs les plus longs. Pour répondre à cette question, nous développons deux autres méthodes de sélection. La première méthode prend en compte un seul critère de sélection qui est le critère β (le deuxième critère qui cherche si le groupe choisi est inclus dans un autre groupe plus long n'est pas pris en compte). La deuxième méthode prend en compte le critère qui correspond à la longueur des groupes mais le prend en premier lieu, c'est à dire avant le critère de β . En d'autres termes, nous commençons par analyser les groupes les plus longs, nous choisissons parmi eux le groupe ayant le β le plus grand et nous supprimons tous les groupes qui sont en intersection avec lui. Dans le paragraphe [3.5](#), un autre point a été également discuté ; celui qui concerne les attributs qui ne figurent dans aucun groupe d'attributs liés. Nous pensons que le fait qu'un attribut n'apparaît dans aucun groupe d'attributs liés et ne reflète aucune région dense ne signifie pas qu'il est inutile pour la phase de classification. Pour cela, nous avons proposé d'ajouter chacun de ces attributs dans un groupe seul et de le traiter comme un attribut indépendant. Afin de confirmer notre intuition, nous testons dans cet annexe notre méthode de sélection sans ajouter les attributs "ratés".

Pour résumer, cinq méthodes de sélection sont testées et comparées dans cette partie :

- M1 : la première est notre méthode proposée dans [3.3.3](#) (la méthode SIFRA). Elle consiste donc à utiliser les deux critères de sélection (β puis la longueur des groupes) et aussi à garantir la formation d'une partition de l'ensemble des attributs de départ.
- M2 : la deuxième méthode utilise les mêmes étapes que la première mais sans ajouter les attributs "ratés" (qui n'apparaissent dans aucun groupe d'attributs liés).
- M3 : la troisième correspond à l'utilisation des mêmes critères que SIFRA mais sans former une partition.
- M4 : la quatrième consiste à considérer un seul critère : le groupe ayant le β le plus grand.
- M5 : finalement la cinquième méthode consiste à commencer par les groupes les plus longs puis à choisir, parmi ces groupes, le groupe ayant le β le plus grand

Nous fournissons dans le tableau [16](#) les taux de bonne classification et le nombre de règles (entre parenthèses) obtenus par les cinq méthodes de sélection différentes appliquées sur différentes bases de données. Nous fournissons aussi dans le tableau [17](#) les groupes d'attributs obtenus par les différentes méthodes pour certaines bases de données. Pour mener ces tests expérimentaux, nous avons fixé les valeurs suivantes des paramètres d'entrée :

	M1	M2	M3	M4	M5
Iris	96.00 (18)	96.00 (18)	94.00 (41)	94.66 (17)	96.00 (18)
Wine	98.87 (39)	98.31 (45)	96.62 (111)	97.75 (44)	97.75 (44)
Vehicle	63.94 (139)	64.18 (117)	62.52 (787)	62.17 (113)	63.94 (177)
Sonar	66.34 (76)	61.05 (29)	66.34 (1150)	66.34 (75)	66.34 (76)
Glass	57.47 (14)	51.86 (9)	56.54 (38)	59.81 (23)	57.47 (14)
Diabetes	72.01 (28)	72.01 (28)	74.60 (49)	71.61 (23)	72.01 (28)
Heart-statlog	77.40 (14)	78.14 (11)	80.00 (58)	72.22 (14)	75.55 (15)
Ecoli	59.22 (15)	59.22 (15)	58.33 (56)	50.29 (13)	54.76 (15)
Balance-scale	77.12 (17)	77.12 (17)	76.16 (51)	77.12 (17)	77.12 (17)
Ionosphere	91.73 (121)	89.74 (9)	91.16 (130)	91.73 (121)	91.73 (121)

Tableau 16 — Méthodes de sélection des groupes d'attributs finaux : comparaison des taux de bonne classification

- méthode de regroupement : SIFRA
- algorithme de génération des itemsets fréquents : FP-Growth
- méthode de discrétisation pour la phase de regroupement d'attributs : régulière à 3 intervalles
- *minsupp* : pour chaque base de données, nous avons fixé une valeur unique de *minsupp* utilisée par les cinq tests :
 - 0.1 pour les plus petites bases : Iris, Balance-scale et Ecoli
 - 0.3 pour les bases de dimension moyenne : Diabetes, Glass, Wine, Heart-statlog et Vehicle
 - 0.5 et 0.6 pour les plus grandes bases à savoir Ionosphere et Sonar
- β_{min} : 0.7
- méthode de partition floue : supervisée

D'après le tableau [16](#), nous remarquons d'abord que notre méthode SIFRA (M1) fournit les meilleurs taux de bonne classification dans la majorité des bases de données. Concernant le nombre de règles générées, la deuxième méthode (M2), celle qui ignore les attributs manqués, génère le plus petit nombre de règles. Ceci s'explique par le fait qu'une partie des attributs descriptifs a été ignorée. Le tableau [17](#) confirme que certains attributs ont été ignorés par cette méthode dans le cas des bases de données Vehicle (les attributs [0] [6] [8] [14] [17]), Glass (les attributs [2] [6]) et Heart-statlog (les attributs [0] et [2]). Cependant, en utilisant cette méthode, la performance de classification s'est

considérablement dégradée par rapport à notre méthode qui considère tous les attributs, et ceci dans plusieurs bases de données comme les bases Sonar, Glass et Ionosphere. Ce résultat confirme notre hypothèse que les attributs "ratés" peuvent avoir un grand pouvoir discriminant et être utile pour la phase de classification.

On s'intéresse maintenant à la troisième méthode (M3) qui garde tous les groupes d'attributs liés sans chercher à construire une partition de l'ensemble des attributs de départ. Ce qu'on peut remarquer pour cette méthode c'est qu'elle génère un nombre de règles important par rapport à toutes les autres méthodes. Ce nombre de règles pourrait être très grand dans certains cas tels que les bases Vehicle (787) et Sonar (1150). Ce grand nombre de règles était prévisible vu que cette méthode génère plusieurs groupes d'attributs supplémentaires, comme le montre le tableau [17](#). Voyons par exemple les bases Vehicle et Heart-statlog, le tableau [17](#) montre qu'un plus grand nombre de groupes d'attributs ont été construits par cette méthode M3 ("Sans partition") par rapport à toutes les autres méthodes.

En comparant la méthode M4 (où β est le seul critère considéré) avec la méthode SIFRA (M1), nous observons que les nombres de règles générées par les deux méthodes sont très proches. De plus, les mêmes taux de classification sont obtenus dans les bases de données Sonar, Balance-scale et Ionosphere. Pour le reste des bases (à l'exception de la base Glass), nous constatons une dégradation des taux de bonne classification avec la méthode de sélection qui ne considère que le critère β . Ceci confirme que la prise en compte des groupes les plus longs améliore la performance de classification. En analysant les groupes d'attributs dans le tableau [17](#), nous remarquons que cette méthode génère des groupes contenant deux attributs au maximum. Les groupes de plusieurs attributs n'ont pas été détectés parce que généralement ils possèdent des valeurs faibles du degré d'association β .

Voyons enfin la méthode de sélection qui commence par les groupes les plus longs (M5). Nous remarquons que cette méthode fournit presque les mêmes taux de bonne classification que SIFRA (M1) sauf dans les bases Wine, Heart-statlog et Ecoli. Dans ces trois dernières bases, les taux de bonne classification sont considérablement diminués par la méthode M5. Le nombre de règles générées par cette méthode est similaire à celui généré par la méthode SIFRA dans toutes les bases de données sauf la base Vehicle. Dans cette dernière base, un nombre de règles plus important a été généré par rapport à SIFRA (177 contre 139). Le tableau [17](#) montre qu'un groupe d'attributs long (qui contient 6 attributs) a été généré par cette méthode alors que des groupes d'attributs moins longs (contenant 3 ou 4 attributs) ont été générés par SIFRA.

Pour conclure, les critères de sélection adoptés dans SIFRA (méthode M1) aboutissent aux meilleurs résultats de classification dans la majorité des cas. La méthode M2 améliore les taux de bonne classification dans un seul cas (la base Vehicle) mais les résultats sont considérablement détériorés dans la plupart des autres bases. La méthode M3, quant à elle, semble intéressante en terme de taux de bonne classification dans seulement les bases Diabète et Heart-statlog. Cependant, elle présente l'inconvénient de générer un

grand nombre de règles dans presque toutes les bases à cause des groupes d'attributs supplémentaires qu'elle génère. De son côté, la méthode M4 est plus intéressante que M1 dans seulement le cas de la base Glass. En revanche, une dégradation des taux de bonne classification a été remarquée dans la plupart des autres bases. Finalement, la méthode M5 ne fournit en aucun cas des résultats meilleurs que M1.

M1 (SIFRA)	M2 Ignorer les attributs "ratés"	M3 Sans partition	M4 seul critère β	M5 Commencer par les groupes longs
Iris	[0, 2, 3] [1]	[0, 2, 3] [1, 2, 3] [0, 1]	[2, 3] [0, 1]	[0, 2, 3] [1]
Vehicle	[3, 4, 5, 13] [7, 10, 11] [15] [2] [9, 12] [1] [16] [0] [6] [8] [14] [17]	[3, 4, 5, 13] [4, 5, 7, 10, 11, 13] [4, 5, 13, 15] [2, 4, 5, 7, 13] [4, 5, 12, 13] [4, 5, 16] [3, 10] [4, 5, 9, 13] [1, 4, 5, 13] [3, 4, 5, 7] [1, 4, 5, 9] [9, 12] [0] [6] [8] [14] [17]	[3, 4] [5, 10] [11, 13] [2, 7] [1, 9] [12] [15] [16] [0] [6] [8] [14] [17]	[4, 5, 7, 10, 11, 13] [1, 9] [12] [15] [16] [2] [3] [0] [6] [8] [14] [17]
Glass	[3, 4, 5, 7] [1] [8] [0] [2] [6]	[3, 4, 5, 7] [3, 5, 7, 8] [3, 4, 5, 8] [1, 5] [4, 5, 7, 8] [0] [2] [6]	[3, 5] [4, 7] [1] [8] [0] [2] [6]	[3, 4, 5, 7] [1] [8] [0] [2] [6]
Heart- statlog	[1, 6] [5, 12] [3, 8] [9, 10] [7, 11] [4] [0] [2]	[1, 6] [5, 12] [6, 8] [3, 5, 11] [5, 9, 10] [7, 9, 11] [9, 12] [5, 6, 11] [6, 9, 11] [11, 12] [3, 9, 11] [5, 10, 11] [5, 7, 11] [4, 12] [4, 6] [3, 4] [5, 6, 9] [3, 5, 9] [4, 10] [1, 3] [3, 8] [5, 7, 9] [1, 10] [5, 9, 11] [4, 11] [4, 5] [8, 9] [0] [2]	[5, 12] [6, 9] [3, 11] [4, 10] [1] [7] [8] [0] [2] [5, 6, 11] [6, 9, 11] [3, 9, 11] [4, 5, 6, 9] [3, 4] [5, 6, 9] [1, 3] [3, 8] [5, 9] [1, 10] [5, 9, 11] [4, 5] [8, 9] [0] [2]	[6, 9, 11] [5, 12] [3, 4] [1, 10] [7] [8] [0] [2]
Ecoli	[0, 2, 3, 4] [1] [5, 6]	[0, 2, 3, 4] [1, 2, 3, 4] [2, 3, 4, 5, 6] [0, 2, 3, 5] [0, 2, 3, 6] [0, 1, 2, 3]	[0, 3] [2, 5] [4, 6] [1]	[2, 3, 4, 5, 6] [0, 1]

Tableau 17 — Méthodes de sélection des groupes d'attributs finaux : comparaison des groupes d'attributs obtenus

Étude des types d'associations

Nous avons élaboré une étude empirique fine des différentes formes d'associations qui ont été détectées par notre méthode SIFRA en comparaison avec celles détectées par SIFCO. RStudio⁴ fournit des représentations graphiques de la distribution des données entre chaque paire d'attributs. En utilisant ces graphiques, nous pouvons observer les formes des dépendances détectées par la méthode de regroupement d'attributs et nous pouvons faire une analyse empirique des types d'associations trouvées. Nous présentons dans la figure 57 les associations découvertes par SIFCO uniquement. La figure 58 contient les associations obtenues par SIFCO et SIFRA. Enfin, la figure 59 représente les associations trouvées par SIFRA seulement. Nous ne présentons pas toutes les associations obtenues par SIFCO et SIFRA, mais nous donnons un exemple de chaque type d'association afin d'illustrer nos propos.

Nous observons que la plupart des associations découvertes par SIFCO sont des corrélations linéaires. Néanmoins, SIFCO a également détecté certaines corrélations non linéaires comme dans les figures 57.b et 57.c, ce qui est inattendu puisque SIFCO est censée chercher uniquement des corrélations linéaires. Il a été mentionné dans Saporta, 2006 que le coefficient de Pearson n'est pas robuste et qu'il peut parfois révéler des corrélations non linéaires et les considérer comme linéaires. La figure 58 montre que certaines corrélations linéaires ont également été détectées par SIFRA. De plus, SIFRA est en mesure de trouver d'autres types d'associations. La figure 59 en représente quelques-unes.

Les graphiques (a), (b), (c), (d), (f) et (g) de la figure 59 représentent une forte concentration des données dans une région de l'espace. Des associations non linéaires et non monotones peuvent être détectées par SIFRA comme le cas du graphique (e). Enfin, une fonction constante entre deux attributs est également considérée comme une association dans SIFRA (voir le graphique (h)).

Pour conclure, les résultats expérimentaux montrent que SIFRA peut détecter différents types de dépendances entre les attributs, y compris les corrélations linéaires et non linéaires. En fait, SIFRA ne cherche pas une forme spécifique de dépendance, mais elle se focalise sur la concentration des données dans certaines régions de l'espace.

4. <https://www.rstudio.com/products/RStudio/>

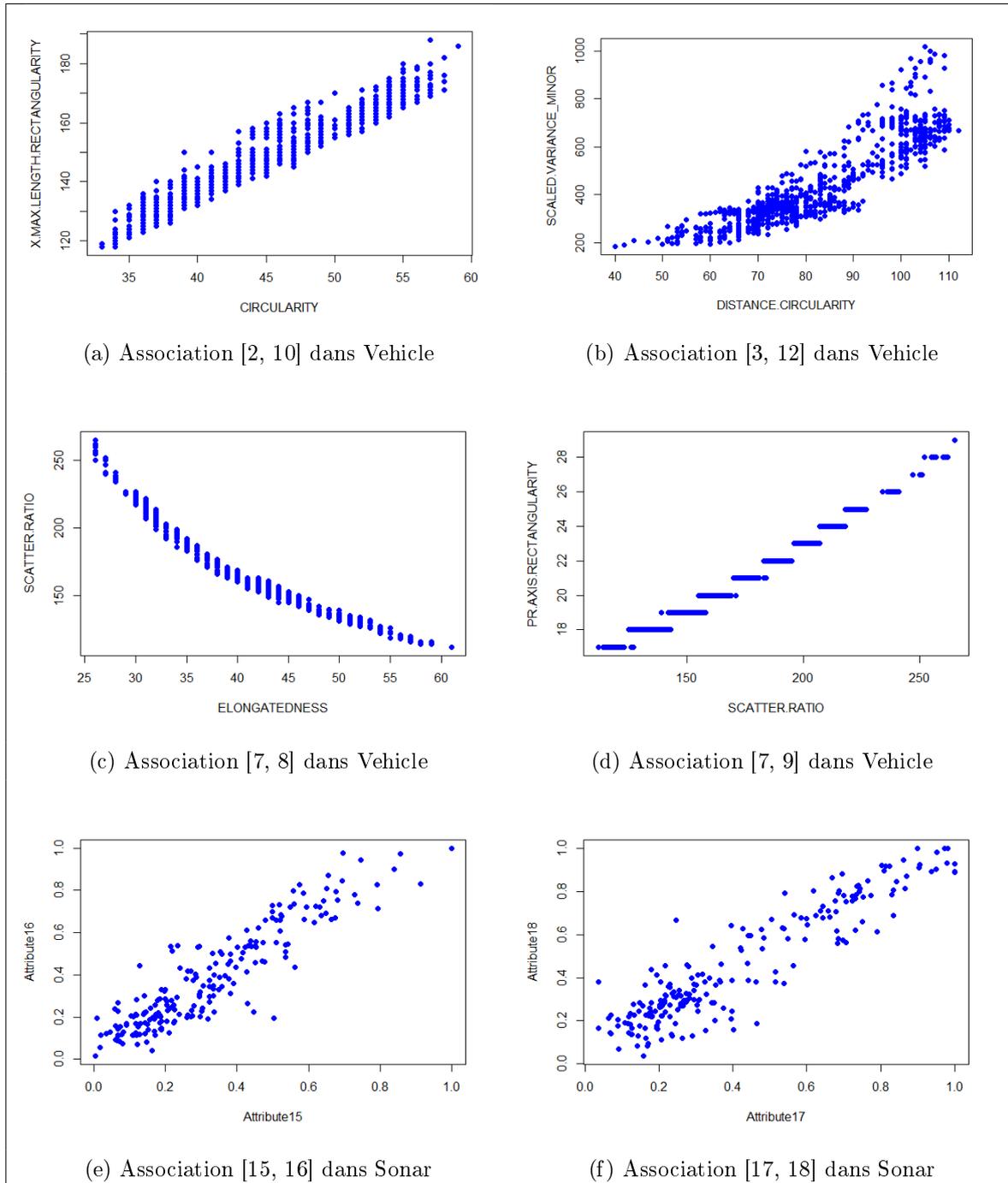


Figure 57 — Associations détectées par SIFCO

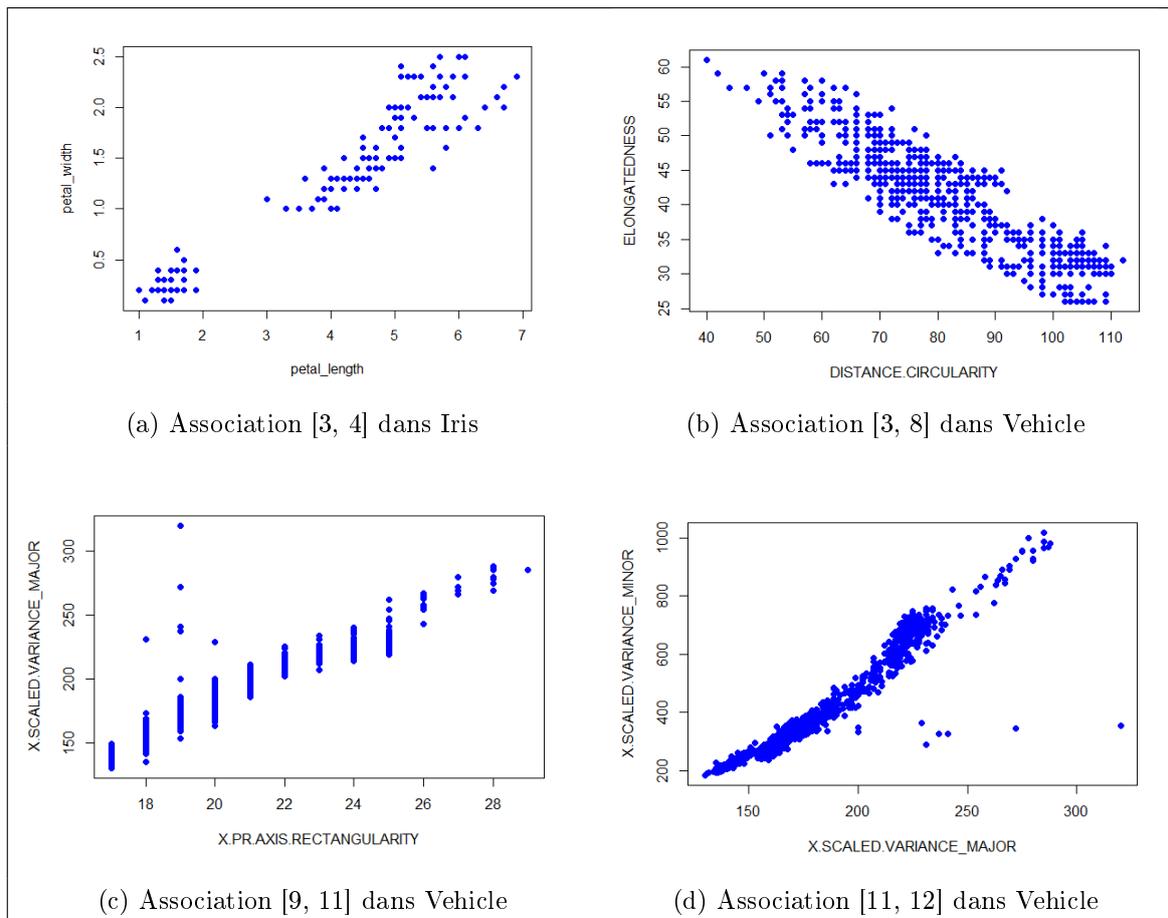


Figure 58 — Associations détectées par SIFCO et SIFRA

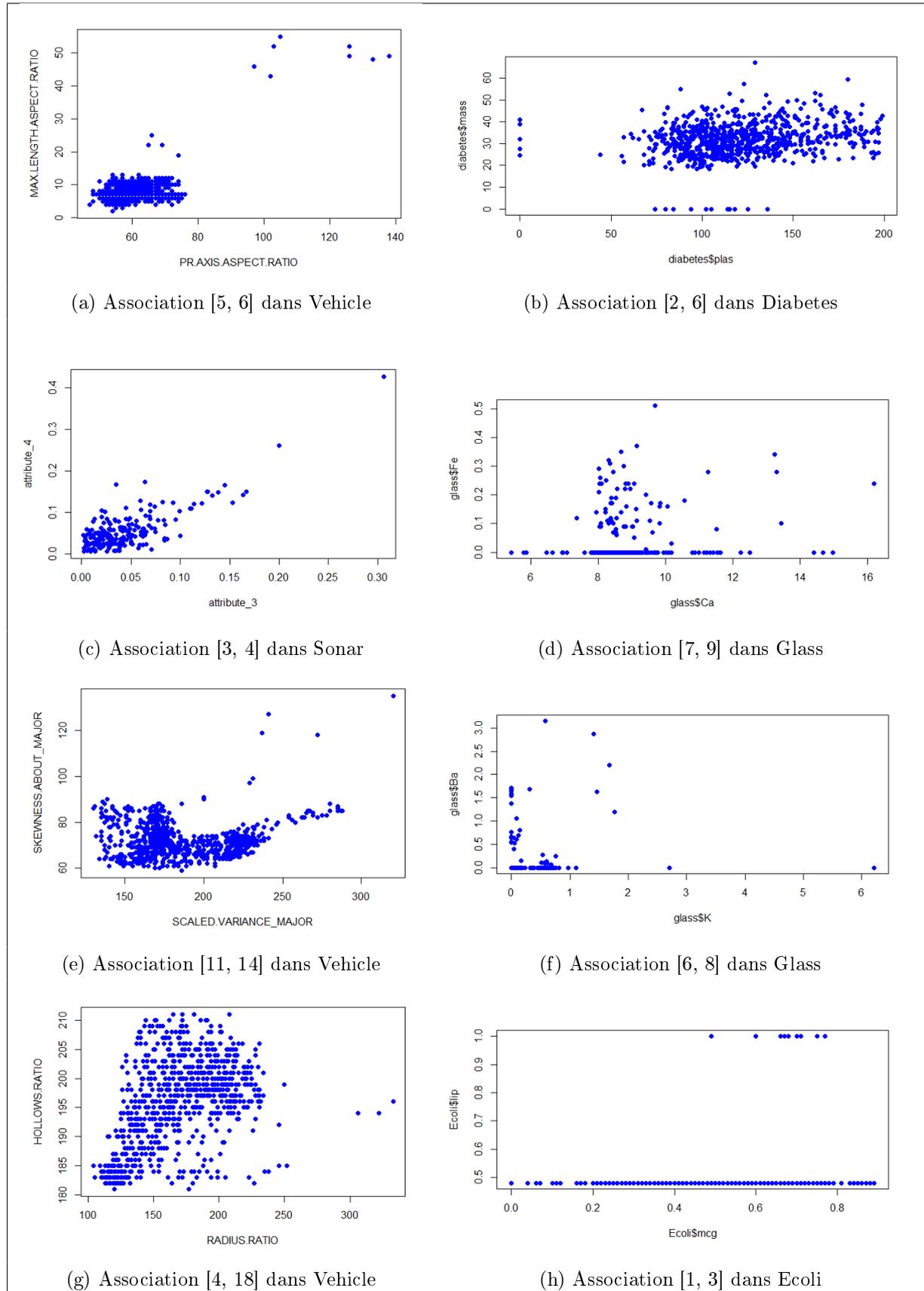


Figure 59 — Associations détectées par SIFRA

Réduction du coût de la méthode SIFRA

Afin de réduire le coût de la méthode SIFRA, nous avons proposé dans le paragraphe 3.6 (chapitre 3) trois solutions principales :

1. l'utilisation de l'algorithme FP-Growth au lieu de Apriori pour la génération des itemsets fréquents
2. l'utilisation de l'algorithme FPclose pour la génération des itemsets fermés au lieu de générer tout l'ensemble des itemsets fréquents.
3. l'utilisation de l'algorithme FPmax pour la génération des itemsets maximaux qui génère un ensemble plus réduit d'itemsets en gardant seulement les itemsets les plus longs

Nous avons implémenté ces différentes méthodes dans le cadre de notre système SIFRA. Nous rappelons que, le code de la phase de sélection des groupes d'attributs finaux (phase 3 de la méthode de regroupement d'attributs de SIFRA) est modifiée lorsque l'algorithme FPmax est utilisé (voir paragraphe 3.6).

Nous rappelons aussi que, dans le cas de l'utilisation de l'algorithme FPclose, des régions liées peuvent être ratées (voir paragraphe 3.6), ce qui pourrait diminuer la performance de la méthode SIFRA. Cependant, il est possible de retrouver toutes les régions ratées en utilisant les supports des sur-ensembles des itemsets ratés, puisque FPclose élimine les itemsets qui possèdent des sur-ensembles ayant le même support. Pour cela, nous avons proposé d'adapter le code de la phase de détermination des attributs liés (phase 2 de la méthode de regroupement d'attributs de SIFRA) pour retrouver tous les itemsets fréquents ratés lors de la détermination des attributs liées. Ce code peut toutefois mettre un temps de réponse supplémentaire. Pour cette raison, nous proposons de tester les deux versions de la phase de détermination des attributs liés :

1. la première utilise l'algorithme FPclose pour la génération des itemsets fermés et déduit les groupes d'attributs liés sans retrouver les régions ratées.

2. la deuxième utilise l'algorithme FPclose pour la génération des itemsets fermés et retrouve les régions ratées pour déterminer les groupes d'attributs liés sans perdre aucune information.

Cinq versions de SIFRA sont alors testées et comparées dans cette section :

1. SIFRA avec Apriori : SIFRA avec l'utilisation de l'algorithme Apriori pour la génération des itemsets fréquents,
2. SIFRA avec FP-Growth : SIFRA avec l'utilisation de l'algorithme FP-Growth pour la génération des itemsets fréquents,
3. SIFRA avec FPclose : SIFRA avec l'utilisation de l'algorithme FPclose pour la génération des itemsets fermés,
4. SIFRA avec FPclose adaptée : SIFRA avec l'utilisation de l'algorithme FPclose pour la génération des itemsets fermés et la modification du code pour retrouver les régions liées ratées par FPclose,
5. SIFRA avec FPmax : SIFRA avec l'utilisation de l'algorithme FPmax pour la génération des itemsets maximaux.

Nous avons utilisé les valeurs suivantes des paramètres d'entrée pour pouvoir mener la comparaison sous les mêmes conditions :

- la partition floue utilisée dans la phase d'apprentissage est la partition floue supervisée [Soua et al., 2013] avec la méthode de discrétisation MDLP [Fayyad and Irani, 1993]
- la discrétisation utilisée dans la phase de regroupement d'attributs est la discrétisation régulière (EWI) ; le nombre d'intervalles est fixé à 3.
- le seuil d'association β_{min} est fixé à 0.7
- le seuil *minsupp* est fixé à 0.3 pour la plupart des bases de données. Cependant, un nombre très important d'itemsets fréquents est généré lorsqu'on utilise la valeur 0.3 du seuil *minsupp* pour les bases de données Sonar et Ionosphere où un grand nombre d'attributs existe. Pour cette raison, des valeurs plus grandes de *minsupp* sont utilisées pour ces bases de données (0.5 et 0.6). D'un autre côté, pour certaines autres bases (comme Ecoli et Balance-scale), la valeur 0.3 du *minsupp* ne permet de générer aucun itemset fréquent. Pour cela, une valeur plus faible de *minsupp* (0.05) est choisie pour ces bases. En conclusion, la même valeur du seuil *minsupp* est utilisée pour une même base de données dans les cinq tests expérimentaux.

Dans cette étude, les tests expérimentaux sont menés sur 12 bases de données. Cinq de ces bases possèdent un nombre d'attributs inférieur à 10 (Iris, Balance-scale, Ecoli,

Diabetes, et Glass); trois possèdent un nombre d'attributs moyennement important (entre 10 et 20 attributs dans les bases Wine, Heart-statlog et Vehicle). Enfin, quatre bases de données contiennent un nombre important d'attributs (entre 30 et 80 attributs pour les bases Unbalanced, Ionosphere, Sonar et mfeat-fourrier). Un nombre d'exemples qui varient entre 150 et 2000 instances caractérisent ces bases.

Nous présentons les temps de réponse et les taux de bonne classification des différents tests expérimentaux dans des graphiques de type histogramme (Figures [60](#), [61](#), [62](#), [63](#)). Nous avons choisi d'utiliser ces deux critères (temps de réponse et taux de bonne classification) dans la comparaison pour pouvoir vérifier si la réduction du temps de réponse n'engendre pas une perte de la performance de la méthode de classification.

Nous commençons l'analyse par la figure [60](#) qui regroupe des bases de petit nombre ou de nombre moyen d'attributs. Le premier point à remarquer est que l'algorithme Apriori (représenté par la première série de données : la barre en bleu) donne le temps de réponse le plus important dans la plupart des bases de données, sauf quelques exceptions (dans les bases Balance-scale et Wine). L'utilisation de FP-Growth dans ces bases (la barre en rouge) donne toujours un temps de réponse plus réduit que Apriori en gardant le même taux de bonne classification. Concernant l'algorithme FPclose (la barre en vert), il donne généralement un temps de réponse plus réduit par rapport à tous les autres algorithmes. Cependant, en analysant l'histogramme à droite, nous remarquons une baisse de la performance dans certaines bases de données comme Ecoli, Glass et Wine. Cette baisse de performance peut être expliquée par la perte de quelques associations intéressantes causée par l'absence de quelques itemsets fréquents qui peuvent être intéressants pour la classification. Ce résultat confirme que le choix des groupes d'attributs a une influence importante sur les résultats de classification et que la perte des petites associations locales entre les intervalles peut causer une baisse de la performance du classifieur. L'adaptation du code pour pouvoir retrouver toutes les régions liées ratées par FPclose est représentée par la barre violet. Cette adaptation permet de garder une bonne performance de classification dans toutes les bases (ce qui montre le graphique à droite), mais en contre partie, elle donne un temps de réponse plus important que la version simple de FPclose, plus important que FP-Growth et parfois plus important aussi que Apriori! Regardons enfin la version de SIFRA avec l'algorithme FPmax (la barre en bleu ciel), nous remarquons d'après la figure [60](#) que le temps de réponse est réduit généralement par rapport à celui obtenu par Apriori, mais il est parfois plus important que le temps obtenu par FP-Growth. D'autre part, un temps de réponse très important est obtenu avec FPmax dans le cas de la base Balance-scale. Concernant le taux de bonne classification, il n'existe pas un comportement général de FPmax, le taux augmente dans certaines bases de données et diminue dans d'autres (une diminution considérable est détectée dans les bases Ecoli et Balance-scale). Pour conclure, il s'avère que le résultat obtenu par FPmax est très dépendant des données que ce soit pour le temps de réponse ou pour le taux de bonne classification.

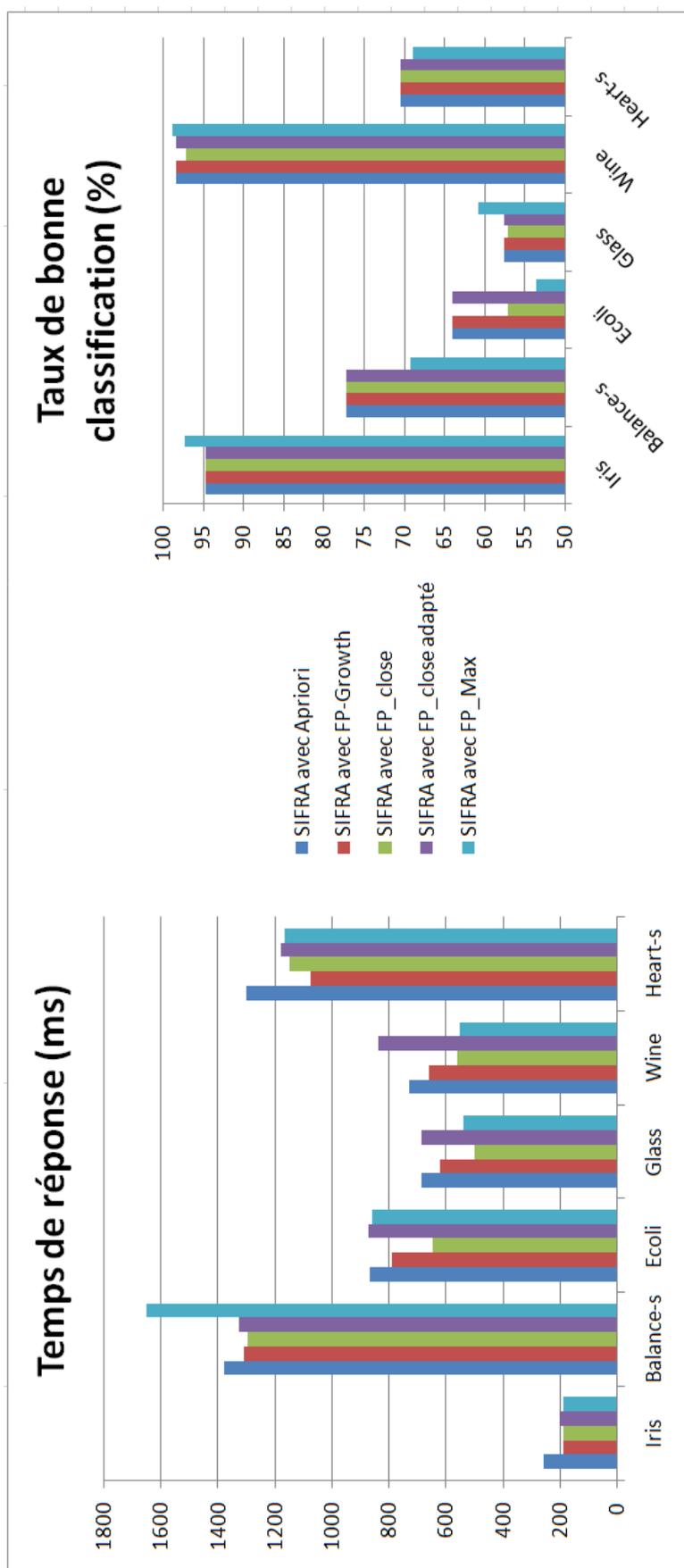


Figure 60 — Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Iris, Balance-scale, Ecoli, Glass, Wine et Heart-statlog

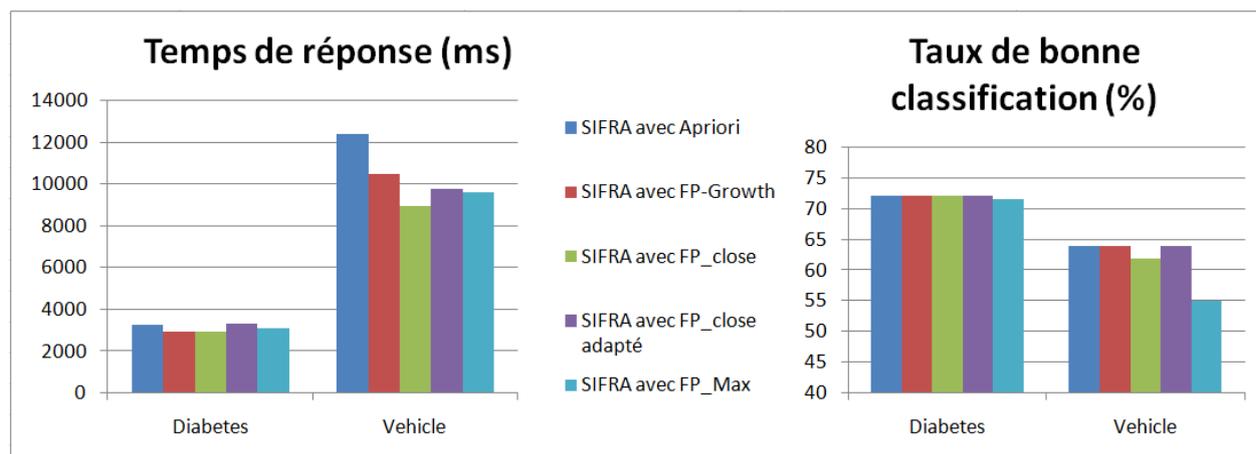


Figure 61 — Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Diabetes et Vehicle

Une analyse fine des groupes d'attributs obtenus par FPmax et les différents autres algorithmes est fournie dans le tableau [18](#). Dans ce tableau, nous ne mettons que les groupes d'attributs détectés. Pour les attributs qui n'apparaissent pas dans le tableau, chacun est considéré comme indépendant et est traité seul par un classifieur. Le tableau [18](#) montre que l'utilisation de l'algorithme FPmax ne permet d'aboutir à aucun groupe d'attributs dans presque toutes les bases : chaque attribut est traité indépendamment et une base de règles locale est générée pour chaque attribut pris seul. En effet, les itemsets de longueur maximale possèdent généralement des supports faibles ; de même, les groupes d'attributs les plus longs possèdent généralement des degrés d'association faibles qui ne peuvent pas excéder le seuil β_{min} . Dans ce cas (lorsque aucun groupe n'est détecté), chaque attribut sera considéré comme indépendant et traité seul par un classifieur.

Concernant la base Diabetes, l'analyse de la figure [61](#) montre des temps de réponse proches avec les différentes versions de SIFRA ; une petite amélioration est obtenue par les algorithmes FP-Growth, FPclose et FPmax par rapport à Apriori. Cependant la version adaptée de FPclose (la barre en violet) a nécessité cette fois aussi un temps de réponse supplémentaire. Pour la base Vehicle, FPclose (la barre verte) a mis le temps de réponse le plus réduit, mais en contre partie, le taux de bonne classification diminue. FPmax réduit aussi le temps de réponse par rapport à Apriori et FP-Growth, mais une réduction considérable du taux de bonne classification est notée.

Pour les bases de données qui possèdent les nombres d'attributs le plus importants (Figures [62](#) et [63](#)), nous remarquons que FPclose donne le meilleur temps de réponse dans la plupart des bases en gardant un taux de bonne classification très proche ou similaire à celui de Apriori. Ce résultat confirme que la génération des itemsets fermés au lieu de tout l'ensemble des itemsets fréquents dans les bases de grande dimension pourrait diminuer considérablement le temps de réponse. Un point important à noter dans ces bases est que l'absence de quelques régions liées (causée par l'élimination des itemsets fréquents qui possèdent des sur-ensembles de même support) n'a pas trop influencé la performance de classification. Ceci peut être expliqué par le fait que plusieurs associations peuvent

être détectées lorsque le nombre d'attributs est important et que l'absence de quelques associations n'affecte pas trop le résultat de classification, par contre, dans les bases ayant un petit nombre d'attributs, un petit nombre d'associations est découvert par la méthode de regroupement d'attributs et l'élimination de quelques régions liées peut supprimer toutes les associations trouvées dans ces bases. Afin de mieux comprendre ce résultat, une analyse des groupes d'attributs liés détectés par les différentes méthodes est fournie dans le tableau [18](#). Nous nous focalisons sur les bases de données dont une baisse de la performance a été notée dans le cas de FPclose. Ces bases (qui sont Ecoli, Wine, Glass et Vehicle) possèdent un nombre moyen d'attributs (entre 7 et 18). Nous observons que FPclose a gardé un seul groupe par rapport à Apriori dans les bases Ecoli et Vehicle. Dans la base Wine, deux groupes d'attributs ont été ratés par FPclose et deux autres les ont remplacés. Enfin, pour la base Glass, la seule association détectée par Apriori [4, 5, 6, 8] a abouti à un taux de classification très satisfaisant. Ce groupe a été raté par FPclose et a été remplacé par le groupe [5, 6, 8, 9]. Voyons maintenant les bases qui possèdent un nombre d'attributs plus important et dont le même taux de classification a été obtenu par Apriori et FPclose : Ionosphere, Unbalanced, Sonar et mfeat-fourrier. Pour les bases Ionosphere et mfeat-fourrier (qui possèdent respectivement 35 et 76 attributs), les mêmes groupes d'attributs ont été obtenus par FPclose et Apriori. Concernant la base Sonar, trois parmi cinq groupes d'attributs détectés par Apriori n'ont pas été découverts par FPclose, cependant, FPclose a trouvé d'autres associations. Ces nouvelles associations n'ont pas dégradé la performance de classification. La même remarque est valable dans le cas de la base Unbalanced : la perte de quelques groupes d'attributs par FPclose n'a pas influencé les résultats de classification.

Concernant la version adaptée de FPclose (la barre violet), un temps de réponse énorme est obtenu par cette méthode (surtout dans le cas des bases Unbalanced, Ionosphere et Sonar). Ceci confirme que la recherche de tous les itemsets fréquents qui ont été ratés par FPclose nécessite un coût supplémentaire qui peut être très important dans certaines bases. Donc, cette méthode n'est pas intéressante dans ce cas.

D'après la figure [62](#), il est aussi à noter que l'algorithme FP-Growth a mis un temps de réponse plus réduit que FPclose dans la base Unbalanced, bien que FPclose est plus rapide dans la génération des itemsets ! On peut donc supposer que le temps de réponse supplémentaire trouvé dans le cas de FPclose a été passé dans la phase d'apprentissage ou de génération des règles.

Finalement, un temps de réponse très réduit est obtenu lorsqu'on utilise l'algorithme FPmax dans les bases de données Unbalanced, Sonar et Ionosphere (Figures [62](#) et [63](#)). Le taux de classification dans ces bases reste aussi similaire à celui obtenu par Apriori. Cependant, le tableau [18](#) montre que l'utilisation de FPmax fait perdre toutes les associations entre les attributs dans la plupart des bases de données : aucun groupe d'attributs n'est détecté, chaque attribut sera donc traité seul par un classifieur et une base locale sera générée pour chaque attribut pris seul.

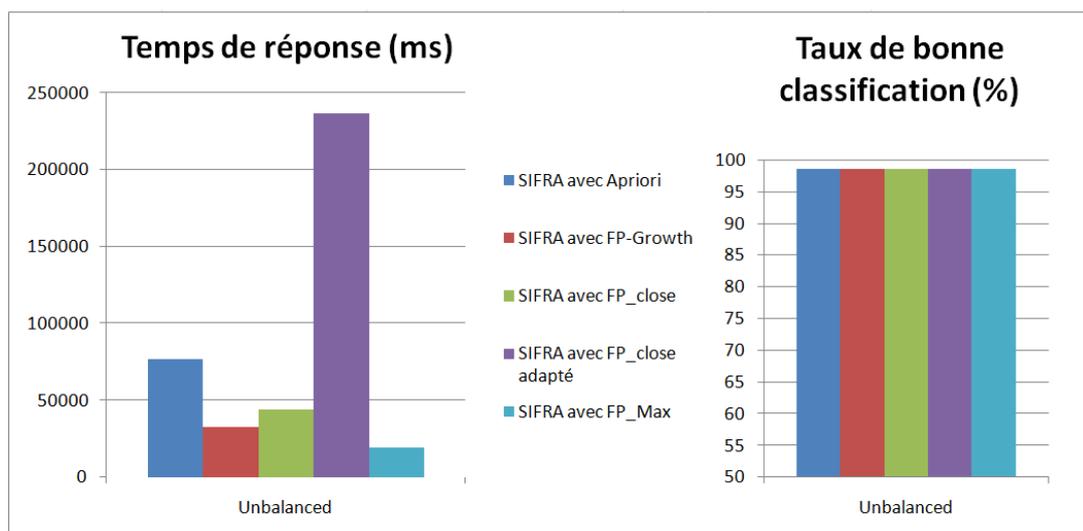


Figure 62 — Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour la base Unbalanced

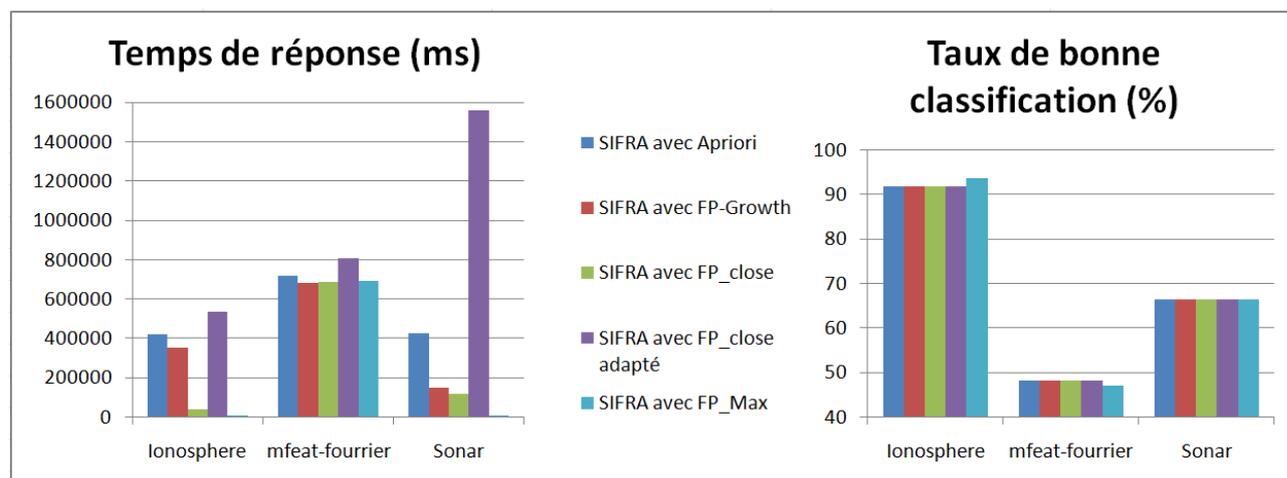


Figure 63 — Temps de réponse et taux de bonne classification des cinq versions de SIFRA pour les bases Ionosphere, mfeat-fourrier et Sonar

Pour conclure, l'utilisation de FPclose dans SIFRA avec l'adaptation du code pour retrouver toutes les régions liées ratées (la barre violet) ne semble pas intéressante surtout dans le cas des bases de données ayant un grand nombre d'attributs. En effet, le traitement qui cherche les régions ratées nécessite un temps de réponse parfois beaucoup plus important que le temps mis par Apriori. D'un autre côté, l'utilisation de l'algorithme FPmax présente des résultats intéressants dans certaines bases de données (surtout les grandes bases) mais aussi dégrade considérablement les résultats dans d'autres bases de données (comme Vehicle, Balance-scale, Ecoli et Heart-statlog). Par ailleurs, l'analyse des groupes d'attributs générés par SIFRA en utilisant FPmax montre qu'aucun groupe d'attributs n'a été détecté dans la plupart des bases de données. De ce fait, l'utilisation de FPmax dans la phase de regroupement d'attributs ne semble pas être adaptée à notre méthode SIFRA. Une extension possible qui peut être proposée dans le cas de FPmax consiste à

éliminer l'utilisation du seuil β_{min} afin d'éviter la perte de tous les groupes d'attributs. En d'autres termes, tous les groupes d'attributs qui possèdent au moins une région liée seront considérés comme liés. Le filtrage se fait par la suite en choisissant le groupe ayant le plus grand degré d'association et en éliminant les chevauchements entre les groupes (étapes de la figure [34](#)). Le choix des groupes les plus longs est implicite dans cette méthode (à travers l'utilisation de FPmax). Cette idée pourrait être envisagée comme une perspective de ce travail. Les deux choix intéressants dans notre méthode SIFRA consiste alors en l'utilisation des algorithmes FP-Growth ou FPclose (sans adaptation). FPclose possède l'avantage de réduire considérablement le temps de réponse. Cependant, une baisse de la performance de classification pourrait être obtenue à cause de la perte de quelques régions liées. Cet inconvénient a été bien marqué dans les petites bases, mais il n'a pas trop influencé les grandes bases. D'autre part, l'algorithme FP-Growth garantit l'obtention de tous les groupes d'attributs envisagés et donc n'affecte pas la performance de classification. De plus, l'algorithme FP-Growth garantit la réduction du temps de réponse par rapport à Apriori et a montré aussi une réduction considérable du temps de réponse par rapport à FPclose dans certaines bases de données comme Unbalanced, Sonar et Heart-statlog.

	SIFRA avec Apriori	SIFRA avec FP-Growth	SIFRA avec FPclose	SIFRA avec FPclose Adapté	SIFRA avec FPmax
Iris	[3, 4] [1, 2]	[3, 4] [1, 2]	[3, 4] [1, 2]	[3, 4] [1, 2]	Aucun groupe
Balance-scale	[1, 4] [2, 3]	[1, 4] [2, 3]	[1, 4] [2, 3]	[1, 4] [2, 3]	Aucun groupe
Ecoli	[1, 2, 3, 4, 6] [5, 7]	[1, 2, 3, 4, 6] [5, 7]	[1, 2, 3, 4, 6]	[1, 2, 3, 4, 6] [5, 7]	Aucun groupe
Glass	[4, 5, 6, 8]	[4, 5, 6, 8]	[5, 6, 8, 9]	[4, 5, 6, 8]	Aucun groupe
Wine	[7, 12] [3, 5] [4, 11]	[7, 12] [3, 5] [4, 11]	[5, 7] [4,11] [3, 10]	[7, 12] [3, 5] [4, 11]	Aucun groupe
Heart-statelog	[6, 13] [7, 10, 12]	[6, 13] [7, 10, 12]	[6, 13] [7, 10, 12]	[6, 13] [7, 10, 12]	Aucun groupe
Diabetes	[4, 5, 6, 7]	[4, 5, 6, 7]	[4, 5, 6, 7]	[4, 5, 6, 7]	Aucun groupe
Vehicle	[4, 5, 6, 14] [8, 11, 12]	[4, 5, 6, 14] [8, 11, 12]	[4, 5, 6, 14]	[4, 5, 6, 14] [8, 11, 12]	Aucun groupe
	[10, 13]	[10, 13]		[10, 13]	Aucun groupe
Unbalanced	[9, 22, 24, 29, 32]	[9, 22, 24, 29, 32]	[9, 22, 24, 29, 32]	[9, 22, 24, 29, 32]	Aucun groupe
	[18, 26] [13, 15]	[18, 26] [13, 15]	[13, 15] [17, 19]	[18, 26] [13, 15]	
	[11, 20] [4, 6]	[11, 20] [4, 6]	[16, 21, 23]	[11, 20] [4, 6]	
	[17, 19] [16, 21, 23]	[17, 19] [16, 21, 23]		[17, 19] [16, 21, 23]	
Sonar	[2, 3, 4, 5, 51, 52, 60]	[2, 3, 4, 5, 51, 52, 60]	[2, 3, 4, 5, 51, 52, 60]	[2, 3, 4, 5, 51, 52, 60]	Aucun groupe
	[56, 58, 59] [1, 57]	[56, 58, 59] [1, 57]	[56, 58, 59]	[56, 58, 59] [1, 57]	
	[46,55][47,50]	[46,55][47,50]	[55,57] [46, 47]	[46,55][47,50]	
Ionosphere	[1, 2]	[1, 2]	[1, 2]	[1, 2]	Aucun groupe
mfeat-fourrier	[24, 35] [31, 34]	[24, 35] [31, 34]	[24, 35] [31, 34]	[24, 35] [31, 34]	[24, 45]
	[16, 19] [10, 36]	[16, 19] [10, 36]	[16, 19] [10, 36]	[16, 19] [10, 36]	[4, 34] [10, 30]

Tableau 18 — Groupes d'attributs obtenus par les différentes versions de SIFRA

G

Effet des seuils $minsupp$ et β_{min} sur le coût de la méthode SIFRA

L'application de notre méthode de regroupement d'attributs par Règles d'Association nécessite le choix des valeurs de certains paramètres d'entrée tels que le seuil de support des itemsets $minsupp$ et le seuil d'association β_{min} .

Il est à noter que les valeurs de ces deux seuils peuvent influencer considérablement le temps de réponse ainsi que le taux de classification de notre système d'apprentissage. En effet, la diminution des seuils $minsupp$ et β_{min} entraîne l'augmentation du nombre des itemsets générés ce qui peut affecter significativement le coût de la phase de génération des itemsets fréquents et celui de la phase de détermination des groupes d'attributs liés (voir section 3.6). Pour cela, nous présentons dans cette section une analyse fine de l'effet des seuils $minsupp$ et β_{min} sur le temps de réponse de la méthode SIFRA. Nous utilisons dans cette analyse l'algorithme FP-Growth pour la génération des itemsets fréquents. L'analyse est réalisée sur les deux bases de données Sonar et Ionosphere puisque ces deux bases possèdent un nombre d'attributs moyennement important et donc nécessitent un temps de réponse relativement important pour la génération des modèles d'apprentissage.

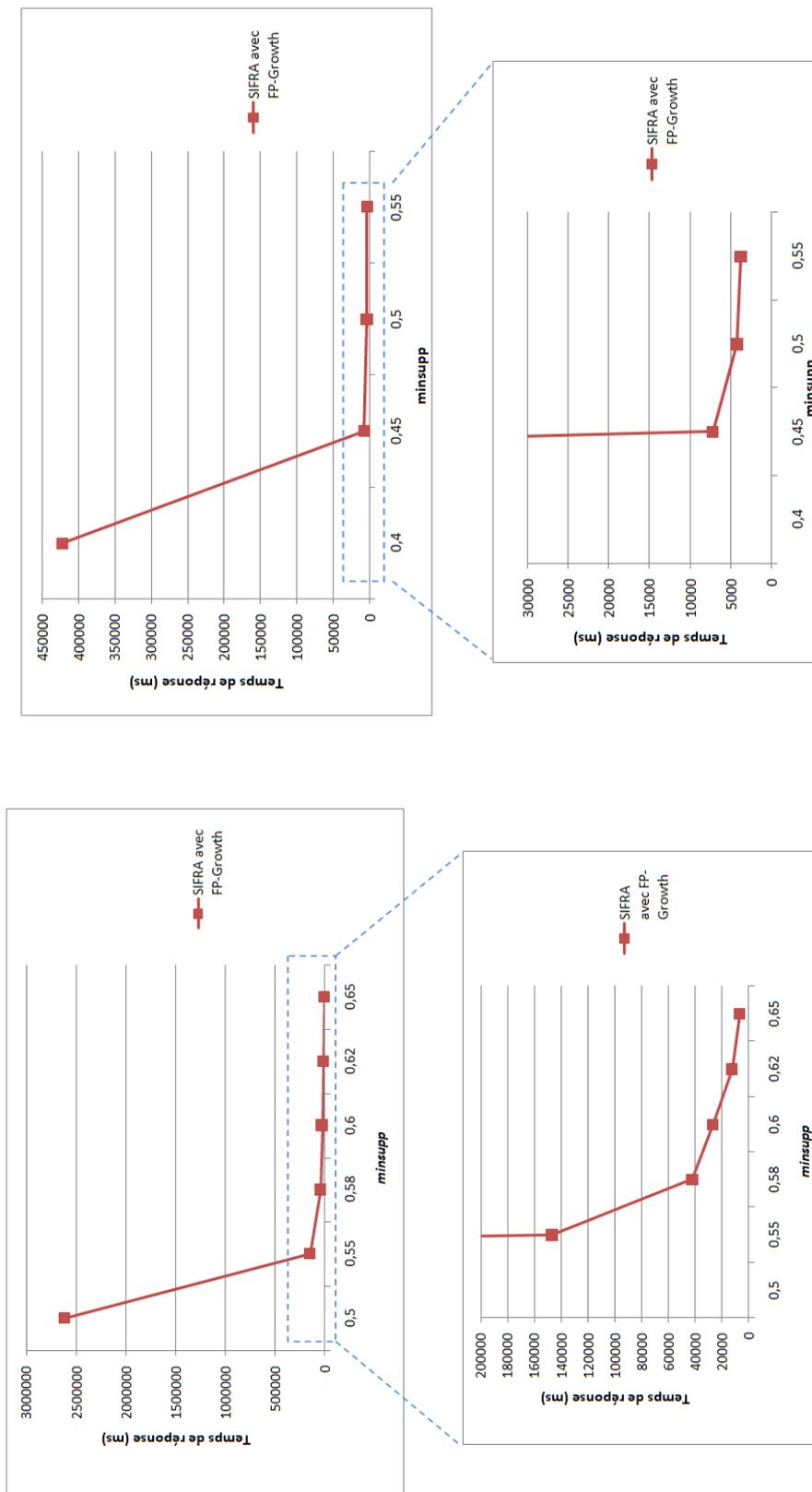
La figure 64 illustre l'effet du seuil $minsupp$ sur le temps de réponse de la méthode SIFRA pour les bases de données Sonar et Ionosphere. Le temps de réponse est calculé en faisant varier la valeur du seuil $minsupp$ et en gardant une valeur fixe du seuil β_{min} ($\beta_{min} = 0.7$ pour les deux bases de données). D'un autre côté, la figure 65 représente l'effet du seuil β_{min} sur le temps de réponse de SIFRA également pour les mêmes bases de données. Dans cette figure, le temps de réponse est calculé en faisant varier le seuil β_{min} et en utilisant une valeur fixe de $minsupp$ pour chaque base ($minsupp = 0.55$ pour la base de données Sonar et $minsupp = 0.45$ pour la base de données Ionosphere).

D'après la figure 64, nous remarquons que les deux courbes (les courbes sans le zoom) possèdent presque la même allure. Dans la base Sonar, la méthode SIFRA a mis un temps de réponse important (plus de 2500 secondes) pour la valeur 0.5 du seuil $minsupp$. Cependant, à partir de la valeur 0.55 ou plus, le temps de réponse diminue d'une manière considérable et devient presque stabilisé. La même remarque est notée dans le cas de la

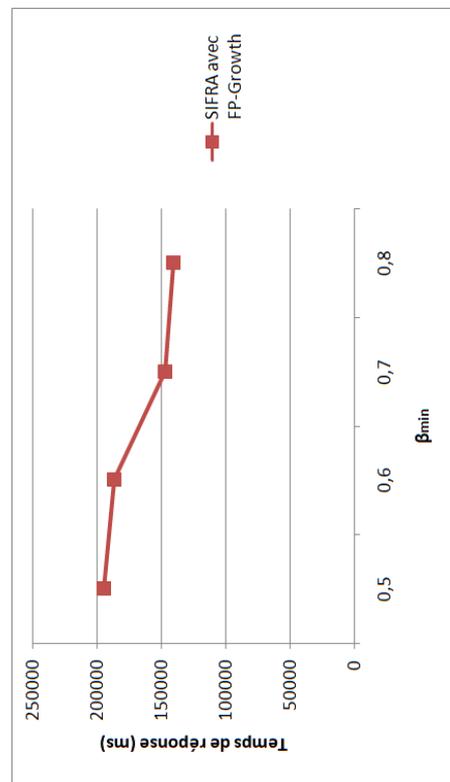
base de données Ionosphere : le temps de réponse diminue avec l'augmentation de la valeur de $minsupp$ et devient presque inchangé à partir de la valeur 0.45 du seuil $minsupp$. Il est à noter que les courbes en haut de la figure 64 possèdent une échelle grossière ; c'est pour cette raison que les temps de réponse paraissent très proches à partir d'une certaine valeur de $minsupp$. Afin de pouvoir analyser de près ces valeurs proches, nous réalisons un zoom sur l'axe de temps de réponse pour les deux bases de données. À travers ce zoom, nous notons que le temps de réponse diminue toujours avec l'augmentation du seuil $minsupp$, mais à partir d'une valeur spécifique de $minsupp$, la diminution devient légère.

Nous nous intéressons maintenant à l'effet du seuil β_{min} sur le temps de réponse de SIFRA (Figure 65). À travers cette figure, nous notons que le changement du seuil β_{min} n'a pas un grand impact sur le temps de réponse de SIFRA. Le temps de réponse diminue évidemment avec l'augmentation du seuil β_{min} , mais la diminution est très légère. Une diminution claire du temps de réponse a été obtenue entre les valeurs 0.6 et 0.7 dans le cas de la base Sonar et entre les valeurs 0.7 et 0.8 pour la base Ionosphere, mais la diminution n'est pas considérable et le temps de réponse reste du même ordre de grandeur. En effet, le seuil d'association β_{min} n'affecte que la troisième phase du regroupement d'attributs de SIFRA (la phase de sélection des groupes d'attributs finaux) : une faible valeur de β_{min} engendre l'obtention de plus de groupes d'attributs liés. Cependant, nous avons vu, d'après l'analyse du coût de la méthode SIFRA (dans la section 3.6) que les deux premières phases de notre méthode de regroupement d'attributs sont plus coûteuses que la troisième phase. Cela explique le fait que l'augmentation du coût de cette troisième phase n'influence pas beaucoup le coût global de la méthode SIFRA.

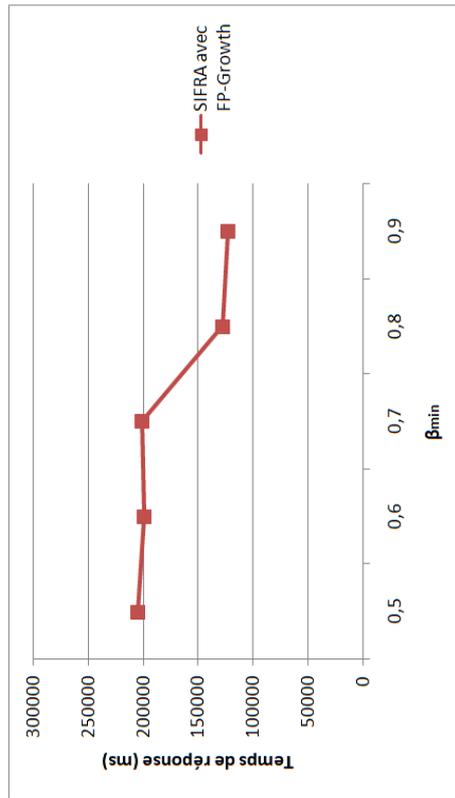
Pour conclure, nous notons que le paramètre $minsupp$ est le principal paramètre influent sur le coût de SIFRA. Varier la valeur de $minsupp$ peut modifier totalement l'ordre de grandeur du temps de réponse de SIFRA. Donc, pour traiter le problème de la complexité de SIFRA dans les grandes bases de données, la solution la plus facile à appliquer consiste à augmenter la valeur du seuil $minsupp$. Toutefois, il est à noter que l'augmentation de la valeur de $minsupp$ engendre aussi la diminution du nombre d'associations détectées et des groupes d'attributs générés. Une étude fine de l'effet du seuil $minsupp$ sur les groupes d'attributs générés par SIFRA est présentée dans le chapitre 5, section 5.5. Cette étude montre que le choix de la valeur appropriée de $minsupp$ dépend fortement des données traitées.



(a) Base de données Sonar
Figure 64 — Effet du seuil $minsupp$ sur le temps de réponse de SIFRA (avec un Zoom sur l'axe de temps)
 (b) Base de données Ionosphere



(a) Base de données Sonar



(b) Base de données Ionosphere

Figure 65 — Effet du seuil β_{min} sur le temps de réponse de SIFRA

Effet des paramètres d'entrée sur la méthode SIFRA

Nous nous intéressons dans cet annexe à l'analyse de l'effet de certains paramètres d'entrée sur les résultats de la méthode SIFRA, notamment la partition floue utilisée dans la phase de génération des règles et les seuils $minsupp$ et β_{min} requis pour la phase de regroupement d'attributs.

Effet de la méthode de partition floue

Nous commençons par analyser l'effet de la méthode de partition floue sur les résultats de notre méthode SIFRA ; les résultats sont traités en terme de taux de bonne classification et du nombre de règles générées.

Le tableau [19](#) présente une comparaison entre les deux méthodes de partition. Pour la partition simple, plusieurs valeurs de k (la taille de la partition) ont été testées. Afin de pouvoir comparer les deux méthodes de partition sous les mêmes conditions, les mêmes valeurs des seuils $minsupp$ et β_{min} sont fixées pour chaque base de données. Pour chaque méthode de partition, le taux de bonne classification ($Taux$) et le nombre de règles ($Nb. de règles$) sont indiqués dans le tableau [19](#). Les meilleures valeurs dans chaque base sont marquées en gras.

Nous analysons d'abord le taux de bonne classification. Le tableau [19](#) montre que les meilleurs taux sont obtenus par la partition floue supervisée dans la majorité des bases de données. Une amélioration considérable de presque 20% est obtenue dans le cas de la base Vehicle, de 14.6% pour la base Glass et plus de 10% dans la base Heartstatlog. Des améliorations au alentour de 5% sont aussi obtenues pour les bases Balance-scale, Diabetes et Wine. Enfin, une petite amélioration de moins de 1% est réalisée pour les bases Iris et Ionosphere. La partition floue simple a conduit aux meilleurs taux de classification uniquement avec les bases Ecoli, Sonar et Mfeat-fourrier. Dans ces trois bases, les meilleurs taux sont obtenus pour des valeurs différentes de la taille de la partition k . D'après le tableau [19](#), la meilleure valeur de k change d'une base de données à une

autre. La détermination de la taille de la partition simple appropriée pour chaque base de données n'est pas évidente. Ce problème ne persiste pas dans le cas de la partition supervisée puisque la taille de la partition de chaque attribut est déterminée d'une manière automatique.

Nous passons à l'analyse du nombre de règles générées. Si nous considérons le cas de la partition floue simple, nous remarquons que le nombre de règles augmente avec l'augmentation de la valeur de k . Rappelons que le nombre de règles générées par la méthode de la grille floue simple est égal à k^N (où N est le nombre d'attributs et k est la taille de la partition floue). Il est évident que l'augmentation de la valeur de k engendre nécessairement l'augmentation du nombre de règles floues. Donc, afin de diminuer le nombre de règles, il faut évidemment choisir une valeur faible de k .

En utilisant la partition floue supervisée, des valeurs différentes de k sont obtenues pour chaque attribut. Le nombre de règles générées dans ce cas est $k_1 \times k_2 \times \dots \times k_N$ (où k_i est la taille de partition de l'attribut X_i). Dans ce cas, il n'est pas possible de contrôler les valeurs de k_i puisque la discrétisation supervisée détermine automatiquement la taille de partition pour chaque attribut. Néanmoins, nous obtenons dans le tableau [19](#) un nombre de règles acceptable par rapport à la partition floue simple dans la majorité des bases de données (Iris, Wine, Diabetes, Balance-scale, Ionosphere et Mfeat-fourrier). Le nombre de règles obtenu dans ces bases est proche de celui généré par la partition floue simple où la valeur de k est égale à 3 ou 4. Une valeur de k égale ou supérieure à 5 engendre un nombre de règles important dans la plupart des cas. Pour les bases de données Glass, Heart-statlog et Ecoli, la partition floue supervisée a conduit à un nombre de règles réduit (plus intéressant que la partition floue simple). De plus, un nombre de règles très intéressant a été obtenu dans le cas des bases Sonar et Unbalanced. Dans le cas de la base Sonar, 76 règles floues ont été obtenues par la partition floue supervisée alors que 206 règles ont été générées par la partition floue simple de taille 3. Concernant la base Unbalanced, 10 règles floues ont été construites (partition floue supervisée) contre 118 (partition floue simple de taille 3). Le seul cas où la méthode de partition supervisée a conduit à un nombre de règles important est la base Vehicle. Cependant, dans cette base, le taux de classification obtenu par la partition floue supervisée est beaucoup plus intéressant que ceux obtenus par la partition simple.

Pour conclure, la partition floue supervisée montre des résultats très intéressants dans la plupart des bases de données, que ce soit en termes du nombre de règles ou du taux de bonne classification. Cette méthode de partition floue permet d'avoir un compromis entre ces deux facteurs. Pour cette raison, nous utilisons la partition floue supervisée dans le reste des expérimentations présentées dans ce rapport.

	Partition simple												Partition supervisée	
	k=3			k=4			k=5			k=6			MDLP	
	Taux	Nb. de règles		Taux	Nb. de règles		Taux	Nb. de règles		Taux	Nb. de règles		Taux	Nb. de règles
Iris	94.66	12		83.33	16		94.66	20		96.66	24		97.33	14
Wine	90.44	39		93.25	50		93.82	63		92.13	75		98.88	39
Vehicle	51.41	76		53.07	150		56.14	210		56.38	274		67.61	273
Sonar	67.30	206		71.15	271		71.63	381		70.67	466		66.34	76
Glass	51.40	29		43.92	45		49.06	67		54.20	81		62.14	13
Diabètes	65.49	29		69.53	52		71.22	82		70.31	106		75.26	35
Heart-statelog	68.14	24		69.62	46		70.00	39		66.66	66		77.40	14
Ecoli	53.27	22		60.71	33		65.47	48		64.28	72		63.98	20
Balance-scale	58.88	8		73.76	18		60.32	31		73.12	32		77.12	17
Ionosphere	78.34	95		90.88	131		90.88	157		88.88	195		91.73	121
Unbalanced	98.59	118		98.59	165		98.59	219		98.59	288		98.59	10
Mfeat-fourier	52.95	236		52.75	320		51.40	413		50.90	509		48.15	253

Tableau 19 — Effet de la partition floue sur la méthode SIFRA

Effet des seuils $minsupp$ et β_{min}

Nous notons que le choix des valeurs des seuils $minsupp$ et β_{min} influence significativement les groupes d'attributs générés. Nous menons donc une étude expérimentale pour comprendre l'effet de ces valeurs sur les résultats de SIFRA et savoir quelles valeurs appropriées doit-on choisir pour chaque seuil.

Nous choisissons pour cette étude 6 bases de données avec différents nombres d'attributs et d'observations. Nous avons testé plusieurs valeurs des seuils $minsupp$ et β_{min} .

Concernant le seuil $minsupp$, nous avons testé les valeurs 0.1, 0.2, 0.3, 0.4 et 0.5 pour les trois premières bases de données. Ces dernières sont caractérisées par un nombre d'attributs moyen ou petit (respectivement 4, 8 et 13 attributs pour les bases Iris, Diabetes et Wine). Pour les autres bases de données qui possèdent un nombre d'attributs plus important, des valeurs plus grandes du seuils $minsupp$ ont été choisies. En effet, une valeur faible de $minsupp$ engendre la génération d'un nombre très important d'itemsets fréquents surtout dans les grandes bases. Cela alourdit considérablement la méthode SIFRA. D'autre part, une valeur faible de $minsupp$ engendre la génération d'itemsets plus longs, et donc de groupes d'attributs plus longs. Pour conclure, plus le nombre d'attributs traités est grand, plus on doit augmenter la valeur choisie du seuil $minsupp$.

Pour le seuil β_{min} , son rôle étant d'écarter les groupes qui possèdent des associations faibles ; nous considérons qu'une association de degré inférieur à 0.5 est une association faible. Pour cela, nous avons testé les valeurs 0.6, 0.7, 0.8 et 0.9 pour le seuil β_{min} .

D'après le tableau [20](#), nous notons que le nombre de règles diminue généralement avec l'augmentation des seuils $minsupp$ et β_{min} , mais quelques exceptions figurent dans les bases Iris et Sonar. Nous notons que, en adoptant la méthode de regroupement d'attributs, le nombre de règles atteint sa valeur minimale lorsque chaque attribut est pris seul et est traité indépendamment. Rappelons que le nombre de règles générées par la méthode de regroupement d'attributs est égal à : $NR_{regrp} = \sum_{i=1}^G k^{N_i}$ (avec G est le nombre de groupes d'attributs générés, N_i est le nombre d'attributs réunis dans le groupe G_i et k est la taille de la partition floue). Si chaque attribut est pris seul, alors $N_i = 1$ pour tout les groupes. Donc, sachant que $\sum N_i = N$, le nombre de règles est dans ce cas $k \times N$. Cela explique le fait que le nombre de règles prend une valeur fixe (qui est sa valeur minimale) dès que le seuil $minsupp$ dépasse 0.4 dans le cas des bases Iris, Diabetes et Wine (à partir de cette valeur de $minsupp$, chaque attribut est pris seul et est considéré comme indépendant).

En revanche, une faible valeur du nombre de règles ne coïncide pas toujours avec un bon taux de classification. Prenons par exemple le cas de la base Vehicle, le meilleur taux de bonne classification (67.61%) a été obtenu avec le plus grand nombre de règles (273 règles). De même pour la base Diabetes, le meilleur taux de classification (75.91%) a été obtenu pour un nombre de règles moyennement important.

Nous nous intéressons maintenant à l'effet des deux seuils sur les taux de bonne clas-

	$\beta_{min} \rightarrow$ <i>minsupp</i> \downarrow	0.6		0.7		0.8		0.9	
		Taux	Nb. règles	Taux	Nb. règles	Taux	Nb. règles	Taux	Nb. règles
Iris	0.1	96.00	18	96.00	18	96.00	19	94.66	13
	0.2	95.33	16	94.66	14	94.66	13	94.66	13
	0.3	94.66	13	97.33	14	97.33	14	97.33	14
	0.4	97.33	14	97.33	14	97.33	14	97.33	14
	0.5	97.33	14	97.33	14	97.33	14	97.33	14
Diabetes	0.1	75.52	80	75.91	64	75.26	35	72.01	25
	0.2	74.86	50	74.73	47	74.73	36	71.61	26
	0.3	72.39	34	72.01	28	71.35	22	71.48	21
	0.4	72.01	43	71.87	26	71.35	22	71.48	21
	0.5	72.01	43	71.87	26	71.35	22	71.48	21
Wine	0.1	95.50	68	95.50	53	96.06	51	96.62	44
	0.2	95.50	68	95.50	53	96.06	51	96.62	44
	0.3	98.31	41	98.87	39	98.87	38	98.87	38
	0.4	98.87	38	98.87	38	98.87	38	98.87	38
	0.5	98.87	38	98.87	38	98.87	38	98.87	38
Vehicle	0.15	66.54	295	67.61	273	66.43	200	60.40	133
	0.2	64.65	235	65.72	218	62.41	175	59.33	137
	0.3	64.77	147	63.94	139	64.06	132	62.05	114
	0.4	65.13	131	57.56	95	57.56	95	57.56	95
	0.5	64.77	132	57.09	96	57.09	96	57.09	96
Ionosphere	0.4	86.89	132	91.73	121	93.73	122	93.73	122
	0.5	86.89	132	91.73	121	93.73	122	93.73	122
	0.6	86.89	132	91.73	121	93.73	122	93.73	122
	0.7	91.73	121	91.73	121	93.73	122	93.73	122
Sonar	0.55	65.38	78	66.34	76	66.34	78	66.34	80
	0.6	65.38	78	66.34	76	66.34	78	66.34	80
	0.65	66.34	78	66.34	76	66.34	78	66.34	80
	0.7	66.34	78	66.34	76	66.34	78	66.34	80

Tableau 20 — Effet des seuils *minsupp* et β_{min} sur la méthode SIFRA

sification. Nous remarquons d'après le tableau 20 qu'il n'existe pas une valeur spécifique du seuil *minsupp* qui donne le meilleur résultat dans toutes les bases. La valeur 0.4 de *minsupp* a conduit aux meilleurs taux de bonne classification dans les bases Iris et Wine. Quant aux bases Diabetes et Vehilce, des faibles valeurs de *minsupp* (0.1 et 0.15 respectivement) ont abouti aux meilleurs résultats. D'un autre côté, le seuil *minsupp* n'affecte pas trop les résultats de classification dans les bases de plus grande dimension (Ionosphere et Sonar). C'est seulement lorsque $\beta_{min} = 0.6$ que le taux de bonne classification augmente avec l'augmentation du seuil *minsupp* dans ces bases. Cela peut être dû au fait qu'un nombre important d'itemsets fréquents est généré dans les grandes bases de données et que la variation légère du seuil *minsupp* ne change pas trop le résultat des itemsets générés. Tester des valeurs très différentes de *minsupp* pourrait être envisagé pour pouvoir remarquer leur effet sur les résultats de classification. Quant au seuil β_{min} , nous remarquons que la valeur 0.7 donne toujours le meilleur taux de classification à l'exception de la base Ionosphere (dans cette base, le meilleur taux est obtenu avec $\beta_{min} = 0.8$). Nous remarquons aussi que l'augmentation de la valeur de β_{min} dans les bases Wine, Ionosphere et Sonar améliore la performance de SIFRA. Cependant, avec les autres bases de données (Iris, Diabetes et Vehicle), on ne peut rien conclure sur l'effet de la variation de β_{min} sur le taux de classification.

Pour conclure, il est clair que le choix des valeurs appropriées du seuil *minsupp* dépend fortement des données. Le comportement de ce seuil sur les résultats de classification diffère d'une base à une autre. Le choix de β_{min} dépend aussi des données, mais sur les bases testées, la valeur 0.7 a donné presque toujours les meilleurs taux de bonne classification. Il serait donc intéressant de proposer une méthode de paramétrage automatique qui détermine, pour chaque base, les meilleurs seuils d'une manière automatique.

Effet des méthodes de discrétisation sur la performance de SIFRA et SIFRA-SAR

Il a été mentionné dans le chapitre 3 que le choix de la méthode de discrétisation utilisée dans la phase de regroupement d'attributs peut influencer considérablement les résultats que ce soit en termes des groupes d'attributs générés ou du taux de classification obtenu.

Nous présentons ici une étude expérimentale de l'effet de la méthode de discrétisation choisie sur les résultats de notre méthode SIFRA et aussi sur ses deux extensions "SAR avec Attribut Classe" et "SAR avec facteur d'homogénéité".

Effet des méthodes de discrétisation sur SIFRA

Nous avons mené des tests expérimentaux de SIFRA avec trois méthodes de discrétisation différentes :

- la discrétisation régulière : la méthode EWI avec 3 intervalles
- la discrétisation supervisée : la méthode MDLP de [Fayyad and Irani, 1993](#)
- la discrétisation supervisée-multivariée : la méthode nommée "Clustering+MDLP" de [Gupta et al., 2010](#)

Nous présentons les taux de bonne classification et les groupes d'attributs liés obtenus par ces trois méthodes de discrétisation dans les tableaux 21 et 22, respectivement.

Nous présentons également sous forme graphique (figure 66) les taux de bonne classification obtenus par les trois méthodes de discrétisation afin de bien illustrer l'effet de ces méthodes et de pouvoir les comparer. L'écart relatif des taux de bonne classification obtenus par les deux méthodes de discrétisation suggérées (la discrétisation supervisée et

la discrétisation supervisée-multivariée) par rapport à ceux obtenus par la discrétisation régulière simple est illustré dans la figure [67](#).

	discrétisation régulière	discrétisation supervisée	discrétisation supervisée-multivariée
Iris	97.33	97.33	98.00
Wine	98.87	98.87	98.87
Vehicle	67.61	60.52	59.57
Sonar	66.34	66.82	66.82
Glass	62.14	63.55	64.01
Diabetes	75.91	76.04	74.21
Heart-statlog	77.40	75.18	75.92
Ecoli	63.98	64.28	63.39
Balance-scale	77.12	79.20	80.00
Ionosphere	93.73	94.58	94.30

Tableau 21 — Taux de classification obtenus par les différentes méthodes de discrétisation

	discrétisation régulière	discrétisation supervisée	discrétisation supervisée-multivariée
Iris	[1] [2] [4] [3]	[1] [2] [4] [3]	[3] [1, 4] [2]
Wine	[3, 8]	[3, 8]	[3, 8]
Vehicle	[1, 3, 5, 6, 8] [2, 9, 11, 12, 14] [4, 18] [7, 10] [15, 16]	[14, 17, 18] [5, 15, 16] [1, 4] [2, 6] [10, 13] [8, 9] [7, 11, 12]	[5, 15] [16, 18] [6, 9] [7, 8, 12]
Sonar	[2, 3, 4, 5, 51, 52, 60] [56, 58, 59] [1, 57] [46, 55] [47, 50]	[17, 19, 30, 40, 41, 53, 55, 56, 60] [2, 54] [15, 39, 58] [29, 57] [26, 34] [44, 45]	[2, 53, 55, 56, 57, 58, 60] [7, 29] [54, 59]
Glass	[1, 2, 5, 6, 8] [7, 9]	[2, 3, 4, 6, 7, 8]	[3, 6, 8] [2, 5]
Diabetes	[1, 2, 5, 7] [3, 6] [4, 8]	[1, 5, 7, 8] [2, 6]	[1, 3, 6] [2, 8] [5, 7]
Heart-statlog	[2, 7] [6, 13] [4, 9] [10, 11] [8, 12]	[2, 11] [3, 9, 13] [8, 10, 12]	[2, 9, 13] [8, 11, 12] [3, 10]
Ecoli	[1, 2, 3, 4, 6] [5, 7]	[3, 4, 5, 6, 7] [1, 2]	[2, 3, 4, 6, 7] [1, 5]
Balance-scale	[1, 4] [2, 3]	[1, 2, 4] [3]	[1, 2, 3] [4]
Ionosphere	Aucun groupe	[1, 9, 17, 19, 23, 25, 27, 31]	[1, 15, 17, 19, 23, 25, 27, 29, 31] [2, 21]

Tableau 22 — Groupes d'attributs obtenus par les différentes méthodes de discrétisation

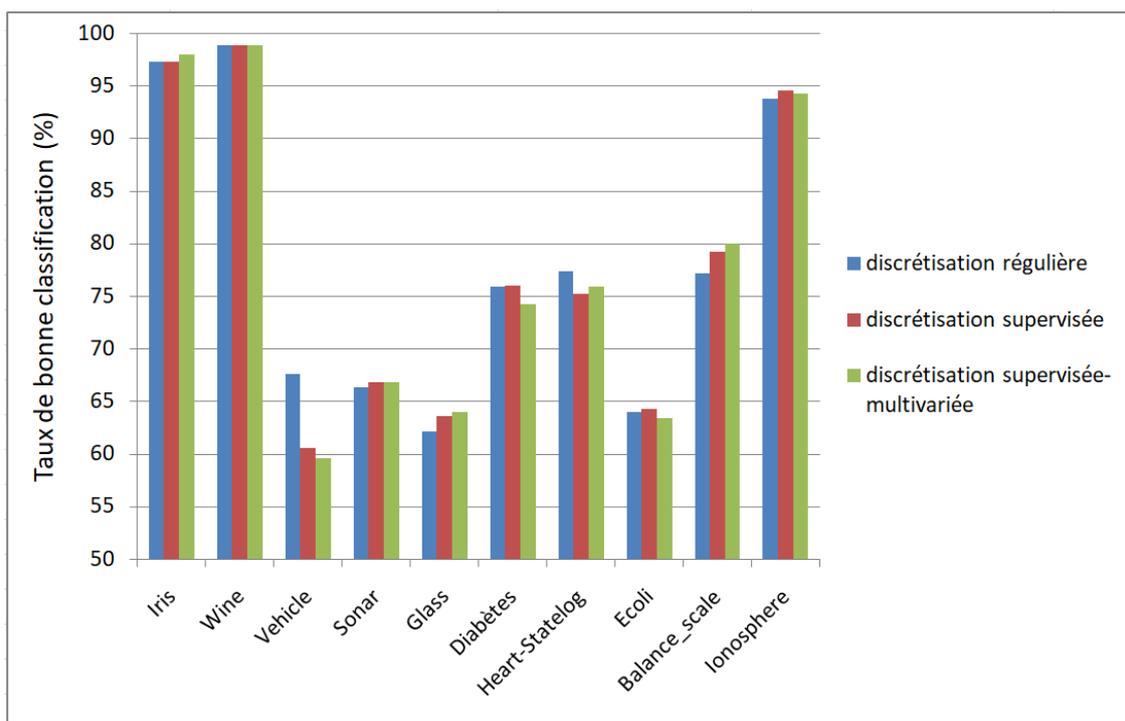


Figure 66 — Effet des méthodes de discrétisation sur la performance de la méthode SIFRA

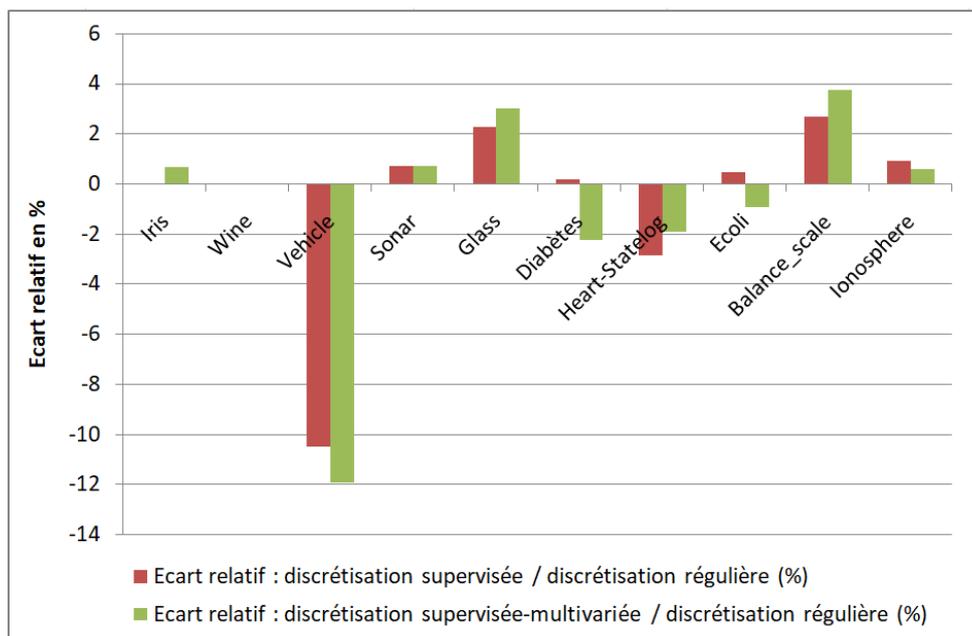


Figure 67 — Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans SIFRA

D'après le tableau 21 et la figure 66, nous remarquons que la discrétisation supervisée-multivariée réalise de meilleurs résultats par rapport aux deux autres méthodes de discrétisation dans le cas des bases de données Iris, Glass et Balance-scale. Une amélioration de 0.68% est obtenue dans la base Iris et plus de 3% est notée dans les bases Glass et Balance-scale, et ce par rapport à la discrétisation régulière (Figure 67). Voyons le cas des bases de données Diabetes, Ecoli et Ionosphere, nous trouvons que les meilleurs taux de bonne classification sont obtenus avec la discrétisation supervisée. La discrétisation supervisée-multivariée est moins performante que la discrétisation régulière dans les deux bases Diabetes et Ecoli. À travers l'écart relatif illustré dans le graphique 67, nous notons aussi que la discrétisation supervisée donne généralement des résultats meilleurs que la discrétisation régulière. Une dégradation des taux de bonne classification obtenus par la discrétisation supervisée est notée dans deux bases seulement (Vehicle et Heart-statlog). Ce résultat peut confirmer que l'utilisation de la même méthode de discrétisation dans les deux premières phases de SIFRA (phase de regroupement d'attributs et phase d'apprentissage) pourrait améliorer les résultats de classification.

Il est aussi à noter que la discrétisation régulière donne parfois des résultats de classification plus intéressants que ceux obtenus par les deux autres méthodes de discrétisation (voir figure 67). Dans le cas de la base Vehicle par exemple, nous notons que la discrétisation supervisée donne une dégradation du taux de classification de 10.5% par rapport à la discrétisation régulière. La discrétisation supervisée-multivariée entraîne aussi une diminution du taux de classification de 11.9% par rapport à la discrétisation régulière. Avec la base Heart-statlog, la discrétisation régulière s'avère aussi plus intéressante que les deux autres méthodes de discrétisation (une diminution de 2.9% est notée par la discrétisation supervisée et de 1.9% par la discrétisation supervisée-multivariée). Finalement, nous notons qu'avec la base Wine, les trois méthodes de discrétisation réalisent les mêmes taux de classification. Ceci s'explique par le fait que les mêmes groupes d'attributs sont obtenus par les trois méthodes (voir Tableau 22). Ce tableau montre aussi que généralement des groupes d'attributs différents sont obtenus par les trois méthodes de discrétisation. Ceci confirme que le choix de la méthode de discrétisation a un impact direct sur les groupes d'attributs liés générés.

Pour conclure, nous notons que le choix d'une discrétisation supervisée-multivariée peut être intéressant pour certaines bases de données (comme Iris, Glass et Balance-scale). Cependant, il s'avère que la discrétisation supervisée présente aussi un choix intéressant dans plusieurs autres bases de données. Ceci peut être expliqué par le fait que la même méthode de discrétisation est utilisée dans la phase d'apprentissage de SIFRA. Néanmoins, dans les bases Vehicle et Heart-statlog, la discrétisation régulière présente des résultats très satisfaisants par rapport aux deux autres méthodes. Finalement, le choix de la méthode de discrétisation appropriée s'avère être dépendant des données. Par conséquent, il serait intéressant d'élargir les bases de tests et d'analyser la nature des données testées afin de confirmer ce résultat.

Effet des méthodes de discrétisation sur les approches SIFRA-SAR

Comme la méthode SIFRA-SAR utilise une approche de regroupement d'attributs supervisé, la prise en compte de l'attribut *Classe* dans la phase de discrétisation pourrait aussi être intéressant.

Dans ce contexte, nous proposons de tester les trois méthodes de discrétisation (régulière, supervisée, et supervisée-multivariée) dans les deux versions de l'approche SAR : "SAR avec Attribut Classe" et "SAR avec facteur d'homogénéité". Les figures [68](#), [69](#), [70](#) et [71](#) montrent, sous forme graphique, l'effet de la méthode de discrétisation utilisée dans les deux méthodes "SAR avec attribut Classe" et "SAR avec facteur d'homogénéité", respectivement.

En utilisant l'approche de regroupement d'attributs "SAR avec l'attribut Classe", les figures [68](#) et [69](#) montrent que la discrétisation supervisée-multivariée paraît plus intéressante que la discrétisation régulière uniquement pour les bases de données Iris, Balance-scale et Ionosphere. Nous remarquons également que la méthode de discrétisation supervisée améliore légèrement la performance de la méthode SAR par rapport à la discrétisation régulière dans le cas des bases Sonar, Balance-scale et Ionosphere. Pour le reste des bases de données (Vehicle, Glass, Diabetes, Heart-statlog et Ecoli), la discrétisation régulière semble être la plus appropriée et donne les meilleurs résultats de classification.

En ce qui concerne la méthode "SAR avec facteur d'homogénéité", les figures [70](#) et [71](#) montrent une amélioration légère des taux de bonne classification dans les bases de données Glass, Diabetes et Balance-scale, et cela lorsque la discrétisation supervisée-multivariée est adoptée. Cependant, ce type de discrétisation dégrade la performance de la méthode SAR dans le cas des bases Iris, wine, Heart-statlog, Ionosphere, Vehicle et Ecoli. Dans ces deux dernières bases de données, la discrétisation supervisée donne les meilleurs taux de classification. Finalement, la discrétisation régulière apparaît comme la meilleure méthode de discrétisation dans plusieurs bases de données comme Wine, Heart-statlog et Ionosphere.

Pour conclure, comme le cas de la méthode SIFRA, l'utilisation d'une méthode de discrétisation supervisée-multivariée dans le contexte des méthodes SAR semble intéressante dans quelques bases de données. Quant à la discrétisation supervisée, elle s'avère plus intéressante lorsqu'elle est utilisée avec SIFRA. Elle dégrade la performance de classification dans la plupart des bases de données (Vehicle, Glass, Diabetes, Heart-statlog, Ecoli) lorsqu'elle est appliquée avec SIFRA-SAR. D'un autre côté, la discrétisation régulière présente les meilleurs choix dans plusieurs bases de données dans le contexte des méthodes SIFRA-SAR. Nous pouvons donc déduire que combiner la discrétisation supervisée avec une méthode de regroupement d'attributs supervisée n'est pas toujours intéressant. Le résultat obtenu pourrait être dépendant aux données et pour le confirmer il faudrait tester d'autres bases de données.

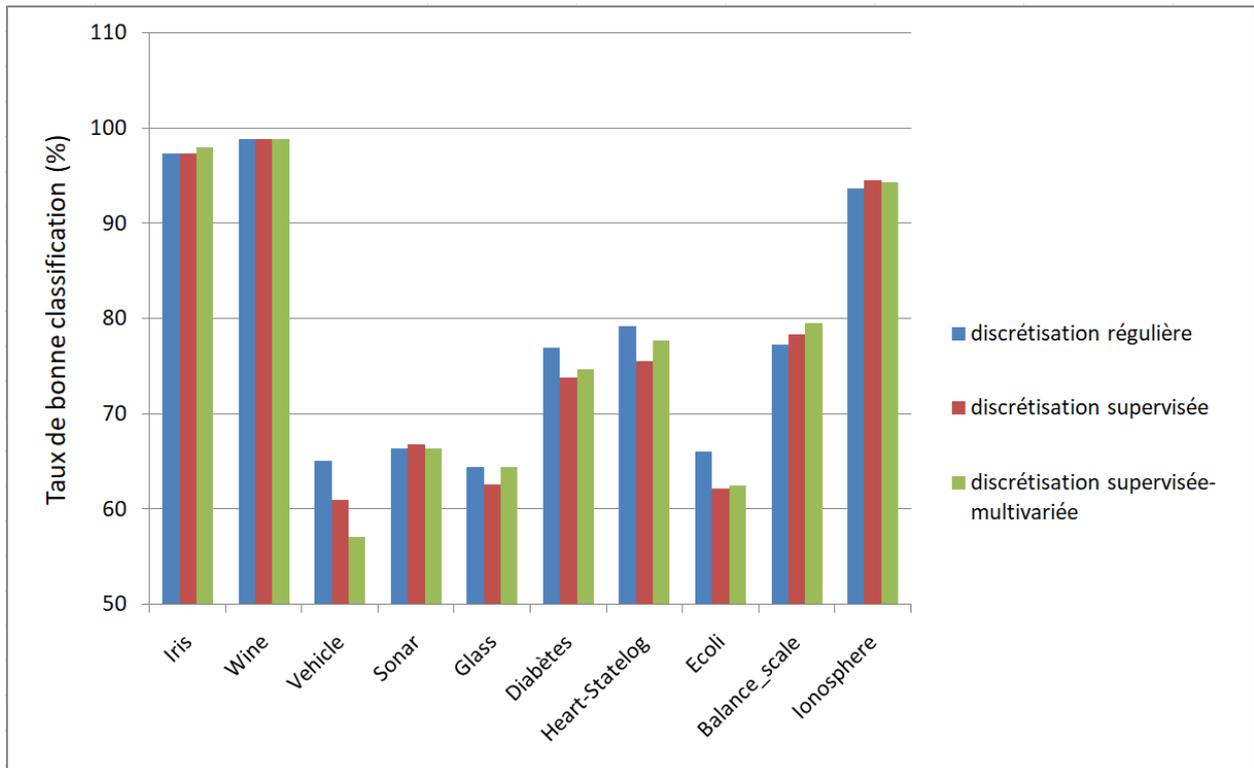


Figure 68 — Effet des méthodes de discrétisation sur la performance de la méthode "SAR avec l'attribut Classe"

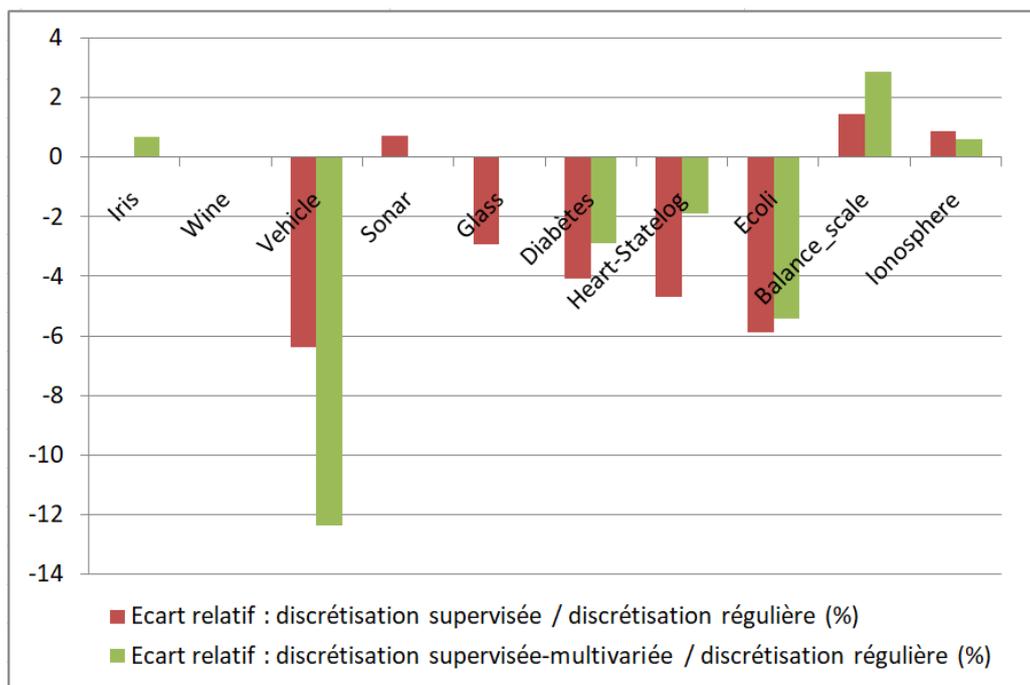


Figure 69 — Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans "SAR avec l'attribut Classe"

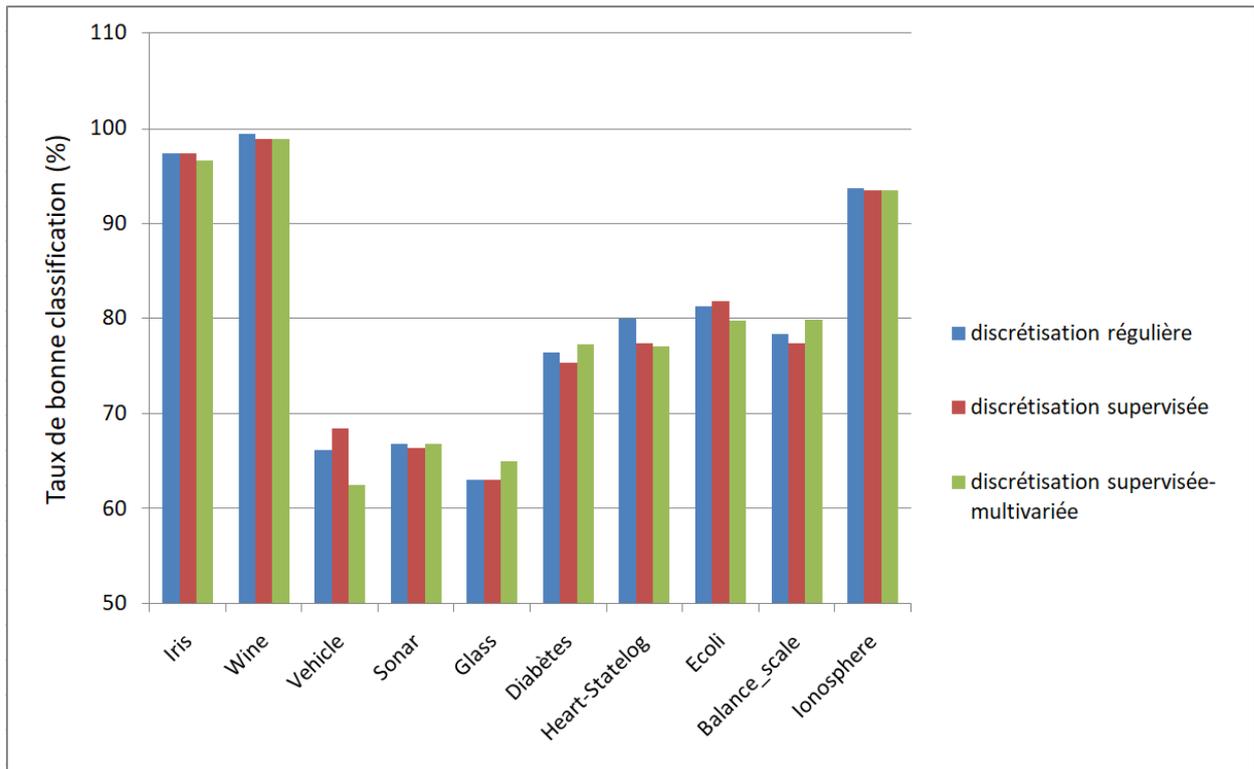


Figure 70 — Effet des méthodes de discrétisation sur la performance de la méthode "SAR avec facteur d'homogénéité"

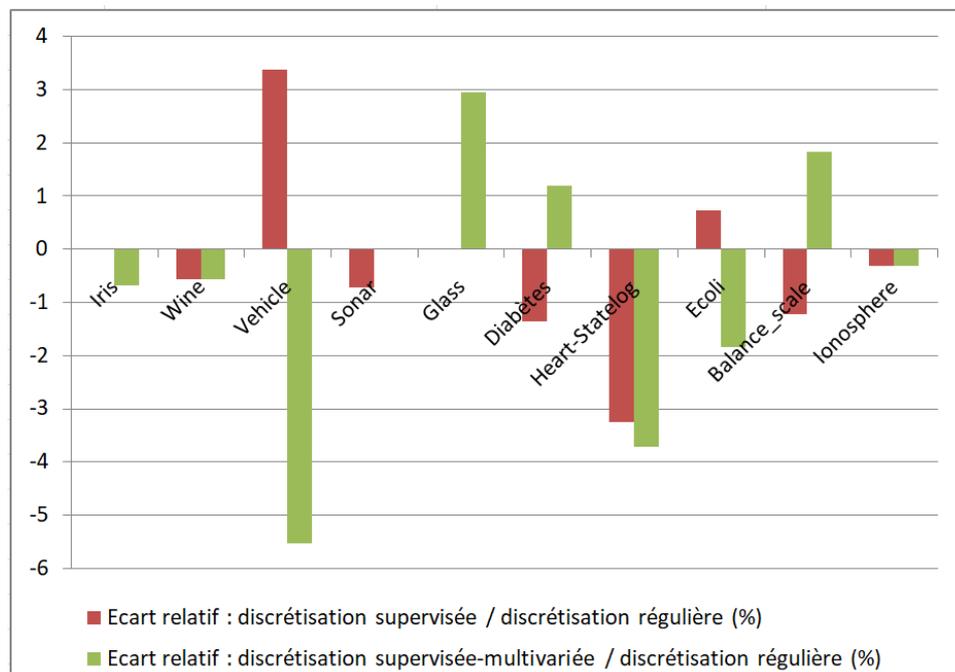


Figure 71 — Écart relatif des taux de classification des discrétisations supervisée et supervisée-multivariée par rapport à la régulière dans "SAR avec facteur d'homogénéité"