Réduction de dimension de sac de mots visuels grâce à l'analyse formelle de concepts

Ngoc Bich Dao

▶ **To cite this version:**

Ngoc Bich Dao. Réduction de dimension de sac de mots visuels grâce à l'analyse formelle de concepts. Traitement des images [eess.IV]. Université de La Rochelle, 2017. Français. NNT : 2017LAROS010 . tel-01753800

HAL Id: tel-01753800
https://theses.hal.science/tel-01753800v1

Submitted on 29 Mar 2018

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.

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE S2IM

LABORATOIRE : L3I

THÈSE présentée par :

Ngọc Bích ĐÀO

soutenue le : **23 juin 2017**

pour obtenir le grade de : **Docteur de l'université de La Rochelle**

Discipline : **Informatique et Applications**

Réduction de dimension de sac de mots visuels grâce à l'Analyse Formelle de Concepts.

RAPPORTEURS	**Marianne HUCHARD**	Professeur, LIRMM, Université de Montpellier
	Engelbert MEPHU-NGUIFO	Professeur, LIMOS, Université Blaise Pascal
EXAMINATEURS	**Frédéric PRECIOSO**	Professeur, I3S, Université Nice Sophia-Antipolis
	Nicolas PASQUIET	Maître de conférences, I3S, Université de Nice Sophia-Antipolis
	Nathalie GIRARD	Maître de conférences, IRISA, ISTIC, Université Rennes 1
	David PICARD	Maître de conférences, ETIS, École ENSEA
DIRECTION	**Karell BERTET**	Maître de conférences HDR, L3i, Université de La Rochelle
	Arnaud REVEL	Professeur, L3i, Université de La Rochelle

Thèse réalisée au Laboratoire Informatique, Image, Interaction
Faculté des Sciences et Technologies
Université de La Rochelle
Avenue Michel Crépeau
17042 La Rochelle cedex 01

Tél : +33 5 46 45 82 62
Fax : +33 5 46 45 82 42

Web : http ://l3i.univ-larochelle.fr

Sous la direction de Karell BERTET karell.bertet@univ-lr.fr

Co-encadrement Arnaud REVEL arnaud.revel@univ-lr.fr

Financement Programme Erasmus Mundus MOVER

Résumé

La réduction des informations redondantes et/ou non-pertinentes dans la description de données est une étape importante dans plusieurs domaines scientifiques comme les statistiques, la vision par ordinateur, la fouille de données ou l'apprentissage automatique. Dans ce manuscrit, nous abordons la réduction de la taille des signatures des images par une méthode issue de l'Analyse Formelle de Concepts (AFC), qui repose sur la structure du treillis des concepts et la théorie des treillis. Les modèles de sac de mots visuels consistent à décrire une image sous forme d'un ensemble de mots visuels obtenus par clustering. La réduction de la taille des signatures des images consiste donc à sélectionner certains de ces mots visuels. Dans cette thèse, nous proposons deux algorithmes de sélection d'attributs (mots visuels) qui sont utilisables pour l'apprentissage supervisé ou non.

Le premier algorithme, RedAttSansPerte, ne retient que les attributs qui correspondent aux irréductibles du treillis. En effet, le théorème fondamental de la théorie des treillis garantit que la structure du treillis des concepts est maintenue en ne conservant que les irréductibles. Notre algorithme utilise un graphe d'attributs, le graphe de précédence, où deux attributs sont en relation lorsque les ensembles d'objets à qui ils appartiennent sont inclus l'un dans l'autre. Nous montrons par des expérimentations que la réduction par l'algorithme RedAttsSansPerte permet de diminuer le nombre d'attributs tout en conservant de bonnes performances de classification.

Le deuxième algorithme, RedAttsFloue, est une extension de l'algorithme RedAttsSansPerte. Il repose sur une version approximative du graphe de précédence. Il s'agit de supprimer les attributs selon le même principe que l'algorithme précédent, mais en utilisant ce graphe flou. Un seuil de flexibilité élevé du graphe flou entraîne mécaniquement une perte d'information et de ce fait une baisse de performance de la classification. Nous montrons par des expérimentations que la réduction par l'algorithme RedAttsFloue permet de diminuer davantage l'ensemble des attributs sans diminuer de manière significative les performances de classification.

Mots clés : réduction de dimension, sélection d'attributs, treillis, irréductible, analyse formelle de concepts, modèle de sac de mots visuels, graphe de précédence, graphe de précédence flou, méthode algébrique, logique floue.

Dimension reduction on bag of visual words with Formal Concept Analysis.

Abstract

In several scientific fields such as statistics, computer vision and machine learning, redundant and/or irrelevant information reduction in the data description (dimension reduction) is an important step. This process contains two different categories: feature extraction and feature selection, of which feature selection in unsupervised learning is hitherto an open question. In this manuscript, we discussed about feature selection on image datasets using the Formal Concept Analysis (FCA), with focus on lattice structure and lattice theory. The images in a dataset were described as a set of visual words by the bag of visual words model. Two algorithms were proposed in this thesis to select relevant features and they can be used in both unsupervised learning and supervised learning.

The first algorithm was the RedAttSansPerte, which based on lattice structure and lattice theory, to ensure its ability to remove redundant features using the precedence graph. The formal definition of precedence graph was given in this thesis. We also demonstrated their properties and the relationship between this graph and the AC-poset. Results from experiments indicated that the RedAttsSansPerte algorithm reduced the size of feature set while maintaining their performance against the evaluation by classification.

Secondly, the RedAttsFloue algorithm, an extension of the RedAttsSansPerte algorithm, was also proposed. This extension used the fuzzy precedence graph. The formal definition and the properties of this graph were demonstrated in this manuscript. The RedAttsFloue algorithm removed redundant and irrelevant features while retaining relevant information according to the flexibility threshold of the fuzzy precedence graph. The quality of relevant information was evaluated by the classification. The RedAttsFloue algorithm is suggested to be more robust than the RedAttsSansPerte algorithm in terms of reduction.

Keywords: dimension reduction, feature selection, lattice, irreducible, formal concept analysis, bag of visual words model, precedence graph, fuzzy precedence graph, algebraic method, fuzzy logic.

Remerciements

Je tiens tout d'abord à remercier Marianne Huchard et Engelbert Mephu-Nguifo d'avoir accepté d'être les rapporteurs de cette thèse. Leurs commentaires, leurs questions et les échanges menés pendant le processus de relecture m'ont été très utiles pour la mis à jour de ce manuscrit pour aboutir à cette version finale.

Je tiens également à remercier Frédéric Precioso, Nicolas Pasquier, Nathalie Girard et David Picard d'avoir accepté de participer à mon jury en tant qu'examinateurs. Leurs commentaires et discussions pendant la soutenance sont une source d'inspiration importante pour la poursuite de travaux de ma thèse.

Je tiens à remercier Karell Bertet et Arnaud Revel qui m'ont encadrée au cours de cette thèse. Leurs thématiques de recherche différentes m'ont permis de m'enrichir et d'apprendre énormément. Grâce à eux, j'ai pu participé à des conférences internationales et d'avoir des discussions intéressantes sur mon sujet. Je voudrais les remercier pour leurs encadrement scientifique, aussi bien que les échanges amicalement au début de ma thèse. Je voudrais remercier particulier Karell pour sa patience et sa gentillesse, et Arnaud pour son nettement. Ils m'ont aidée à apprendre à rédiger le manuscrit de manière plus claire.

Je voudrais de remercier ceux qui m'ont fait profiter de leur expérience de d'enseignement Pedro, Armelle, Bernard, Karell et Arnaud.

Je tiens à remercier les personnes du service du laboratoire de m'avoir accueilli et facilité ma vie administrative (Geneviève, Erlandri, Dominique, Kathy, etc.). Je voudrais aussi remercier la direction du laboratoire (Yacine, Jean-Christophe) et le directeur de l'école doctorale (Alain Gaugue) de m'avoir soutenus et encouragés.

Je tiens à remercier Michel, Christophe D. et tous les amis (doctorants ou pas) (Audrey, Christophe, Clément, Cyrille S., Dounia, Elodie, Guillaume, Hien, Joseph, Marcela, Nouredine, Oanh, Phuong, Sébastien, Vincent, ainsi que tous ceux qui ne sont pas dans cette liste mais qui se reconnaîtrons) pour leurs soutiens pendant ces longues années de thèse. Sans eux, ce document n'aurai pas pu voir le jour. Merci Elodie et Guillaume d'avoir lu mon manuscrit et m'avoir donnés de bons conseils pour améliorer mon manuscrit. Merci Chloé, Damien, Lionel et Wafa pour leurs fraîcheur dans l'ambiance du laboratoire.

Je tiens à remercier ma famille, et tout particulièrement mon cher mari de m'avoir supporté et de m'avoir soutenu pendant toute la longueur de la thèse, dans les moments les meilleurs comme les pires. Je remercie aussi mon beau père pour son courage d'avoir

lu mon manuscrit et m'avoir fait des retours intéressants d'un point de vue étranger au domaine dans un délai record.

Merci aux personnes ayant travaillé sur l'ensemble des logiciels et langages de programmation libres que j'ai pu utiliser durant cette thèse (LATEX, Java, C++, Python, etc.).

Et plus globalement, je remercie toutes les personnes avec lesquelles mes rapports furent aussi divers qu'enrichissant.

Table des matières

Résumé i

Abstract v

Remerciements vii

Table des matières ix

Table des figures xiii

Liste des tableaux xix

Introduction générale 1

État de l'art 9

1 Etat de l'art sur la réduction de dimension 11

1.1 Introduction 11
1.2 Types des données 13
1.3 Approche par extraction d'attributs 14
1.3.1 Application au cas de l'apprentissage supervisé 15
1.3.2 Application au cas de l'apprentissage non supervisé 16
1.3.2.1 Analyse en Composantes Principales (ACP) 16
1.3.2.2 Poursuite de Projection (PP) 18
1.3.2.3 Analyse en Composantes Indépendantes (ACI) 19
1.3.2.4 Analyse en facteurs booléens (Boolean Factor Analysis) . . 19
1.4 Approche par sélection d'attributs 20
1.4.1 Application au cas de l'apprentissage supervisé 21
1.4.1.1 Approche par filtrage 23

1.4.1.2 Approches par encapsulation (Wrapper) . . . 27
1.4.1.3 Approches par intégration (embedded) . . . 31
1.4.2 Application au cas de l'apprentissage non supervisé . . . 34
1.4.2.1 Approches par filtrage . . . 35
1.4.2.2 Approches par encapsulation (Wrapper) . . . 37
1.4.2.3 Approches par intégration (embedded) . . . 39
1.5 Conclusion . . . 43
Points clés . . . 46

2 Analyse Formelle de Concepts 47
2.1 Introduction . . . 47
2.2 Notions . . . 48
2.2.1 Contexte formel et treillis des concepts / treillis de Galois . . . 49
2.2.2 Sous-hiérarchie de Galois (AOC-poset) et AC-poset . . . 56
2.2.3 Système de fermeture et treillis des fermés . . . 58
2.3 Discussion . . . 62
Points clés . . . 65

3 Modèle de sac de mots visuels 67
3.1 Introduction . . . 67
3.2 Extraction des caractéristiques d'images . . . 69
3.2.1 Scale-Invariant Feature Transform (SIFT) . . . 70
3.2.1.1 Détection des extrema dans l'espace des échelles . . . 71
3.2.1.2 Localisation des points d'intérêt . . . 72
3.2.1.3 Affectation d'orientation . . . 74
3.2.1.4 Description des points d'intérêt . . . 75
3.2.2 Color Moment Invariants (CMI) . . . 76
3.2.3 Détecteur de Harris-Laplace . . . 77
3.3 Construction d'un dictionnaire des mots visuels . . . 78
3.4 Encodage des caractéristiques d'une image dans un descripteur d'images (sac de mots visuels) . . . 79
3.5 Conclusion . . . 80
Points clés . . . 81

Réduction de dimension 83

4 Réduction de dimension 85
4.1 Introduction . . . 85

4.2 Normalisation et binarisation . . . 86
4.2.1 Normalisation . . . 87
4.2.1.1 Normalisation par ligne (max) . . . 87
4.2.1.2 Normalisation par colonne . . . 89
4.2.1.3 Normalisation par ligne (somme) . . . 89
4.2.2 Binarisation . . . 90
4.3 Réduction exacte des attributs . . . 91
4.3.1 Introduction . . . 91
4.3.2 Graphe de précédence . . . 93
4.3.3 Algorithme RedAttsSansPerte . . . 97
4.3.3.1 Clarification exacte . . . 99
4.3.3.2 Standardisation exacte ($|E_x| = 0$) . . . 100
4.3.3.3 Réduction exacte ($|E_x| > 1$) . . . 100
4.4 Réduction floue des attributs . . . 103
4.4.1 Graphe de précédence flou . . . 104
4.4.2 Algorithme RedAttsfloue . . . 109
4.4.2.1 Clarification floue . . . 109
4.4.2.2 Réduction floue . . . 113
4.4.2.3 Algorithme RedAttsFloue . . . 116
4.5 Discussion . . . 120
Points clés . . . 121

5 Expérimentations et évaluations 123
5.1 Contexte . . . 123
5.1.1 Méthode d'évaluation . . . 124
5.1.2 Description des données . . . 126
5.1.2.1 Caltech-256 . . . 126
5.1.2.2 Pascal (VOC) . . . 126
5.1.2.3 MIR flickr . . . 127
5.1.2.4 COREL . . . 129
5.1.2.5 NIPS2003 . . . 130
5.2 Evaluation de l'algorithme RedAttsSansPerte . . . 132
5.2.1 Point de vue quantitatif . . . 132
5.2.1.1 Réduction par l'algorithme RedAttsSansPerte . . . 132
5.2.1.2 Effets de l'étape de normalisation sur la réduction . . . 143
5.2.1.3 Effet de l'étape de création des sacs de mots visuels sur la réduction . . . 143
5.2.2 Point de vue qualitatif . . . 147
5.2.2.1 Réduction par classe . . . 147
5.2.2.2 Évaluation par la F-mesure . . . 149

5.3 Evaluation de l'algorithme RedAttsFloue . . . 151
5.3.1 Point de vue quantitatif . . . 151
5.3.1.1 Seuil de flexibilité fixe . . . 151
5.3.1.2 Variation du seuil de flexibilité . . . 154
5.3.2 Point de vue qualitatif . . . 155
5.4 Conclusion . . . 158
Points clés . . . 160

Conclusions et perspectives **161**

Annexes **167**
A.1 Mesures de la performance de classification . . . 169
A.1.1 F-mesure (F-score) . . . 169
A.1.2 Balanced Error Rate (BER) . . . 170
B.1 Résultat de réduction de l'algorithme RedAttsFloue . . . 171
C.1 Isomorphisme de graphes . . . 175
C.2 Catégorisation des méthodes de réduction de dimension . . . 175
C.2.1 Méthode statistique . . . 175
C.2.2 Méthode logique . . . 176
C.3 Matrice d'affinité (Affinity matrix) . . . 176
C.4 Indépendance statistique (Statistical independence) . . . 176
C.5 Distribution gaussienne . . . 177

Bibliographie **179**

Table des figures

1 Illustration schématique présentant de manière générale la chaîne de traitement à partir d'une base d'images afin d'obtenir les sacs de mots visuels réduits en trois étapes : l'extraction de signature des images, le pré-traitement de données discrètes à données binaires, la réduction de dimension des signatures des images. 2

2 Illustration présentant de manière générale la représentation des images sous forme de sacs de mots visuels. Les images sont prises à partir de la base d'images CALTECH256 [Griffin 2007]. 3

1.1 Illustration schématique présentant de manière générale la réduction de dimension. 12

1.2 Les trois approches de sélection des attributs : approche par filtrage, approche par encapsulation et approche par intégration. 22

1.3 L'ACP est parfois un discriminateur pauvre. Cette figure est issue du travail de [Fukunaga 1990]. Dans cet exemple, les points carrés oranges et les points ronds bleus représentent deux groupes différents dans un espace de deux dimension x et y. Pour passer de deux dimensions à une dimension, l'ACP choisit la projection qui a la variance la plus haute de sorte que la dimension choisie soit l'axe a. Cependant, l'axe a n'est pas un bon discriminant entre deux groupes. Pour distinguer ces deux groupes, l'axe b est mieux que l'axe a. 44

2.1 Treillis des concepts correspondant au contexte 2.1. 51

2.2 Table des irréductibles du treillis des concepts du contexte 2.1. Les infimum-irréductibles de ce treillis des concepts sont entourés en pointillés rouges et les supremum-irréductibles de ce treillis des concepts sont remplis en points bleus. 52

2.3 Illustration du théorème de Barbut en 1970. 56

2.4 Sous-hiérarchie de Galois correspondant au treillis des concepts du contexte 2.1. 57

2.5 Sous-hiérarchie de Galois et l'OC-poset, l'AC-poset correspondant au contexte 2.1. 57
2.6 Sous-hiérarchie de Galois et l'OC-poset, l'AC-poset correspondant au contexte 2.1 réduit. 58
2.7 Treillis des concepts et treillis des fermés correspondant au contexte 2.1. . 60
2.8 Application du théorème 2.2.1 sur le treillis des concepts et le treillis des fermés réduits correspondant au contexte 2.1. 61
2.9 Synthèse : le treillis des concepts, les treillis des fermés, l'AOC-poset et l'AC-poset, l'OC-poset correspondant au contexte 2.1. 62
2.10 Synthèse : le treillis des concepts, le treillis des fermés et l'AC-poset correspondant au contexte 2.1 et au contexte attributs-réduits. 63

3.1 Approche classique permettant de construire le modèle de sac de mots visuels. 68
3.2 Pour chaque octave de l'espace d'échelle, la convolution de l'image par un filtre gaussien de paramètre σ est répétée et produit des images filtrées par une gaussienne qui sont présentées à gauche. Les images filtrées par une gaussienne côte à côte sont soustraites comme dans l'équation 3.2.1.1 pour produire l'image de DoG à droite. A chaque octave, l'image filtrée par une gaussienne est sous échantillonnée d'un facteur 2, et le processus est répété jusqu'au traitement de toutes les octaves. Cette figure est issue de la figure de l'article de Lowe [Lowe 2004]. 72
3.3 Comparaison d'un pixel (marqué avec X) avec ses 26 voisins (marqué avec des cercles) dans une zone de 3 échelles $\times$ 3 espaces servant à détecter les maximums et les minimums locaux. Cette figure est issue du travail de Lowe [Lowe 2004]. 73
3.4 Histogramme d'orientations avec 36 boîtes contenant $360°(2\pi)$. Cette figure est inspirée du travail de Lowe [Lowe 2004]. 74
3.5 Un descripteur de points d'intérêt est créé en calculant la magnitude du gradient et l'orientation des voisins autour de la position du point d'intérêt, comme présenté à gauche. La fenêtre circulaire Gaussienne est représentée par le cercle bleu superposé. Les valeurs de l'orientation du gradient et la fenêtre circulaire gaussienne des voisins dans une sous-région de la taille 4×4 sont cumulées dans un histogramme d'orientation représenté au milieu. Cette figure illustre un ensemble de descripteurs de point d'intérêt de taille 2×2 (4 sous-régions) à partir d'un ensemble de 8×8 voisins. Dans son article, Lowe utilise un ensemble de descripteurs de taille 4×4 à partir d'un ensemble de 16×16 voisins. Cette figure est inspirée du travail de Lowe [Lowe 2004]. 75

4.1 Sous-hiérarchie de Galois et les graphes de précédences correspondant au contexte 4.7 (contexte 2.1 du chapitre 2). 95
4.2 Isomorphisme entre l'AC-poset et le graphe de précédence sur l'ensemble des attributs correspondant au contexte 4.7 (contexte 2.1 du chapitre 2). . 97
4.3 La relation entre un contexte, son treillis des concepts et son contexte attributs-réduits. 98
4.4 Graphe de précédences $G_A(\mathcal{A}, E_A)$ correspondant au contexte 4.7. 98
4.5 Graphe de précédence flou $\tilde{G}_A(A, \tilde{E}_A)$ avec $\delta = 0.2$ correspondant au contexte 4.7 (contexte 2.1 du chapitre 2). 104
4.6 Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs. 111
4.7 Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs. 112
4.8 Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs. 113
4.9 Relations entre les sommets x et y, leurs prédécesseurs et leurs successeurs. 114

5.1 Chaîne de traitement finale réalisée par nos algorithmes en partant de la base d'images et en allant vers le sac de mots visuels réduit. 124
5.2 Chaîne de traitement finale réalisée par l'algorithme RedAttsSansPerte en partant du sac de mots visuels et en allant vers le sac réduit de mots visuels. 125
5.3 Base de données CALTECH256. 127
5.4 Base de données VOC2012. 128
5.5 Base de données MIR fmickr. 129
5.6 Base de données COREL. 130
5.7 Chaîne de traitement réalisée par nos algorithmes en partant de la base d'images et en allant vers le sac réduit de mots visuels. 133
5.8 Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur trois bases de données : VOC2005, CALTECH256 et COREL5k. 135
5.9 Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur deux bases de données : VOC2012 et MIRflickr25000. 136
5.10 Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur trois bases de données : Arcene, Dexter, Gisette. La ligne pointillée est le ratio du nombre des bruits ajoutés sur le nombre des attributs initiaux. 137

5.11 Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_1, la courbe rouge (trait plein), FA_2, la courbe bleue (trait tiret), et FA_3, la courbe vert (trait points et tirets alternés). 139
5.12 Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_1, la courbe rouge (trait plein), FA_2, la courbe bleu (trait tiret), et FA_3, la courbe vert (trait points et tirets alternés). 140
5.13 Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_{12}, la courbe rouge (trait plein), FA_{23}, la courbe bleu (trait tiret), et FA_{34}, la courbe vert (trait points et tirets alternés). 141
5.14 Croisements entre le nombre d'attributs réduits (la courbe bleu, trait tiret) et le nombre d'images réduites (la courbe rouge, trai plein) en fonction du seuil de binarisation dans le cas d'une normalisation par ligne ou d'une normalisation par colonne. 142
5.15 Fraction d'Attributs réduits par l'algorithme RedAttsSansPerte sur le sac de 500 mots visuels de la base de données CALTECH256 en fonction du type de normalisation. 144
5.16 Chaîne de traitement classique permettant d'obtenir le modèle de sac de mots visuels. Les méthodes utilisées sont SIFT, SIFT, K-means et FLANN-based pour les étapes de détection, description, clustering et encodage respectivement. 145
5.17 Différents capacités de réduction d'attributs de l'algorithme RedAttsSansPerte sur la base de données VOC2012 avec différentes méthodes de construction du sac de mots visuels. 146
5.18 Comparaison du résultat de F-mesure sur les données avant et après l'application de l'algorithme de réduction RedAttsSansPerte. La courbe est la Fraction d'Attributs réduits (FA) de l'algorithme RedAttsSansPerte. . . 150
5.19 Proportion d'attributs réduits de l'algorithme de réduction RedAttsFloue et proportion d'attributs réduits par seuil de binarisation sur l'ensemble initial d'attributs des trois ensembles de données Arcene, Dexter et Gisette. 152
5.20 Fraction d'Attributs réduits (FA) de deux dernières étapes de l'algorithme RedAttsFloue avec le seuil de flexibilité est égal à 0.1 : FA_4, l'étape de la réduction floue, FA_5, l'étape de la clarification floue. 153
5.21 Fraction d'Attributs réduits des deux étapes floues de l'algorithme de réduction RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de binarisation [0, 0.8] et du seuil de flexibilité [0.1, 0.9]. 155
5.22 Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux par l'algorithme RedAttsFloue (où le seuil de binarisation est égal à 0) en fonction du seuil de flexibilité : FA_{45}, l'étape de la *réduction floue*, FA_{56}, l'étape de la *clarification flou*. 156

5.23 Comparaison du résultat de FA et de F-mesure sur les données au seuil de binarisation 0.0 de deux algorithmes de réduction : RedAttsSansPerte et RedAttsFloue en fonction du seuil de flexibilité. 157

Liste des tableaux

1.1 Le tableau de contingence permettant d'obtenir la dépendance entre la classe c $\{c_P, c_N\}$ et l'attribut f $\{v_1,.., v_r\}$. 23

1.2 Synthèse des méthodes de réduction de dimension selon quatre propriétés : la nature de ces méthodes, le cadre d'application de ces méthodes, le type de la méthode et le type de données d'entrée de ces méthodes. 45

2.1 Contexte formel. 49

4.1 Illustration d'une transformation de M_{nm} en $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$. 87

4.2 Données initiales. 88

4.3 Après normalisation par NormLigneMax. 88

4.4 Après normalisation par NormColonne. 89

4.5 Après normalisation par NormLigneSomme. 90

4.6 Illustration pour la binarisation. 91

4.7 Contexte formel (exemple 2.1 du chapitre 2). 93

4.8 Exemple pour la réduction d'attributs avec le graphe de précédence exact 4.4 (étape de réduction). 101

5.1 Tableau de synthèse représentant les jeux des données de la base NIPS 2003. 131

5.3 Nombre d'attributs restants (sur les 262 attributs initiaux) après réduction par catégorie pour la base VOC 2005 - dataset 1, en fonction du seuil de binarisation. La dernière colonne donne le nombre d'attributs subsistant si on considère l'ensemble des 4 classes. Dans ce cas, on ne peut pas supprimer un attribut s'il est utilisé pour catégoriser des classes. De ce fait, il faut prendre l'union de tous les attributs. 148

5.4 Nombre d'attributs restant (sur les 1000 attributs initiaux) après réduction par catégorie pour la base CALTECH256, en fonction du seuil de binarisation. La dernière colonne donne le nombre des attributs subsistant pour l'union des attributs restants des 256 catégories. La réduction se fait sur le sac normalisé de mots visuels par la normalisation NormLigneMax. 149
5.5 Fraction d'Attributs réduits par les deux étapes 4 et 5 de l'algorithme RedAttsFloue sur la base Dexter avec des seuils de binarisation égaux à 0.0 et 0.1. 154

A.1 Matrice de confusion. 169

B.1 Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de flexibilité et du seuil de binarisation. 171
B.2 Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de flexibilité et du seuil de binarisation. 172
B.3 Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Dexter en fonction du seuil de flexibilité et du seuil de binarisation. 172
B.4 Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Dexter en fonction du seuil de flexibilité et du seuil de binarisation. 173
B.5 Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Gisette en fonction du seuil de flexibilité et du seuil de binarisation. 173
B.6 Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Gisette en fonction du seuil de flexibilité et du seuil de binarisation. 174

Introduction générale

Cadre général

L'agrégation de données en vue de leur analyse est une activité essentielle dans les domaines scientifique. La diversité des domaines scientifiques produit des données très hétérogènes en termes de type et d'informations portées. De nombreux projets ont été menés au sein de notre laboratoire [1] pour extraire des éléments sémantiques à partir d'images provenant d'horizons différents, comme par exemple les images de lettrines [Coustaty 2011], les documents manuscrits [Prum 2013], les images naturelles [Awad 2015], les planches de bandes dessinées [Guerin 2013, Rigaud 2015], les documents numériques [Eskenazi 2017]. En ce qui nous concerne, nos travaux se focalisent principalement sur l'analyse de données visuelles, extraites d'images.

Le modèle de sac de mots visuels est une des approches qui consiste à construire une description d'image aussi appelée signature d'image. La figure 1 illustre une partie du processus de représentation d'une image sous forme d'un sac de mots visuels. Cette approche a été proposée par Sivic en 2003 [J. Sivic 2003]. Elle s'appuie sur **le modèle de sac de mots** du domaine du traitement des documents textuels. Ce modèle original a été proposé en 1975 [Salton 1975]. La description des images par les sacs de mot visuels permet surtout, par cette signature, de les caractériser puis les classifier ou les regrouper les unes par rapport aux autres. L'objectif plus global est d'accomplir l'identification et la reconnaissance d'images. Lewis [Lewis 1992] et Wolf [Wolf 2005] montrent que le modèle de sac de mots dans le contexte de traitement de texte ne contient que 5% d'informations pertinentes pour la classification et/ou le regroupement de documents textuels. Nous nous sommes ainsi demandé ce qu'il en était pour les mots visuels, et en particulier s'il est possible d'en réduire la taille des sacs de mots visuels tout en conservant leur capacité de description. Dans leur version classique, les sacs de mots visuels d'images sont des vecteurs de fréquences des mots visuels. Ils se présentent sous forme tabulaire (voir la figure 2), chaque mot visuel étant un groupe (cluster) de caractéristiques. Ces caractéristiques sont les informations contenues dans l'image, par exemple la couleur, le contour, la forme, etc. Dans la base d'images, la comparaison entre images consiste alors

1. Laboratoire L3i, université de La Rochelle.

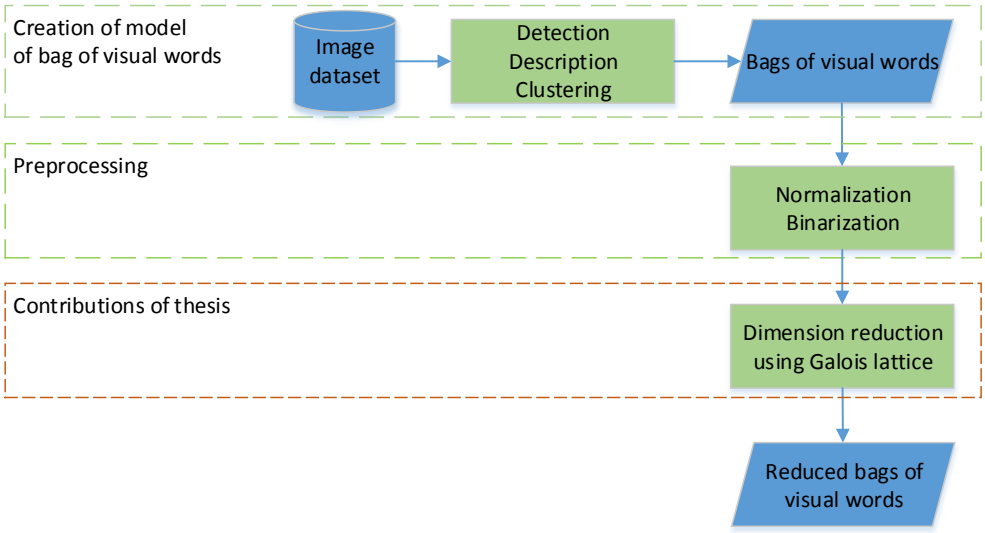


FIGURE 1: Illustration schématique présentant de manière générale la chaîne de traitement à partir d'une base d'images afin d'obtenir les sacs de mots visuels réduits en trois étapes : l'extraction de signature des images, le pré-traitement de données discrètes à données binaires, la réduction de dimension des signatures des images.

à comparer leurs vecteurs de mots visuels. La taille importante du vecteur de mots visuels rend les traitements difficiles lorsque le modèle de sac de mots visuels est appliqué aux grandes bases d'images que l'on rencontre de nos jours. Par exemple, la base VOC2012 contient 22500 images naturelles des différents objets. Chaque image peut être représentée par un vecteur de 3000 mots visuels [2]. Dans cette thèse, nous cherchons à réduire la taille de ces vecteurs en utilisant une méthode de réduction de dimension du vecteur de mots visuels.

De nombreux travaux sur **la réduction de dimension** existent dans plusieurs domaines scientifiques comme les statistiques, la vision par ordinateur ou l'apprentissage automatique. Elle peut être vue comme la réduction des informations redondantes et/ou non-pertinentes dans la description de données. De nombreuses méthodes de réduction de dimension sont proposées dans la littérature, en fonction du type de méthode (*i.e.* statistiques, probabilistes, logiques ou algébriques) mais aussi de l'approche utilisée pour le choix des attributs (*i.e.* extraction ou sélection), ou du type d'attributs analysés (*i.e.* attributs qualitatifs/symboliques ou attributs quantitatifs/numériques). Ces méthodes sont détaillées dans le chapitre 1 du manuscrit. En général, nous pouvons les classer en deux approches :

– **les méthodes d'extraction d'attributs** transforment l'ensemble d'attributs ini-

2. http ://host.robots.ox.ac.uk/pascal/VOC/voc2012/

		Mots visuels						
		a1	**a2**	**a3**	**a4**	**a5**	**a6**	**a7**
Images		8	1	0	2	0	1	7
		2	9	1	3	8	0	0
		2	0	1	9	0	1	0
		1	4	8	1	0	9	1

FIGURE 2: Illustration présentant de manière générale la représentation des images sous forme de sacs de mots visuels. Les images sont prises à partir de la base d'images CALTECH256 [Griffin 2007].

tiaux en un ensemble réduit de nouveaux attributs représentatifs.

– **les méthodes de sélection d'attributs** choisissent un sous-ensemble d'attributs initiaux.

Les termes mot visuel ou attribut sont utilisés de manière équivalente et sans distinction dans ce manuscrit. Les méthodes d'extraction d'attributs sont très efficaces en termes de réduction du nombre d'attributs. Cependant, Fukunaga montre que ces méthodes sont plutôt utiles pour compresser les données, mais peu adaptées pour catégoriser ou regrouper les données [Fukunaga 1990]. De plus, dans certains domaines, il est préférable de garder les attributs originaux pour leurs propriétés (*i.e.* dans le domaine de la bioinformatique, plus spécifiquement la sélection génétique liée à l'expression génomique). Les méthodes de sélection d'attributs ont donc été choisies pour traiter ce cas et ont été bien étudiées dans le cadre de l'apprentissage automatique [Ruck 1990, Battiti 1994, Kwak 2002, Peng 2005, Romero 2008, Bolón-Canedo 2014]. Cependant, elles sont souvent guidées par les performances expérimentales (*i.e.* la précision de la classification) et la vérité terrain [3]. En d'autres termes, ces méthodes sont couramment utilisées dans le cadre de l'apprentissage supervisé. Dans le cadre de l'apprentissage non supervisé, où la vérité terrain est manquante, la question de l'évaluation du résultat de la réduction se pose. La plupart des méthodes de réduction de dimension de la littérature s'appuient sur des statistiques, des probabilités, et très peu sur des mesures algébriques ou logiques.

3. La vérité terrain correspond au fait de savoir associer une étiquette à une donnée. En d'autres termes, l'information de la classe d'un élément dans les données est connue.

L'analyse formelle de concepts (AFC) [Ganter 1999] a été appliquée dans des domaines comme l'analyse de données ou la fouille de données. Ces applications ont popularisé la structure de treillis des concepts qui est une approche logique/algébrique. Le treillis des concepts se construit à partir d'un tableau de données binaires (images × mots visuels) appelé **contexte** (objets × attributs).

Un **treillis des concepts** est un graphe où chaque nœud est appelé **concept formel**, qui contient un ensemble maximal d'objets et leurs attributs communs. Deux concepts formels sont reliés par une relation d'inclusion entre objets et d'inclusion inverse entre attributs, appelée relation de spécialisation/généralisation. Dans le cas le plus pessimiste, ce graphe peut avoir une taille exponentielle. La montée rapide en puissance de calcul des ordinateurs ainsi que la démocratisation de ces puissances de nos jours nous permet de développer des applications utilisant ce type de graphes.

Le treillis des concepts a été utilisé dans le cadre de l'extraction des caractéristiques [Nguifo 1998] en générant les attributs numériques à partir des attributs binaires. Il a été appliqué aussi dans une étape d'une méthode de sélection d'attributs [Grissa 2016] [4]. Cependant, à notre connaissance au commencement de cette thèse, il n'a jamais été appliqué comme une méthode de sélection des attributs. Pourtant, il semble posséder des propriétés intéressantes pour son exploitation dans ce domaine. En 2014, Ikeda et Yamamoto [Ikeda 2014] ont appliqué l'Analyse Formelle de Concepts pour la prédiction des classes dans la base de test simultanément à la sélection des attributs locaux.

La structure de treillis a fait l'objet d'une recherche théorique abondante dès la fin du 19ème siècle. Le premier ouvrage de référence sur la théorie des treillis en donne une définition algébrique à partir des opérations de borne inférieure et de borne supérieure dans le livre de Birkhoff de 1940 [Birkhoff 1940]. Ensuite, en 1970, Barbut et Monjardet introduisent le terme de **treillis de Galois** [5] en le décrivant de manière structurelle sous la forme d'un graphe [Barbut 1970]. Dans ce graphe, l'auteur fait la distinction entre les éléments ne correspondant pas à une borne supérieure (sup-irréductible) ou une borne inférieure (inf-irréductible). Les éléments irréductibles sont rassemblés sous forme d'une table binaire, la table des irréductibles, les sup-irréductibles en lignes et les inf-irréductibles en colonnes. Dans leur livre, Barbut et Monjardet présentent le résultat fondamental de la théorie des treillis qui établit que tout treillis fini est isomorphe au treillis de Galois de sa table des irréductibles [Barbut 1970]. Cette table décrit la structure du treillis et permet sa reconstruction. Elle est minimale pour cette propriété. Cette propriété permet de réduire le nombre d'attributs en gardant les informations pertinentes pour distinguer les objets au sein des concepts.

La structure du treillis et les combinaisons objets/attributs sont maintenues en ne conservant que les attributs correspondant aux inf-irréductibles. Il s'agit d'une réduction

4. Il a été appliqué dans la stratégie de recherche des sous-ensembles possibles de l'approche par encapsulation pour sélectionner un sous-ensemble des attributs.

5. Le treillis de Galois est aussi appelé treillis des concepts.

logique. Dans ce manuscrit, nous proposons un algorithme de réduction qui repose sur ce principe de sélection des attributs irréductibles. Dans ce manuscrit, nous nous intéressons à la réduction d'attributs sur le modèle de sac de mots visuels des bases d'images.

Objectifs

L'objectif de cette thèse est de proposer et d'évaluer une méthode de réduction de dimension pour réduire le nombre des attributs (mots visuels) tout en gardant leur capacité de distinguer les images. Pour cela, nous nous focaliserons sur :

- *la réduction des attributs en gardant les inf-irréductibles du treillis*, qui repose sur le théorème fondamental de la théorie des treillis [Barbut 1970], et qui utilise le graphe de précédence pour réaliser la réduction. Cet algorithme est expérimenté sur plusieurs bases de données d'images ;
- *la réduction des attributs en supprimant les attributs irréductibles "similaires"*, qui repose sur une extension approximative du graphe de précédence, utilisé de façon similaire à l'algorithme précédent. Cet algorithme de réduction floue réduit les attributs en acceptant une baisse de performances par rapport à l'algorithme de réduction précédent, en expérimentant cet algorithme sur une base du domaine de sélection des attributs.

Organisation de la thèse

Ce manuscrit comprend deux parties distinctes, intitulées "État de l'art" et "Réduction de dimension".

La première partie "État de l'art" sert à introduire de manière générale le cadre de cette thèse et le domaine de recherche de la réduction de dimension.

Le chapitre 1 présente, au travers d'une étude bibliographique, **les méthodes de réduction de dimension**. Nous présentons d'abord les différents types de données en positionnant les données binaires que nous traitons dans cette thèse. Nous détaillons les méthodes pertinentes de la littérature. Nous concluons par un tableau comparatif de ces méthodes en fonction de leur type (*i.e.* statistique ou logique), du type de données d'entrées (*i.e.* continue ou discrète), du type d'apprentissage (*i.e.* supervisé ou non) et du type de réduction (*i.e.* sélection ou extraction).

Le chapitre 2 porte sur le formalisme et les notions relatives à l'AFC utiles à

la compréhension de ce manuscrit. Tout d'abord, nous rappelons la base de la théorie des treillis à travers leurs définitions, plus particulièrement celles des treillis des concepts, aussi appelés treillis de Galois. Nous introduisons les éléments irréductibles et **le théorème de Barbut** [Barbut 1970] qui établit que tout treillis fini est isomorphe au treillis de Galois de sa table des irréductibles. Ce théorème prouve que la structure du treillis ne change pas en ne gardant que les attributs "irréductibles" du contexte. Nous introduisons aussi les sous-hiérarchies de Galois (l'AOC-poset et l'AC-poset) qui sont les sous-ordres du treillis de Galois. Enfin, nous introduisons la notion de système de fermeture et son treillis des fermés qui nous permettent d'étendre l'utilisation de l'algorithme RedAttsSansPerte d'un contexte à un système de fermeture quelconque.

Le chapitre 3 présente le modèle de sac de mots visuels. Nous commençons par présenter la chaîne de traitement classique pour obtenir les sacs de mots visuels à partir des images. Puis nous détaillons les phases de construction que sont l'extraction des attributs, la construction du dictionnaire de la base et l'encodage du sac de mots visuels pour chaque image. Enfin, nous présentons les algorithmes que nous avons utilisés dans cette thèse pour obtenir les sacs de mots visuels.

La seconde partie "Réduction de dimension" présente nos différentes contributions.

Le chapitre 4 est une présentation des algorithmes de réduction. Tout d'abord, nous donnons la définition du graphe de précédence où deux attributs sont en relation si les ensembles d'objets qui les contiennent sont inclus l'un dans l'autre. Ce graphe d'attributs est proche de l'AC-poset. Puis, nous présentons notre algorithme de réduction d'attributs, **RedAttsSansPerte**, qui ne garde que les attributs qui correspondent à des inf-irréductibles du treillis des concepts. Cet algorithme utilise le graphe de précédence et comprend trois étapes : clarification, standardisation et réduction. Puis, nous proposons l'algorithme de réduction, **RedAttsFloue**, qui repose sur le même principe, mais en utilisant une extension floue du graphe de précédence. Cet algorithme est une extension de l'algorithme RedAttsSansPerte. Il s'agit de supprimer les attributs irréductibles "similaires" tout en conservant une grande partie des informations pertinentes pour la classification selon un "seuil de flexibilité" qui permet de construire le graphe de précédence flou. Les propriétés de ce graphe permettent un traitement en cinq étapes : clarification, standardisation, réduction, réduction floue et clarification floue.

Le chapitre 5 présente les protocoles expérimentaux mis en place pour évaluer quantitativement et qualitativement la capacité de réduction des deux algorithmes RedAttsSansPerte et RedAttsFloue. Nous analysons le comportement de nos algorithmes

sur différents ensembles de données : des bases de données d'images et une base du domaine de sélection des attributs. Les expérimentations montrent que la réduction par l'algorithme RedAttsSansPerte, permet de **diminuer la taille de l'ensemble des attributs tout en conservant la performance** de classification. Nous montrons par des expérimentations qu'avec un bon seuil de flexibilité, **l'algorithme RedAttsFloue réduit davantage d'attributs que l'algorithme RedAttsSansPerte** tout en gardant de bonnes performances de classification (*i.e.* F-mesure). Cependant, un seuil de flexibilité plus élevé entraine mécaniquement une perte d'information et par la même l'occasion une baisse possible de performance de la classification.

En conclusion, nous rappelons les objectifs et les solutions proposées, ainsi que les perspectives d'amélioration de l'algorithme de réduction des attributs s'appuyant sur la théorie de treillis.

État de l'art

« Compliments of predecessors are always easy to hear, but their critiques are really helpful. » "

(Vladimir Tendryakov)

Chapitre 1

Etat de l'art sur la réduction de dimension

Sommaire

1.1 Introduction 11
1.2 Types des données 13
1.3 Approche par extraction d'attributs 14
1.3.1 Application au cas de l'apprentissage supervisé 15
1.3.2 Application au cas de l'apprentissage non supervisé 16
1.4 Approche par sélection d'attributs 20
1.4.1 Application au cas de l'apprentissage supervisé 21
1.4.2 Application au cas de l'apprentissage non supervisé 34
1.5 Conclusion 43
Points clés 46

1.1 Introduction

L'analyse de données est un vaste domaine dont l'un des buts est de chercher à donner un sens à des données. Il est possible de représenter des données par des vecteurs d'attributs, où le nombre d'éléments composant un vecteur est appelé **dimension** du vecteur. Lors du traitement de données de grandes dimensions, plusieurs raisons incitent à réduire la dimension de ces données. Par exemple, la réduction de dimension peut potentiellement augmenter la précision de la classification/regroupement, améliorer la visualisation et la

compréhension des données, diminuer les temps de calcul et le stockage. Elle permet aussi d'éviter le problème "Large p Small n" [West 2003] : "Large p Small n" est un défi connu dans le domaine statistique où certains algorithmes statistiques ne peuvent pas fonctionner correctement quand le nombre d'attributs p est trop grand par rapport au nombre d'objets n. Néanmoins, au cours des vingt dernières années, les procédures ont été développées ou adaptées pour fournir des résultats pratiques pour répondre à ce défi. Les auteurs de [Johnstone 2009] fournissent un article de synthèse sur ce sujet.

En fonction du domaine d'application, l'objectif de la réduction de dimension n'est pas le même mais cette étape est communément nécessaire dans la mise en place de l'application. En effet, dans la communauté statistique, les approches visent à réduire la dimension de telle sorte que la représentation soit aussi fidèle que possible aux données originales. En apprentissage automatique, en reconnaissance de motifs ou en bioinformatique, la réduction de dimension est utilisée comme une étape de pré-traitement des données afin d'augmenter la compréhension des données et les performances de classification ou de clustering.

Les méthodes peuvent se distinguer de deux manières :
- le moyen de faire la réduction (i.e. la nature des méthodes permet leur utilisation dans la sélection ou l'extraction d'attributs.) ;
- l'application à un type d'apprentissage particulier (i.e. la nature des méthodes permet d'application dans le cas supervisé ou le cas non-supervisé.).

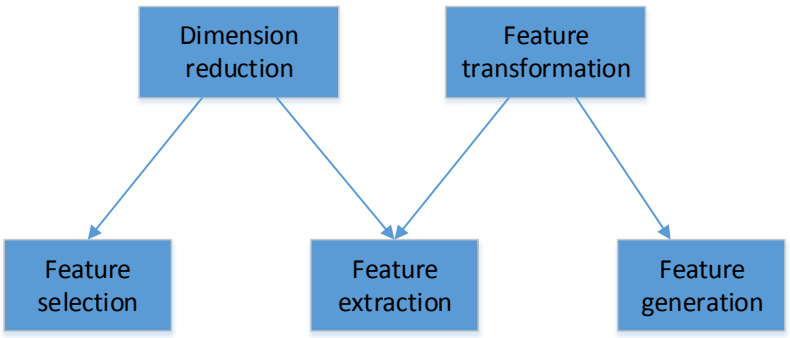


FIGURE 1.1: Illustration schématique présentant de manière générale la réduction de dimension.

Les approches de **réduction de dimension** peuvent se différencier selon qu'on cherche une **extraction** des attributs ou une **sélection**.

Extraction : On parle d'**extraction des attributs** lorsque l'ensemble des attributs transformés est plus petit que l'ensemble initial d'attributs. L'extraction est une sous-catégorie de la transformation. L'idée principale des méthodes de **transfor-**

mation est de transformer l'ensemble initial des attributs en un nouvel ensemble d'attributs. Ce nouvel ensemble d'attributs conservant au mieux l'information originale. Si ce nouvel ensemble d'attributs est plus grand que l'ensemble original, la méthode est appelée méthode de **génération des attributs**.

Sélection : Les méthodes de **sélection des attributs** se proposent de choisir un sous-ensemble d'attributs à partir de l'ensemble original de telle façon que ce sous-ensemble contienne les informations essentielles pour représenter les objets. Dans le domaine de l'apprentissage automatique, l'objectif est de choisir un sous-ensemble d'attributs pour catégoriser/regrouper les objets. L'approche de sélection des attributs est préférable dans les domaines où les attributs originels (non transformées) sont nécessaires afin de maintenir les propriétés physiques des attributs.

Dans la communauté de l'apprentissage automatique, ces méthodes de **réduction de dimension** peuvent aussi se répartir en deux catégories : celles qui s'appliquent dans le cas de l'apprentissage **supervisé** et celles qui s'appliquent dans le cas de l'apprentissage **non supervisé**.

Supervisé : L'apprentissage supervisé a pour objectif de catégoriser des objets dans des classes où le nombre de classes et la classe de chaque objet est connue *a priori*.

Non supervisé : L'objectif de **l'apprentissage non supervisé** est de regrouper des objets dans des clusters selon leur similarité de telle façon que les objets d'un cluster donné se ressemblent plus que les objets de clusters différents.

Au cours de ces dernières décennies, l'étude des méthodes d'extraction d'attributs en général a énormément progressé [Hotelling 1933, Fisher 1936, Friedman 1974]. L'approche de la sélection d'attributs dans le cas de l'apprentissage supervisé est bien étudiée aussi [Quinlan 1986, Kass 1980, Hall 1997]. Par contre, dans le cas non supervisé, il reste encore de grands enjeux malgré de multiple propositions [Dy 2004, Hong 2008, Li 2008].

Ce chapitre commence par un rappel sur les différents types de données. Nous exposerons ensuite des travaux récents sur la réduction de dimension qui se distinguent par les deux critères présentés en amont.

1.2 Types des données

Dans cette section, nous nous intéresserons aux différents types de données de base. Elles peuvent être **qualitatives**, **quantitatives** et **textuelles** (dans certains cas particuliers). Les données qualitatives (ou catégorielles) peuvent être **nominales** ou **ordonnées**. Les données quantitatives (ou numériques) peuvent être **continues** ou **discrètes**.

Les données **qualitatives** sont des données dont l'ensemble des valeurs est fini. Si elles ne sont pas ordonnées (nous ne pouvons pas les comparer par une relation d'ordre $\geqslant$. Exemple : homme, femme.) alors elles sont dites nominales. Les données ordinales peuvent être rangées dans la famille des données discrètes et traitées comme elles (*e.g.* vieux, jeune, adolescence).

Les données **quantitatives** sont des données sur lesquelles nous pouvons effectuer des opérations arithmétiques. De plus elles sont ordonnées. Les données **continues** sont les données dont l'ensemble des valeurs est un sous-ensemble infini de l'ensemble $\mathcal{R}$ des réels (*e.g.* le salaire). Les données **discrètes** sont des données dont l'ensemble des valeurs est un sous-ensemble fini ou infini de l'ensemble $\mathcal{N}$ des entiers naturels (*e.g.* le nombre de chiens).

Les données **textuelles** sont des données qui sont écrites en langage naturel telle une lettre, un résumé, etc.

En informatique, il existe aussi les données binaires. Les données binaires sont des données qui ne peuvent prendre que deux valeurs (*e.g.* 0 ou 1, aou b, oui ou non, O ou N, etc.). C'est un cas particulier de données discrètes.

Dans le cas de données non binaires (i.e. données qualitatives, données textuelles, données continues), un pré-traitement permet d'obtenir des données binaires. La transformation de données non binaires en données binaires entraine un certain nombre de conséquences. Par exemple, pour des données quantitatives, le processus obligatoire de transcription des modalités d'un attribut en plusieurs attributs (chaque modalité devient un attribut) augmente le nombre final d'attributs. Avec des données numériques (continues ou discrètes), il est possible de construire des intervalles, disjoints ou non, qui deviennent des attributs. Par conséquent, cette action mène à une perte d'informations.

1.3 Approche par extraction d'attributs

Nous présentons dans cette section quelques méthodes connues d'extraction des attributs. Nous les distinguons naturellement par l'application dans l'apprentissage supervisé ou l'apprentissage non supervisé.

En général, les méthodes qui peuvent s'appliquer dans le cas non supervisé peuvent s'appliquer directement dans le cas supervisé sans aucun problème. Cependant, l'opération inverse n'est pas toujours possible. En d'autres termes, l'apprentissage supervisé fournit

une vérité terrain [1] tandis que l'apprentissage non supervisé n'a pas cette information. Par conséquent, les méthodes qui nécessitent la présence de la vérité terrain peuvent s'appliquer immédiatement dans l'apprentissage supervisé mais ne peuvent pas être appliquée entièrement dans le cas non supervisé. Les modifications de ces méthodes sont nécessaires voire parfois impossible.

Par exemple, la méthode d'Analyse en Composantes Principales (ACP) est appliquée dans le cadre de l'apprentissage non supervisé et la méthode d'Analyse Discriminante Linéaire (ADL) est appliquée dans le cadre de l'apprentissage supervisé. Ces deux méthodes seront présentées en aval dans la sous-section 1.3.2 et la sous-section 1.3.1 respectivement. En général, l'ACP ne considère que la structure globale des données, tandis que l'ADL utilise la vérité terrain (i.e. elle maximise la distance entre les classes à l'aide des informations de classes de la vérité terrain.). Par conséquent, l'ADL ne peut pas fonctionner correctement sans la vérité terrain.

Le lecteur qui ne serait pas familier avec les notions abordées dans les sous-sections suivante peut passer à l'annexe C où sont abordées certaines notions du domaine des statistiques et des mathématiques telles que l'indépendance statistique, la distribution gaussienne, etc.

1.3.1 Application au cas de l'apprentissage supervisé

Les méthodes d'extraction d'attributs qui ne fonctionnent que dans le cas d'apprentissage supervisé sont assez rare. Nous présentons ci-dessous une méthode qui n'est applicable que dans le cas d'apprentissage supervisé.

Analyse Discriminante Linéaire (ADL) - Linear Discriminant Analysis (LDA)

L'ADL [Fisher 1936] est une généralisation du discriminant linéaire de Fisher. Cette méthode est utilisée dans les domaines des statistiques, de la reconnaissance de formes et de l'apprentissage automatique (machine learning). L'ADL est une méthode supervisée linéaire de transformation d'attributs. Elle tient compte des étiquettes de classes. La génération de l'ADL pour un problème de classification à k classes est la projection de n dimensions à $(k-1)$ dimensions (n est le nombre des dimensions originales, $n \gg k$). L'idée principale est de découvrir une transformation qui maximise la séparation entre les

1. La vérité terrain correspond aux données avec des étiquettes. Autrement dit, l'information de classe d'un objet est connue.

classes et minimise la séparation entre les éléments d'une classe :

$$argmax(\frac{w^T S_B w}{w^T S_w w}) \tag{1.3.1}$$

où,
- $S_w = \sum_{i=1}^{k} S_i$ la somme des matrices de dispersion des k classes.
- $S_i = \sum_{x \in C_i} (x - m_i)(x - m_i)^T$ la matrice [2] de dispersion de la classe i.
- $S_B = \frac{1}{k} \sum_{i=1}^{k} (m_i - m)(m_i - m)^T$ la matrice de dispersion entre k classes.
- $m_i = \frac{1}{N_i} \sum_{x \in C_i} x$ la moyenne de la classe C_i.
- m est la moyenne globale.

La solution est donc constituée des (k-1) vecteurs propres de $S_w^{-1} S_B$.

Les variantes de l'ADL prennent en compte des transformations non linéaires utilisant l'astuce du noyau [Schölkopf 2002]. Nous pouvons citer par exemple les approches : Local Fisher Discriminant Analysis (LFDA) [Sugiyama 2006] ou Generalized Discriminant Analysis (GDA) [Baudat 2000].

L'ADL est liée à l'Analyse en Composantes Principales (ACP) en ce qu'elle cherche des combinaisons linéaires des attributs qui expliquent le mieux les données [Martinez 2001]. Cependant, l'ADL tente explicitement de modéliser la différence entre les classes de données. D'autre part, l'ACP ne prend pas en compte la différence entre les classes, elle est utilisée dans le cas non supervisé.

1.3.2 Application au cas de l'apprentissage non supervisé

Nous présentons ci-dessous quelques méthodes d'extraction d'attributs dont leur nature permet de les appliquer dans le cas d'apprentissage non supervisé.

1.3.2.1 Analyse en Composantes Principales (ACP)

L'ACP [Hotelling 1933] peut être utilisé comme une méthode de réduction de dimension linéaire. L'ACP a été inventée en 1901 par Karl Pearson comme une alternative de la théorème de l'axe principal dans la mécanique [Pearson 1901]. En 1933 Harold Hotelling a développé indépendamment ce concept et l'a nommé analyse en composantes principales. Dans plusieurs domaines, l'ACP est également connue sous divers noms : Analyse Sémantique Latente (ASL), Décomposition en Valeurs Singulières (DVS), décomposition

2. i.e. within class scatter matrix.

spectrale, transformation de Karhunen–Loève (KLT), transformation de Hotelling, décomposition orthogonale aux valeurs propres (Empirical Orthogonal Function - EOF). L'ACP cherche les composantes indépendantes expliquant au mieux la variance des données. Les composantes principales sont les combinaisons linéaires orthogonales des attributs d'origine ayant les plus grandes variances.

En supposant que les données contiennent n observations décrites par p attributs. Nous pouvons les représenter par une matrice $X_{p\times n}$ où chaque colonne représente une observation.

L'idée principale de l'ACP tourne autour de la matrice de variance-covariance à p lignes et p colonnes

$$\Sigma = \frac{1}{n} X X^T \tag{1.3.2}$$

où la diagonale de la matrice Σ représente la variance de chaque attribut, et le reste de la matrice Σ représente la covariance entre les paires d'attributs correspondantes. X est la matrice de p observations et n attributs normalisés dont la moyenne de chaque attribut est égale à zéro et l'écart-type de cet attribut est égal à 1.

Selon le théorème de décomposition d'une matrice en éléments propres, la matrice de covariance Σ de la taille $p \times p$ peut être décomposée sous la forme :

$$\Sigma = W \Lambda W^{-1} \tag{1.3.3}$$

où Λ est la matrice diagonale des valeurs propres ordonnées $\lambda_1 \lneq ... \lneq \lambda_p$,

W est la matrice orthogonale dont la $j^{ème}$ colonne est le vecteur propre qui correspond à la matrice Σ,

et W^{-1} est la matrice inverse de la matrice W.

Mardia et al. [Mardia 1979] montrent que les composantes principales sont les lignes de la matrice $S_{p\times n}$ où

$$S_{p\times n} = W^{-1} X \tag{1.3.4}$$

Les auteurs de [Mardia 1979] prouvent que les k premiers vecteurs propres donnent la plus petite déviation moyenne quadratique (mean square deviation) de X [3] parmi tous les sous-espaces de dimension k [4].

Dans certains cas, les extensions de l'ACP à des cas non linéaires sont intéressantes comme l'ACP avec noyau [Schölkopf 1997] qui s'appuie sur l'astuce du noyau [Schölkopf 2002]

3. La taille de X est $p \times n$.
4. k est le nombre des composantes principales retenues.

ou réseau de l'ACP non linéaire [Fodor 2002] qui extrait les composantes principales à partir d'un réseau de neurones à cinq couches. Une autre méthode d'ordre supérieur qui cherche les projections linéaires, ne nécessitant pas qu'elles soient orthogonales entre elles, est l'Analyse en Composantes Indépendantes (ACI) [Comon 1994]. L'ACI est proche de la notion d'ACP. En effet, la première cherche des attributs indépendants tandis que la seconde recherche les attributs non corrélés.

L'ACP est aussi utilisée comme une méthode de sélection des attributs dans [Mardia 1979].

1.3.2.2 Poursuite de Projection (PP)

La méthode Poursuite de Projection (PP) est une méthode de transformation linéaire [Friedman 1974]. L'objectif de cette méthode est de chercher les projections "intéressantes" afin d'optimiser un indice de projection. Cette méthode peut servir à l'estimation de la densité, la régression, plus spécifiquement la réduction de dimension quand le but est la visualiser de données. Les projections déterminées permettent de dévoiler la plupart des détails de la structure de données (*i.e.* au sens de l'indice précédemment mentionné). La recherche d'une nouvelle projection se fait de manière itérative. Pour chaque projection trouvée, les données sont réduites en supprimant les composantes suivant cette projection. Puis l'étape de recherche d'une projection est répétée avec le résidu.

Le problème théorique principal de cette méthode, est la définition de l'indice de projection servant à définir "l'intérêt" d'une direction. Cet indice peut être une mesure non-gaussienne [5], ou une mesure gaussienne [6]. Si une mesure gaussienne est utilisée comme indice, sous réserve que les projections soient orthogonales, la méthode PP est similaire à la méthode ACP.

La première mise en œuvre réussie de cette méthode avec pour objectif le regroupement de données est présentée dans le travail de Friedman et Tukey [Friedman 1974]. Plusieurs scientifiques ont proposé des indices de projection possibles : des mesures s'appuyant sur l'entropie différentielle [Huber 1985, Jones 1987], des mesures s'appuyant sur la norme L2 pondérée entre la densité de données projetées et la densité de la distribution gaussienne [Friedman 1987, Cook 1993], des mesures s'appuyant sur l'approximation de l'entropie [Hyvärinen 1997]. Ces indices étant le résultat d'une fonction d'ordre supérieure, ils permettent à la méthode de Poursuite de Projection d'exploiter les données dans le cas d'une distribution non gaussienne. Cependant, cette méthode demande un temps de calcul élevé.

5. *i.e.* c'est une mesure d'ordre supérieur au deuxième ordre d'information. Cette mesure est utilisée pour les données avec une distribution non gaussienne.

6. Une mesure gaussienne détermine l'information de deuxième ordre. Elle est utilisée pour les données ayant une distribution normale, aussi appelée distribution gaussienne.

1.3.2.3 Analyse en Composantes Indépendantes (ACI)

L'Analyse en Composantes Indépendantes est une méthode de transformation linéaire ou non-linéaire. L'idée générale de l'ACI est de considérer pour un vecteur aléatoire x, la recherche des projections linéaires $s = Wx$ telles que les composantes s_i soient aussi près que possible de l'indépendance statistique, au sens de la maximisation d'une fonction $f(s_1, ..., s_m)$ qui mesure l'indépendance. Ces projections ne nécessitant pas d'être orthogonales entre elles. Le premier but de l'ACI n'est pas la réduction de dimension. En effet, il existe des versions de l'ACI où le nombre de composantes indépendantes est plus grand que le nombre de variables initiales [Hyvarinen 1999]. Par exemple, les auteurs de [Olshausen 1996, Bell 1997, Oja 2014] utilisent l'ACI pour extraire les attributs à partir des images naturelles.

L'ACI peut être considérée comme une généralisation du concept de l'ACP et du PP.

En effet, l'ACI cherche des variables indépendantes statistiquement or l'ACP cherche les variables non-corrélées. L'indépendance statistique implique [7] la non-corrélation, mais l'inverse n'est pas vérifié. Autrement dit, la condition de l'indépendance statistique est plus stricte que la condition de la non-corrélation. Cependant, il existe un cas particulier où l'indépendance statistique et la non-corrélation sont équivalents. C'est quand les variables aléatoires $y_1, y_2, ..., y_m$ ont une distribution jointe de gaussienne (cf. [Comon 1994]). Dans ce cas, l'ACI peut être considérée comme équivalente à l'ACP.

Par ailleurs, le modèle de l'ACI non bruitée définit l'ACI d'un vecteur aléatoire de m-dimensions x comme l'estimation des composantes s_i d'un vecteur $s = (s_1, ..., s_k)^T$ et d'une matrice A de la taille $m \times k$ tel que $x = As$. Les composantes s_i sont aussi indépendantes que possibles, selon la définition de l'indépendance. Avec cette définition, l'ACI non bruitée est similaire à PP, dont la définition de l'indice de projection correspond à la définition de l'indépendance dans l'ACI.

1.3.2.4 Analyse en facteurs booléens (Boolean Factor Analysis)

La méthode d'analyse en facteurs booléens (AFB) est la décomposition / factorisation d'une matrice booléenne M de taille $n \times m$ en deux matrices booléennes F et B de taille $n \times k$ et $k \times m$ respectivement telles que la taille de la factorisation k est minimisée et telles que $M = F \circ B$, où $\circ$ est le produit booléen des matrices binaires défini par :

$$(F \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \wedge B_{lj}, \tag{1.3.5}$$

7. Voir l'annexe C.4.

où $\bigvee$ est la disjonction logique et $\wedge$ est la conjonction logique.

La décomposition booléenne de la matrice M en deux matrices F et B correspond à la découverte d'un facteur minimisé de taille k servant à expliquer les données qui sont représentées par M. Ce facteur qui est appelé le facteur optimal est le but de la réduction de dimension. Le problème de la décomposition booléenne est NP-complet, ainsi pour l'approximation $M \approx F \circ B$, c'est pourquoi plusieurs heuristiques ont été proposées pour minimiser la recherche du facteur optimal booléen de taille k [Koyutürk 2003, Lu 2008, Miettinen 2010, Belohlavek 2010, Lu 2011]. Par exemple, l'algorithme proposé par [Belohlavek 2010] utilise les concepts formels [8] comme les facteurs pour l'extraction d'attributs à partir de données binaires dans le cas de l'apprentissage non supervisé. C'est une des méthodes qui s'appuient sur l'approche logique/algébrique et non sur l'approche statistique.

1.4 Approche par sélection d'attributs

Dans le domaine de l'apprentissage automatique, les méthodes de sélection d'attributs n'ont pas été utilisées uniquement pour réduire le nombre d'attributs mais aussi pour mieux comprendre la structure de données afin d'améliorer la catégorisation (classification) ou le regroupement (clustering) des données. Dans la présentation, nous les distinguons par leur nature d'application dans le cas de l'apprentissage supervisé (pour la classification) ou dans le cas de l'apprentissage non supervisé (pour le regroupement).

Dans le cas supervisé, certaines méthodes de sélection des attributs sont guidées par la vérité terrain. Cela permet d'avoir un objectif précis et d'attendre de bons résultats [Ruck 1990, Battiti 1994, Kwak 2002, Peng 2005, Romero 2008, Bolón-Canedo 2014]. Les méthodes de sélection des attributs connues dans le cas supervisé sont catégorisées et présentées dans la section 1.4.1.

Dans le cas non supervisé, la sélection des attributs a pour objectif de trouver un sous-ensemble d'attributs sans l'aide de la vérité terrain. Ce sous-ensemble doit grouper au "mieux" les objets similaires. Cependant, la définition de "mieux" est floue sans information complémentaire.

Par exemple, si nous utilisons le critère : **les performances de regroupement** de l'ensemble d'attributs original et d'un sous-ensemble d'attributs sélectionnés sont identiques, alors ce critère est flou sans vérité terrain. En effet, les performances de regroupement

8. La définition d'un concept formel d'un treillis des concepts est présentée en section 2.2.1 du chapitre 2.

de l'ensemble d'attributs original peuvent ne pas être les meilleures à cause du bruit qui existe dans la représentation et perturbe l'algorithme de regroupement. Nous devons par conséquent choisir un sous-ensemble d'attributs dont les performances de regroupement sont supérieures ou égales à l'ensemble des attributs initiaux. Cependant, il peut y avoir plusieurs sous-ensembles d'attributs ayant ces propriétés en fonction de l'algorithme de regroupement utilisé. Dans ce cas, quel sous-ensemble faut-il choisir ?

Ou bien nous utilisons le critère : "le sous-ensemble d'attributs choisi doit *minimiser la similarité* entre les objets dans le même groupe et **maximiser la différence** entre les objets dans des groupes différents". Le défi avec ce critère est le même : différents sous-ensembles d'attributs choisis peuvent mener à différents regroupements qui satisfont cette condition. Par conséquent, lequel choisir ?

Ces questions ont été soulevées et étudiées dans plusieurs travaux [Dy 2004, Wolf 2005, Chandrashekar 2014]. Nous présenterons les méthodes qui essaient d'y répondre dans la section 1.4.2.

Auparavant, nous présentons quelques méthodes appliquées dans le cas d'apprentissage supervisé dans la section 1.4.1.

1.4.1 Application au cas de l'apprentissage supervisé

Dans le cas supervisé, en fonction de la relation entre le critère de sélection d'attributs et l'algorithme d'apprentissage, nous pouvons distinguer trois approches : le filtrage (filter), l'encapsulation (wrapper) et l'intégration (embedded).

L'approche par filtrage choisit les attributs pertinents indépendamment de l'algorithme d'apprentissage. En revanche, **la technique par encapsulation** profite de l'efficacité de l'algorithme d'apprentissage. En effet, la fonction d'évaluation de l'approche par filtrage est habituellement la mesure de données intrinsèques, tandis que l'approche par encapsulation utilise le taux d'erreur de la classification ou la performance de la classification pour évaluer l'efficacité des attributs.

L'indépendance de l'approche par filtrage par rapport à l'algorithme d'apprentissage est aussi son inconvénient. En effet, la **pertinence** des attributs est communément définie à partir de la performance de la classification de l'algorithme d'apprentissage. Dans l'apprentissage supervisé, cette performance est fiable grâce à la vérité terrain.

Le choix de l'approche par encapsulation est motivé par la disparition de l'inconvénient rencontré dans l'approche par filtrage. Cependant, l'approche par encapsulation est criti-

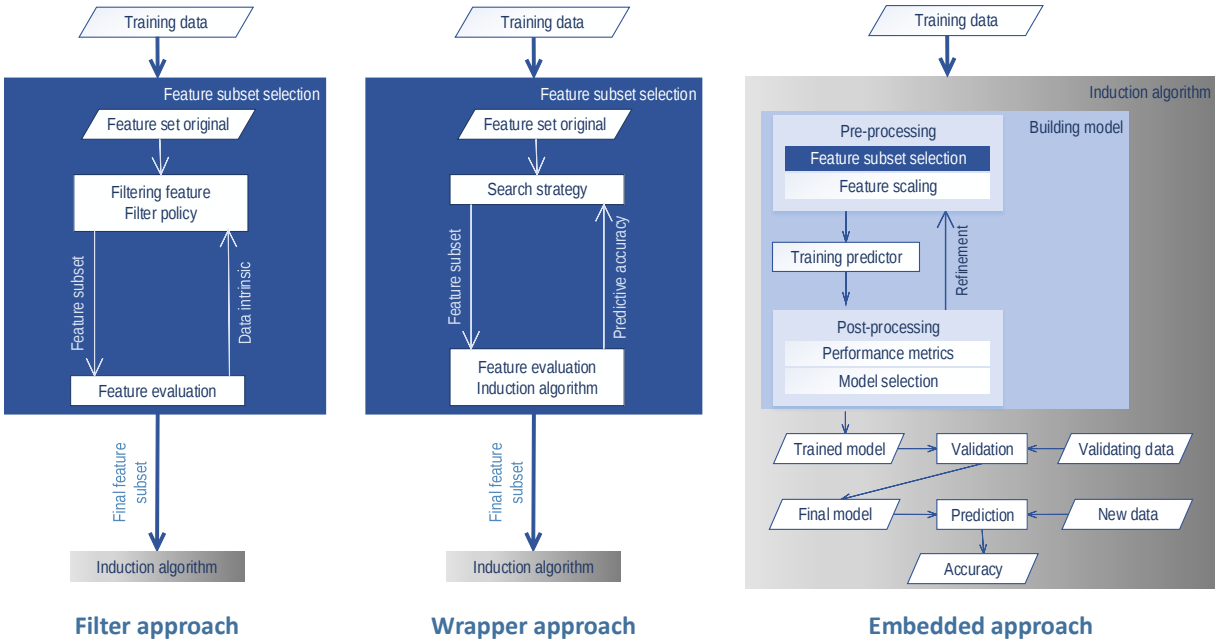


FIGURE 1.2: Les trois approches de sélection des attributs : approche par filtrage, approche par encapsulation et approche par intégration.

quée du fait de son important temps de calcul : elle doit chercher la meilleure solution dans un espace exponentiel [9]. De plus, elle peut potentiellement entraîner du surapprentissage [10] comme le montre la publication de [Loughrey 2004].

Afin de réduire le temps de calcul nécessaire à la re-classification [11] des différents sous-ensembles d'attributs dans l'approche par encapsulation, **l'approche par intégration** incorpore le processus de sélection des attributs dans la construction du modèle de classification. Les méthodes qui combinent l'approche par filtrage et l'approche par encapsulation sont aussi appelées approches par intégration étant données les améliorations prometteuses en termes d'erreur en classification.

La figure 1.2 illustre le fonctionnement de ces trois approches. Nous présentons dans les sous-sections en aval quelques méthodes de sélection des attributs pour chacune de ces trois approches.

9. Tester tous les sous-ensembles d'attributs possibles afin de trouver le meilleur sous-ensemble d'attributs.

10. Une méthode est dite en surapprentissage quand elle perd son pouvoir de prédiction sur de nouveaux échantillons du fait d'avoir trop appris sur l'ensemble de données. Cette notion s'appelle **overfitting** en anglais.

11. Classifier à nouveau.

1.4.1.1 Approche par filtrage

Le concept principal de l'approche par filtrage est de choisir les attributs les plus pertinents en utilisant deux paramètres : un ou plusieurs critères de filtrage [12] combinés à une politique de filtrage.

1.4.1.1.1 Critères de filtrage

Dans les paragraphes suivants, nous montrons quelques filtrages de mesures populaires. Ces filtrages de mesures utilisent le tableau de contingence, en considérant les attributs indépendants entre eux. Un tableau de contingence peut être construit à partir d'une classe c et d'un attribut f comme présenté dans le tableau 1.1.

	c_P	c_N	
v_1	$n_{1P}\,(\mu_{1P})$	$n_{1N}\,(\mu_{1N})$	n_1
...	...	...	
v_i	$n_{iP}\,(\mu_{iP})$	$n_{iN}\,(\mu_{iN})$	n_i
...	...	...	
v_r	$n_{rP}\,(\mu_{rP})$	$n_{rN}\,(\mu_{rN})$	n_r
	n_P	n_N	n

TABLE 1.1: Le tableau de contingence permettant d'obtenir la dépendance entre la classe c $\{c_P, c_N\}$ et l'attribut f $\{v_1,.., v_r\}$.

Ce tableau présente la relation entre la classe c et l'attribut f. Les valeurs possibles de la classe c sont positive c_P ou négative c_N. Les valeurs possibles de l'attribut f sont v_1, v_2, ..., v_r.

n_{ij} est le nombre d'occurrences de l'attribut f valant v_i pour la classe c valant c_j (où $j = \{P, N\}$).

$n_i = n_{iP} + n_{iN}$ est le nombre total d'occurrences de l'attribut f valant v_i dans la classe c.

$n_P = \sum_{i=1}^{r} n_{iP}$ est le nombre total d'occurrences de l'attribut f dans la classe c valant c_P.

12. En d'autres termes : des stratégies de filtrage.

$n_N = \sum_{i=1}^{r} n_{iN}$ est le nombre total d'occurrences de l'attribut f dans la classe c valant c_N.

$n = n_P + n_N$ est le nombre total d'occurrences de l'attribut f dans la classe c. En d'autres termes, n est le nombre total d'occurrences traités dans le tableau.

$\mu_{iP} = \frac{n_i n_P}{n}$ est la valeur attendue dans le cas où les données ont été uniformément distribuées.

$\mu_{iN} = \frac{n_i n_N}{n}$ est la valeur attendue dans le cas où les données ont été uniformément distribuées.

Filtre 1. Gain d'information (Information Gain - IG)

L'utilisation du Gain d'Information a été proposée par Hunt, Marin et Stone en 1966 [Hunt 1966]. Cette technique mesure la diminution de l'entropie lorsqu'un attribut f est donné ou non [Forman 2003]. IG et les mesures similaires comme : index de Gini [Breiman 1984] et la mesure de distance [Lopez de Mantaras 1989] considèrent que ces attributs sont indépendants et par conséquent qu'ils ne sont pas utilisés dans les domaines où les attributs ont une forte dépendance [Kononenko 1994]. L'équation 1.4.1 représente la relation entre l'attribut f et la classe c, où D_v est un sous-ensemble de l'ensemble d'apprentissage D et l'attribut f a la valeur v_i.

$$IG(D, c, f) = Entropy(D, c) - \sum_{i=1}^{r} \frac{|D_{v_i}|}{|D|} Entropy(D_{v_i}, c) \tag{1.4.1}$$

L'entropie, qui dépend de la classe, est définie de la manière suivante avec des notations présentées dans le tableau 1.1 pour la classification binaire :

$$Entropy(D, c) = -\sum_{i=1}^{r} \left(\frac{n_{iP}}{n_i} \log_2 \frac{n_{iP}}{n_i} + \frac{n_{iN}}{n_i} \log_2 \frac{n_{iN}}{n_i} \right) \tag{1.4.2}$$

Filtre 2. Information Mutuelle (Mutual Information)

La mesure de l'IM de référence [Vergara 2014] mesure la dépendance statistique entre un attribut f et une classe c dans un ensemble de données d'images comme présenté ci-dessous :

$$I(f, c) = \log \frac{P_r(f \wedge c)}{P_r(f) \times P_r(c)} \approx \log \frac{A \times N}{(A + C) \times (A + B)} \tag{1.4.3}$$

où A est le nombre d'occurrences simultanées de l'attribut f et de la classe c ;
B est le nombre d'occurrences de l'attribut f sans la classe c ;
C est le nombre d'occurrences de la classe c sans l'attribut f ;
N est le nombre total d'images.

Quand les attributs sont indépendants sous l'hypothèse nulle, l'information mutuelle (empirique) du tableau de contingence est calculée comme suit :

$$n \cdot I = \sum_{i,j} n_{ij} ln \frac{n \cdot n_{ij}}{n_i \cdot n_j} = \sum_{i,j} n_{ij} ln(n_{ij}) - \sum_i n_i ln(n_i) - \sum_j n_j ln(n_j) + nln(n) \quad (1.4.4)$$

où $i = \{1..r\}$ et $j = \{P, N\}$.

Quand un attribut apparaît rarement dans une image, il prend une valeur faible. A l'inverse un attribut qui apparaît couramment dans une image prendra une valeur plus élevée. En effet, la valeur de l'information mutuelle d'un attribut dépend du nombre de ses occurrences dans une image. Pourtant, l'importance d'un attribut dans une image ne dépend pas forcément du nombre de fois où il apparaît dans l'image. Par exemple, dans un document de texte, le mot (attribut) "de" apparaît communément, et le mot "bateau" n'apparaît pas si communément que le mot "de". Toutefois, le mot "bateau" sert à distinguer ce document d'un autre document qui parle de la "voiture" par exemple. Par conséquent, le façon de calculer l'IM est aussi son point faible. Les auteurs de [Novovičová 2004] ont proposé une IM amélioré qui prend en compte l'information mutuelle entre chaque paire d'attributs.

Filtre 3. Mesure de χ^2 (Chi-Squared measure)

La mesure du χ^2 a été utilisée afin de calculer la relation entre une classe c et un attribut f avec un nombre k de degrés de liberté suivant :

$$X^2(c, f) = \sum_{i=1}^{k} \left(\frac{(n_{iP} - \mu_{iP})^2}{\mu_{iP}} + \frac{(n_{iN} - \mu_{iN})^2}{\mu_{iN}} \right) \quad (1.4.5)$$

Le nombre k de degrés de liberté est égal au nombre de colonnes du tableau de contingence moins une multiplié par le nombre de lignes du tableau de contingence moins une, soit : $k = (2 - 1) * (r - 1)$.

Le but du test χ^2 est de déterminer si la valeur observée du χ^2 correspond à l'apparition fréquente d'un attribut f et d'une classe c ou à l'apparition rare d'un attribut f et d'une classe c.

Filtre 4. Odds Ratio (OR) [van Rijsbergen 1981]

L'OR, également appelé *rapport des chances*, *rapport des cotes* ou *risque relatif rapproché* [Bernard 1987], se définit comme le rapport de la cote [13] de la classe c_P sur celle de la classe c_N pour une valeur v_i d'un attribut f. Nous pouvons présenter cette mesure selon les notations présentées dans le tableau 1.1 en considérant qu'un attribut f a seulement 2 valeurs : la valeur positive v_1 et la valeur négative v_2.

$$OR(D, c_+, f) = \frac{n_{1P} n_{2N}}{n_{1N} n_{2P}} \tag{1.4.6}$$

Dans le cas où un attribut ne s'apparente pas à la classe, une petite valeur fixe est assignée à cet attribut afin que la mesure puisse continuer de fonctionner.

1.4.1.1.2 Politique de filtrage

Après qu'un critère ait été choisi pour filtrer les attributs, une politique de filtrage est nécessaire. Plusieurs politiques peuvent être envisagées. Voici quelques exemples :
- Choisir n attribut à partir de m attributs en fonction de leur résultat sur le critère de filtrage (ex : choisir 50 attributs à partir de 100 attributs initiaux).
- Choisir tous les attributs qui ont une valeur de critère de filtrage supérieure à un seuil T.

Quelques variations de l'approche par filtrage sont listées en aval :
- Fast Correlation Based Filtering (FCBF) [Yu 2003],
- Correlation based Feature Selection (CFS) [Hall 1997],
- Mutual Information for Feature Selection (MIFS) [Battiti 1994],
- Multivalued Oblivious Decision Tree Feature Selection (MOD Tree) [Rakotomalala 2002].

Un des inconvénients de l'approche par filtrage est qu'elle traite les attributs de façon isolée, en conséquence, la redondance et la dépendance des attributs sont ignorées. Par exemple, les valeurs du critère de filtrage de deux attributs sont toutes les deux hautes de sorte qu'un attribut peut être redondant quand l'autre attribut a déjà été choisi.

Un attribut peut avoir une valeur du critère de filtrage faible mais en le combinant avec un autre, ces attributs combinés sont efficaces pour catégoriser les objets. Kohavi et John

13. Il s'agit de la cote des parieurs, par exemple la cote d'un cheval : un cheval avec la cote de 3 contre 1 a une chance sur 4 de gagner.

[Kohavi 1997] montrent des exemples concrets pour ces cas et distinguent la différence entre la pertinence et l'optimalité d'un attribut. [Novovičová 2004] propose une nouvelle approche par filtrage, plus complexe mais plus efficace, qui utilise l'Information Mutuelle pour calculer la valeur des groupes d'attributs plutôt que des attributs isolés.

1.4.1.2 Approches par encapsulation (Wrapper)

Les approches par encapsulation profitent de performances de l'algorithme de classification afin de choisir les meilleurs attributs. Par conséquent, les performances de la méthode de sélection d'attributs dépend de l'efficacité de la méthode de classification. L'approche par encapsulation a deux paramètres importants : la stratégie de recherche et l'algorithme de classification. Nous ne rentrerons pas plus en détails sur les algorithmes de classification car cela s'écarte du cadre initial de cette thèse. En revanche, le lecteur peut se référer aux références [Lotte 2007, Aggarwal 2012, Aggarwal 2014] pour plus de précisions.

L'idée de base de l'approche par encapsulation est la vérification de tous les sous-ensembles d'attributs possibles. De fait, elle est coûteuse en terme de calculs, par définition. Pour réduire ce coût, des stratégies de recherche de sous-ensembles d'attributs ont été proposées. Plusieurs stratégies ont été étudiées dans [Dash 1997, Kohavi 1997, Gutierrez-Osuna 2002]. Nous présentons quelques stratégies reconnues dans la communauté dans la sous-section en aval.

1.4.1.2.1 Stratégie de recherche

Dans cette partie, nous présentons cinq stratégies de recherche, qui permettent au lecteur de percevoir la diversité de ces approches :

Stratégie 1. Recherche de séparation et évaluation progressive (branch and bound search)

Cette méthode a été proposée la première fois par Land et Doig en 1960 pour résoudre un problème d'optimisation discrète [Land 1960]. Elle a été appliquée pour la sélection des attributs par Narendra [Narendra 1977].

L'algorithme commence par former la racine d'un arbre à partir de l'ensemble de tous les attributs. Chaque branche de cet arbre est un sous-ensemble d'attributs. Chaque sous-ensemble d'attributs est considéré comme une solution. L'algorithme explore uniquement les branches de cet arbre lorsque leurs valeurs par la fonction f est au dessus d'une borne

déterminée. Cette borne est la meilleure valeur de la fonction f parmi les solutions vérifiées jusqu'à présent. La fonction f est une fonction d'évaluation qui satisfait la propriété de monotonie. En d'autres termes, un sous-ensemble d'attributs A ne peut pas produire une valeur de la fonction f qui soit supérieure à la valeur de la fonction f d'un sous-ensemble d'attributs B si B contient A ($A \subset B$). Pour lutter contre la limite d'une fonction d'évaluation monotone, Foroutan et Sklansky ont introduit la recherche de séparation et d'évaluation progressive monotone approximative [Foroutan 1987].

Stratégie 2. Recherche séquentielle en avant ou arrière (sequential forward ou backward)

Cette approche est assez simple et facile à utiliser. Plusieurs versions de cette approche ont été proposées :

1. **Sélection en avant (Forward Selection, FS)** [Whitney 1971] La recherche séquentielle de sélection en avant commence par un ensemble vide. Le processus itératif commence par ajouter le premier attribut dans une liste classée par ordre décroissant, puis évalue cet attribut en terme de performances de classification en utilisant la validation croisée. Ensuite, il prend le deuxième attribut et le réévalue. Il répète l'opération jusqu'à ce qu'il n'y ait plus d'amélioration des performances. Cette approche a la propriété d'imbrication de sous-ensembles [14] entre les sous-ensembles d'attributs à chaque étape.
2. **Élimination arrière (Backward Elimination, BE)** [Marill 1963] La recherche séquentielle d'élimination arrière commence avec l'intégralité de l'ensemble des attributs dans une liste classée par ordre croissant. L'algorithme calcule les performances de la classification de cet ensemble d'attributs. Le processus itératif commence par calculer les performances de la classification de l'ensemble des attributs sans l'attribut qui a la plus faible note. Puis il compare les performances de la classification avant et après avoir enlevé cet attribut. Si l'amélioration ne baisse pas de façon significative, il supprime cet attribut. L'action est répétée jusqu'à ce qu'une diminution significative apparaisse. Pour éviter le problème de surapprentissage (overfitting), l'algorithme peut appliquer la validation croisée pour calculer les performances de classification. Cette approche a aussi la propriété d'imbrication de sous-ensembles entre les sous-ensembles d'attributs à chaque étape. Elle est coûteuse en terme de mémoire car l'intégralité des attribut doit être traité.
3. **Plus l moins r (plus-l-minus-r)** [Stearns 1976] L'approche plus l moins r est une approche suboptimale qui empêche l'imbrication des sous-ensembles, le point faible de FS et BE. Cette approche commence par vérifier les valeurs l et r. Si

14. Le sous-ensemble d'attributs obtenu après n étapes contient le sous-ensemble d'attributs obtenu après $(n-1)$ étapes. Cette propriété s'appelle "nested subset property" en anglais.

$l > r$, l'algorithme commence à partir de l'ensemble d'attributs vide. Le processus de sélection de l attributs et d'élimination de r attributs est répété jusqu'à ce que la fonction objective /fonction d'évaluation, n'augmente plus. Si $l < r$, l'algorithme commence à partir de l'ensemble d'attributs complet et le processus d'élimination de r attributs et de sélection de l attributs est répété jusqu'à l'apparition d'une diminution significative des performances de classification. Le principal inconvénient de cette approche est le manque d'une théorie pour prédire les valeurs de l et r pour obtenir le sous-ensemble optimal.

4. **Sélection flottante (Floating selection)** [Pudil 1994] Plutôt que de fixer les valeurs de l et de r, cette approche permet de déterminer ces valeurs à partir des données. La sélection séquentielle flottante avant (Sequential Floating Forward Selection SFFS) et l'élimination séquentielle flottante arrière (sequential floating backward sélection SFBS) sont deux variantes de cette approche.
5. **Recherche bidirectionnelle (Bi-directional search)** [Pohl 1971] La recherche bidirectionnelle est une implémentation parallèle de FS et BE. Afin de garantir que FS et BE convergent vers la même solution, les attributs qui sont déjà sélectionnés par FS ne sont pas éliminés par BE et les attributs qui sont déjà éliminés par BE ne sont pas sélectionnés par FS.

Stratégie 3. Recherche glouton du meilleur d'abord (greedy best-first search)

Avec cette stratégie, chaque sous-ensemble d'attributs possible est considéré comme un nœud dans un graphe. L'objectif de l'algorithme est de parcourir ce graphe en développant le nœud le plus prometteur parmi tous les nœuds qui ont été générés jusqu'à présent et qui n'ont pas encore été développés. Les nœuds les plus prometteurs sont déterminés selon une règle spécifique. Par exemple, Xu [Lei Xu 1988] propose de chercher le sous-ensemble optimal d'attributs en cherchant le chemin optimal (optimal path searching problem) dans un graphe orienté pondéré (weighted directional graph). Plusieurs versions plus sophistiquées de la recherche glouton du meilleur d'abord ont été proposées : [Pearl 1982, Aine 2007, Furcy 2005]. Cette stratégie obtient le résultat optimal global et prend moins de temps que la stratégie de séparation et d'évaluation progressive.

Stratégie 4. Recherche en faisceau (beam search)

Le terme "recherche en faisceau" a été proposé par Raj Reddy [Reddy 1977]. La recherche du meilleur d'abord (best-first search) ou la recherche en profondeur (breadth-first search) ont besoin de beaucoup de mémoire. La recherche en faisceau est une version améliorée les besoins en mémoire de la recherche en faisceau du meilleur d'abord (best-first

beam search) [Rich 2014] ou de la recherche en faisceau de parcours en largeur (breadth-first beam search) [Bisiani 1987]. Elle utilise une file d'attente bornée pour limiter la taille de la recherche.

Stratégie 5. Recherche avec génération aléatoire

Cette approche ajoute le hasard dans sa procédure de recherche pour échapper aux minima locaux. Quelques méthodes proposent d'utiliser la génération aléatoire : l'algorithme génétique, le recuit simulé, RGSS [Doak 1992], RMHC-PF1 [Skalak 1994], LVW [Liu 1996], etc. Nous présentons ci-dessous les deux méthodes les plus connues dans la communauté.

1. **Algorithme génétique (Genetic search)** La recherche génétique est inspirée de l'évolution de la nature. Chaque sous-ensemble d'attributs est considéré comme un *chromosome*. L'algorithme manipule un ensemble fini de chromosomes qui s'appelle *population*. Les chromosomes peuvent subir des *croisements* et des *mutations*. Le processus d'optimisation est effectué au cours des cycles qui s'appellent *générations/reproduction*. Au cours de chaque génération, un ensemble de nouveaux chromosomes est créé par croisement, mutation et évaluation. Seul l'ensemble des meilleurs chromosomes est retenu par l'évaluation de la fonction de fitness pour le prochain cycle de reproduction. Les mutations permettent d'éviter de tomber dans des minimaux locaux. Plusieurs auteurs ont exploré l'algorithme génétique pour la sélection des attributs comme [Siedlecki 1989, Leardi 1992, Vafaie 1993, Punch 1993, Yang 1998, Leardi 2000, Frohlich 2003, Oh 2004, Loughrey 2004].
2. **Recuit simulé (Simulated Annealing)** Le recuit simulé est une technique probabiliste pour l'approximation de l'optimum global d'une fonction de critère qui a été proposée par Metropolis [Metropolis 1953] et popularisée par Kirkpatrick [Kirkpatrick 1983]. Le recuit simulé emprunte son intuition fondamentale à la métallurgie. Un sous-ensemble d'attributs est considéré comme une molécule. Les molécules dans un métal cristallisent progressivement à un état de faible énergie quand leur température diminue lentement. Tous les grains de cristaux finiront par atteindre l'état d'énergie le plus bas tant que le métal est chauffé à une température initiale suffisamment élevée, et la vitesse de refroidissement est assez lente. La méthode de recuit simulé peut être contrôlée au moyen des paramètres du *processus de refroidissement* : la méthode de refroidissement, la vitesse de refroidissement et les conditions pour terminer le processus. Le recuit simulé peut échapper aux optima locaux et converger vers l'optimum global sous certaines conditions grâce au processus de refroidissement [Anily 1987, Cruz 1998]. Ci-dessous quelques applications de cette approche dans la sélection des attributs : [Sutter 1995, Debuse 1997, Meiri 2006, Lin 2008].

La combinaison d'une stratégie de recherche avec une méthode de classification permet de créer un grand nombre de méthodes de sélection d'attributs : [Siedlecki 1988], [Langley 1994], [Skalak 1994], [Rich Caruana 1994], [Maron 1994], [Moore 1994], [Fong 1995], [Moninder Singh 1995], [Aha 1996], [Bekkerman 2003].

1.4.1.3 Approches par intégration (embedded)

L'approche par intégration incorpore le processus de sélection des attributs dans la construction du modèle de catégorisation (classification). Par exemple, dans la construction de l'arbre de décision, les critères comme l'entropie, le gain d'information ou le Chi-carré [15] ont été utilisés afin de séparer les nœuds dans l'arbre. Le processus de séparation des nœuds est aussi un processus de sélection des attributs.

La combinaison entre un critère de sélection et une méthode par encapsulation, ou la combinaison entre une stratégie de recherche et un classifieur sont aussi placées dans cette catégorie. En effet, les auteurs de [Ding 2005, Peng 2005] proposent la combinaison entre le critère MRMR [16] et une méthode par encapsulation. Les auteurs de [Battiti 1994] proposent, quant à eux, la combinaison entre la stratégie de recherche glouton et le classifieur par réseaux de neurones. Mundra et Rajapakse incorporent le critère MRMR dans la procédure de catégorisation des attributs du classifieur SVM-RFE pour sélectionner les attributs pertinents en bio-informatique [Mundra 2010].

En outre, il existe des méthodes de sélection d'attributs qui calculent la pondération des attributs dans l'étape de construction du modèle de classification. Par exemple, les méthodes en aval utilisent la classification s'appuyant sur la Machine à Vecteur de Support (SVM) ou sur les réseaux de neurones pour construire le modèle de classification. Différentes approches sont utilisées pour calculer la pondération des attributs :

- SVM-RFE (Support Vector Machine Recursive Feature Elimination) [Guyon 2002],
- EFS-SVM (Embedded Feature Selection-Support Vector Machine) [Archibald 2007],
- TFNN (three-layer feedforward neural network)[Setiono 1997],
- SBS-MLPs (Sequential Backward Selection - Multi-Layer Perceptrons) [Romero 2008].

15. L'algorithme Iterative Dichotomiser 3 (ID3) [Quinlan 1986] utilise l'entropie ou le gain d'information, et l'algorithme CHAID [Kass 1980] utilise le χ^2 pour construire l'arbre de décision.

16. Minimum Redundancy - Maximum Relevance, maximiser la pertinence et minimiser la redondance. Une méthode s'appuie sur l'Information Mutuelle.

Nous vous présentons ci-dessous quelques méthodes de sélection des attributs par intégration.

1.4.1.3.1 Iterative Dichotomiser 3 (ID3)

Quinlan propose l'algorithme ID3 [Quinlan 1986] pour effectuer la classification supervisée. ID3 construit un arbre récursif de décision. Cet arbre sert à classer de nouveaux objets. Plus les données pour l'apprentissage sont variées et nombreuses et plus la classification de nouveaux cas sera fiable.

L'algorithme commence avec l'ensemble des attributs A comme nœud racine. A chaque étape de la récursion, il calcule soit l'entropie $H(A)$ (équation 1.4.7) soit le gain d'information $IG(a)$ (équation 1.4.8) de tous les attributs restants de l'ensemble A pour la branche en cours. Ensuite, l'attribut qui a la valeur d'entropie la plus petite (gain d'information la plus grande) est choisi et l'ensemble A est divisé par l'attribut choisi pour obtenir les sous-ensembles d'attributs restants. L'algorithme continue à explorer récursivement chaque sous-ensemble en ne tenant plus compte que des attributs choisis aux étapes précédentes. L'opération de division d'un sous-ensemble s'arrête lorsque l'une des conditions suivantes est remplie :

- Tous les attributs dans le sous-ensemble appartiennent à la même classe et le nœud devient ainsi une feuille avec l'étiquette de la classe des objets.
- Tous les attributs ont été choisis mais les objets de ce sous-ensemble appartiennent à plusieurs classes différentes. Dans ce cas, le nœud devient une feuille avec l'étiquette de la classe la plus courante des objets.
- Il n'existe aucun objet dans le sous-ensemble. Autrement dit, aucun objet dans l'ensemble de référence ne correspond à une valeur spécifique de l'attribut sélectionné. Dans ce cas, le nœud devient une feuille avec l'étiquette de la classe la plus courante des objets de l'ensemble de référence.

L'entropie $H(A)$ est la mesure de la quantité d'incertitude des données de A :

$$H(A) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{1.4.7}$$

où,

- A est l'ensemble des attributs pour lequel l'entropie est en cours de calcul (change à chaque itération de l'algorithme).
- X est l'ensemble des classes de A.
- $p(x)$ est la probabilité qu'un objet appartienne à la classe x dans l'ensemble A.

Le gain d'information $IG(a, A)$ est la différence d'entropie mesurée à partir des valeurs d'entropie de l'ensemble A obtenues avant et après la division par l'attribut a. Autrement

dit, c'est la diminution de l'incertitude des données de A sur l'attribut a quand l'ensemble A est divisé par l'attribut a.

$$IG(a, A) = H(A) - \sum_{t \in T} p(t)H(t) \tag{1.4.8}$$

où,
- $H(A)$ est l'entropie de l'ensemble A.
- T est les sous-ensembles qui ont été créés à partir de l'ensemble A par l'attribut a tel que $A = \bigcup_{t \in T} t$.
- $p(t)$ est la proportion du nombre d'objets dans le sous-ensemble t par rapport au nombre d'objets dans l'ensemble A.
- $H(t)$ est l'entropie du sous-ensemble t.

1.4.1.3.2 C4.5

L'approche utilisant le gain d'information est biaisée par les attributs ayant un grand nombre de valeurs de sortie. Quinlan a proposé une extension de l'algorithme ID3 qui s'appelle l'algorithme C4.5 [Quinlan 1993]. Pour ce faire, il construit un arbre de décision à partir d'un ensemble de données d'apprentissage de la même manière que l'algorithme ID3 en utilisant le gain d'information normalisé (entropie relative normalisé) (normalized information gain, information gain ratio). Ce gain d'information normalisé est calculé comme suit :

$$IGR(a, A) = \frac{IG(a, A)}{SI(a, D)} \tag{1.4.9}$$

où,
- $IG(a)$ est le gain d'information de l'attribut a.
- $SI(a, D)$ est les informations potentielles générées en divisant l'ensemble de données A en v partitions correspondant à v valeurs de sortie (outcomes) de l'attribut a.

$$SI(a, D) = -\sum_{j=1}^{v} \frac{A_j}{A} \times \log_2(\frac{|A_j|}{A}) \tag{1.4.10}$$

1.4.1.3.3 CHAID (CHi-squared Automatic Interaction Detector)

CHAID a été proposée par Kass en 1980 [Kass 1980]. C'est l'une des plus anciennes méthodes de classification utilisant des *arbres de décision*. CHAID construit un arbre non binaire à partir de l'ensemble des données. A chaque étape de la récursion, il s'appuie sur le test du χ^2 pour déterminer la meilleure répartition suivante (the best next split). La sélection des attributs est effectuée en même temps que la construction de l'arbre.

Cependant, l'ensemble des données utilisées pour CHAID doit être suffisamment grand pour rendre l'analyse fiable parce qu'il utilise la segmentation multiple [17].

Nous avons présenté plus haut quelques méthodes de sélection des attributs qui sont applicables dans l'apprentissage supervisé. Certains chercheurs proposent de choisir un sous-ensemble d'attributs pour chaque classe afin d'optimiser la capacité de classifier des algorithmes de classification [Skalak 1994, Kononenko 1994, Frigui 2004].

1.4.2 Application au cas de l'apprentissage non supervisé

Dans cette sous-section, les techniques sont catégorisées de la même manière que dans la partie supervisée (trois approches) : filtre , encapsulation et intégration. Cependant, la distinction entre ces catégories dans les méthodes appliquées dans l'apprentissage non supervisé est moins nette que celle dans les méthodes appliquées dans l'apprentissage supervisé. En particulier la différence est moins nette entre l'approche par encapsulation et l'approche par intégration. En effet, les deux approches par encapsulation et par intégration choisissent les sous-ensembles d'attributs grâce à un algorithme de regroupement (clustering algorithm) quand l'approche par filtrage est indépendante de la méthode de regroupement. Cependant, l'approche par encapsulation profite de l'efficacité de l'algorithme de regroupement (clustering) pour choisir les attributs pertinents quand l'approche par intégration incorpore le processus de sélection des attributs pertinents dans la construction du modèle de données pour le regroupement.

Dans les deux applications que sont l'apprentissage supervisé et l'apprentissage non supervisé, la définition d'**attribut pertinent** est une question récurrente au sein de la communauté [Kohavi 1997, Blum 1997, Guyon 2003]. Typiquement, dans le processus de sélection des attributs, un ou plusieurs critères seront à définir pour évaluer la qualité des sous-ensembles d'attributs possibles. La pertinence des attributs est l'objectif de la définition de ces critères. Par exemple, les auteurs de [Kohavi 1997] proposent une définition de la **pertinence** d'un attribut et montrent les exemples de la relation entre la **pertinence** et l'**optimum** de l'attribut pour l'objectif de bien distinguer les groupes. La pertinence des attributs peut être déterminée en évaluant les performances du regroupement ou en mesurant la distance, l'entropie ou la dépendance des données.

L'approche par encapsulation est la dénomination lorsque les performances du regroupement sont utilisées pour définir la pertinence des attributs. Par exemple, les auteurs

17. La segmentation multiple d'une population est la séparation de cette population en groupes homogènes et distincts par plusieurs critères significatifs (pertinent, mesurable et accessible).

de [Dy 2004] proposent de chercher le nombre de clusters en fonction du sous-ensemble d'attributs. Les auteurs de [Hong 2008] proposent une méthode de sélection des attributs telle que la solution de regroupement du sous-ensemble des attributs choisis est la plus similaire à celle qui est obtenue par l'algorithme de regroupement d'ensembles [18]. Un algorithme de regroupement d'ensembles est une combinaison de plusieurs algorithmes de regroupement afin d'obtenir un meilleur regroupement par rapport à un algorithme seul.

La plupart des algorithmes proposés dans l'approche par intégration s'appuient sur le calcul de la pondération des attributs qui sont intégrés dans la construction du regroupement pour chercher le sous-ensemble des attributs optimaux. Autrement dit, les auteurs proposent de calculer la pondération des attributs en même temps que le regroupement afin de déterminer les attributs pertinents qui aident le regroupement. Par conséquent, chaque approche par intégration est spécifique à un algorithme d'apprentissage donné. En revanche, il existe plusieurs façons de faire le calcul de pondération s'appuyant sur plusieurs algorithmes de regroupement [Minsky 1969, Widrow 1960, Rumelhart 1986, Littlestone 1988].

En général, avec l'approche par encapsulation, un sous-ensemble d'attributs pertinents est trouvé pour tous les clusters. Toutefois, les propriétés intrinsèques locales des données [19] comptent au cours du regroupement [Li 2008]. Les auteurs de [Li 2008] ont proposé un algorithme de sélection des attributs localisés pour le regroupement qui cherche les sous-ensembles d'attributs optimaux pour chaque cluster. Cependant, cette approche n'est pas extensible pour les données de grande dimension.

Nous présentons ensuite quelques méthodes de sélection d'attributs selon les trois catégories : l'approche par filtrage, par encapsulation et par intégration.

1.4.2.1 Approches par filtrage

L'approche par filtrage choisit les attributs pertinents indépendamment de l'algorithme d'apprentissage.

18. L'algorithme de regroupement d'ensembles utilise plusieurs algorithmes de regroupement afin d'obtenir le meilleur résultat de prédiction de la regroupement. Il s'appelle "ensemble learning algorithm" dans le cas d'apprentissage supervisé ; et il s'appelle "consensus clustering" ou "clustering ensembles" en anglais.

19. En anglais : local intrinsic properties of data.

1.4.2.1.1 Laplacian Score (LS) [He 2005]

LS est une méthode de sélection d'attributs qui peut être utilisée à la fois dans le cas supervisé et le cas non supervisé. Elle s'appuie fondamentalement sur les algorithmes Laplacian Eigenmaps [Belkin 2001], Locality Preserving Projection [He 2004] et Spectral Graph [Chung 1997].

L'algorithme calcule la valeur du Laplacian Score de la $r^{ème}$ attribut L_r de la façon suivante :

$$L_r = \frac{\widetilde{f_r}^T L \widetilde{f_r}}{\widetilde{f_r}^T D \widetilde{f_r}} \tag{1.4.11}$$

où

f_{ri} est la valeur de l'attribut $r^{ème}$ de l'objet $i^{ème}$ (r = 1 .. n ; i = 1 .. m)

$f_r = [f_{r1}, f_{r2}, ..., f_{rm}]^T$

$D = diag(S1)$

$1 = [1, ..., 1]^T$

$\widetilde{f_r} = f_r - \frac{\widetilde{f_r}^T D1}{1^T D1} 1$

S est la matrice pondérée du graphe qui modélise la structure locale de l'espace de données,

La matrice $L = D - S$ est aussi appelée le Laplacien du graphe (Laplacian of graph G).

Parallèlement, l'algorithme construit un graphe pondéré G avec les arêtes reliant les nœuds à proximité les uns aux autres. S_{ij} évalue la similarité entre le $i^{ème}$ nœud et le $j^{ème}$ nœud. Un *bon* attribut est choisi en minimisant la fonction suivante :

$$L_r = \frac{\sum_{ij}(f_{ri} - f_{rj})^2 S_{ij}}{Var(f_r)} \tag{1.4.12}$$

où $Var(f_r)$ est la variance estimée du r-ème attribut. Quand $Var(f_r) = \widetilde{f_r}^T I \widetilde{f_r} = \frac{1}{n}(f_r - \mu 1)^T (f_r - \mu 1)$ avec I est la matrice identité, μ est la moyenne de $f_{ri}, i = \{1, ..., n\}$. $Var(f_r)$ devient la "standard variance"[He 2005].

He, Cai et Niyogi a montré l'efficacité de cette méthode en sélectionnant les attributs pertinents. Les meilleures performances du regroupement sont obtenues avec un nombre d'attributs inférieur à 200. Le nombre d'attributs en entrée est 1024 attributs. Mais étant donné qu'elle choisit les attributs selon leur pouvoir de préservation de la localisation, cela conduit à la possibilité de redondance des attributs.

1.4.2.2 Approches par encapsulation (Wrapper)

L'approche par encapsulation profite de l'efficacité de l'algorithme d'apprentissage pour évaluer l'efficacité des attributs.

1.4.2.2.1 COBWEB & Category Utility (COBWEB & CU) [Devaney 1997]

Cette méthode est un mélange entre la technique d'encapsulation et la technique de filtrage car elle utilise l'algorithme d'apprentissage afin de guider la sélection des attributs, mais la fonction d'évaluation calcule la propriété intrinsèque des données plutôt que la capacité prédictive. Elle utilise la fonction d'évaluation appelée Category Utility [Gluck 1985], le système de clustering AICC (Attribute-Incremental Concept Creator) et s'appuie sur le système COBWEB [Fisher 1987].

COBWEB représente les concepts probabilistes dans une structure arborescente hiérarchique utilisant une fonction d'évaluation pour les attributs symboliques. Les auteurs de [Devaney 1997] remplacent cette fonction d'évaluation par l'algorithme CLASSIT [Gennari 1989] afin de traiter les variables continues. Dans une tentative d'amélioration de l'efficacité de la recherche de sous-ensembles, le système AICC a été utilisée à la place du système COBWEB.

1.4.2.2.2 Feature Subset Selection using Expectation-Maximization clustering (FSS&EM)

Dy et Brodley [Dy 2000] présentent une approche de sélection des attributs par encapsulation utilisant l'algorithme de clustering espérance-maximisation (Expectation-Maximization clustering, EM). Cet algorithme de clustering est une application de l'algorithme espérance-maximisation (Expectation-Maximization algorithm) [Stigler 2007] afin d'estimer les paramètres du maximum de vraisemblance d'un modèle de mélanges gaussien.

Le nombre de clusters k dans l'algorithme de clustering EM dépend du sous-ensemble

d'attributs sélectionnés. Les auteurs utilisent une méthode de pénalisation. Un terme de pénalité est nécessaire pour éviter d'obtenir un cas où chaque objet est considéré comme un cluster car l'estimation du maximum de vraisemblance augmente en fonction du nombre de clusters utilisés. Cette méthode contient deux parties : la méthode de Bouman et al. [Bouman 2005] pour fusionner les clusters et le terme de pénalité pour le critère "log-likelihood" : Bayesian Information Criterion (BIC) [Schwarz 1978]. La nouvelle fonction objective est :

$$F(k, \Phi) = \log(f(X|\Phi)) - \frac{1}{2} L \log(Nd) \tag{1.4.13}$$

où $\log(f(X|\Phi))$ est la valeur de "log-likelihood" des données observées X compte tenu des paramètres Φ, L est le nombre de paramètres libres dans Φ, N est le nombre d'objets et d est le nombre d'attributs dans le sous-ensemble. L et Φ varient en fonction de k. La méthode de Bouman commence avec un grand nombre de clusters $k = K_{max}$. Puis ce nombre est décrémenté séquentiellement jusqu'à ce qu'il ne reste qu'un seul cluster. La valeur k est sélectionnée pour optimiser la fonction objective.

Dans cette approche, les auteurs utilisent aussi la "stratégie de recherche séquentielle avant" et deux critères d'évaluation de sous-ensembles : "Scatter Separability criterion" [Fukunaga 1990] et "Maximum de Vraisemblance" (Maximum Likelihood criterion). Ces deux critères d'évaluation ont des hypothèses, biais et limites différentes. En effet, "Scatter Separability criterion" préfère un sous-ensemble d'attributs dont les centres des clusters sont éloignés quand "Maximum Likelihood criterion" préfère un sous-ensemble d'attributs dont les clusters se conforment au modèle de Gaussien. Dans un article plus récent, ils ont proposé un schéma de normalisation des valeurs de critère "cross-projection" qui est en mesure d'éliminer ces biais [Dy 2004].

1.4.2.2.3 Random Cluster Ensemble (RCE)

L'algorithme RCE se concentre sur l'estimation de l'importance des attributs afin de choisir les attributs pertinents. Elghazel et Aussem présentent une extension du paradigme des forêts aléatoires pour les données non étiquetées conduisant à la sélection des attributs dans le cas d'apprentissage non supervisé [Elghazel 2010]. Cette extension s'appelle Random Cluster Ensembles car elle combine les résultats de plusieurs regroupements (clusterings) dans une partition de données unique sans l'accès aux attributs originaux.

RCE utilise la stratégie "BAGGING" (Bootstrap AGGregatING) [Breiman 1996] pour choisir un nouvel ensemble d'apprentissage avec remplacement à partir de l'ensemble de données original. En parallèle, RCE utilise la stratégie "random subspace" [Ho 1998] pour sélectionner aléatoirement un sous-ensemble d'attributs à partir de tout l'ensemble des attributs. Une solution de regroupement (clustering) est ensuite obtenue grâce à l'exécution

d'un algorithme de regroupement sur le sous-ensemble d'attributs sélectionné avec l'ensemble d'apprentissage choisi. La même étape est répétée r fois jusqu'à ce qu'un ensemble de clusters soit obtenu.

Les auteurs de RCE utilisent la technique dite "evidence accumulation" [Fred 2005] pour construire une matrice de co-association qui représente la similarité entre les modèles (patterns) en prenant les co-occurrences de paires de modèles dans le même cluster. Le regroupement final [20] est obtenu en exécutant l'algorithme traditionnel de regroupement ascendant hiérarchique [21] de la distance moyenne (UPGMA - Unweighted Pair Group Method with Arithmetic mean) sur cette matrice de co-association.

Une fois que le regroupement final est terminé, les attributs pertinents locaux dans chaque groupe final sont choisis grâce à l'aide de l'estimation "out-of-bag" (out-of-bag estimates) [Breiman 2001]. Autrement dit, l'estimation "out-of-bag" calcule la pertinence des attributs [22]. Ensuite, la sélection des attributs pertinents est effectuée par un test d'hypothèse statistique appelé "Scree test" [Cattell 1966]. Le test de "scree" consiste en la sélection des attributs ayant une valeur de seuil appelée "scree". Le "scree" correspond au point où le ralentissement maximal de la courbe se produit [Cattell 1966].

1.4.2.3 Approches par intégration (embedded)

On rappelle qu'une approche est dite embarquée lorsqu'elle intègre le processus de sélection des attributs dans la construction du modèle de classification.

1.4.2.3.1 W-k-means

Le calcul de pondération d'attributs dans le processus de regroupement K-moyennes a été proposé en 1984 par DeSarbo [DeSarbo 1984]. Les auteurs de [Huang 2005] propose un algorithme de K-moyennes qui calcule la pondération d'attributs automatiquement.

A chaque itération de l'algorithme W-k-moyennes, la pondération de chaque attribut s'appuie sur la variance de la distance mesurée à l'intérieur du groupe (intra-cluster, within cluster distance). Cette pondération est utilisée pour aider à déterminer à quel groupe l'attribut appartiendra à l'itération suivante. La pondération optimale de chaque attribut est déterminée quand l'algorithme converge. Autrement dit, elle est déterminée

20. consensus clustering
21. En anglais : Traditional average-link hierarchical agglomerative algorithm.
22. c.f. Voir [Elghazel 2010] pour plus de détails.

quand la fonction-objectif (objective function) est minimisée (voir [Huang 2005] pour plus de détails).

1.4.2.3.2 Discrimination d'attributs et regroupement simultanés (Simultaneous clustering and attribute discrimination, SCAD)

Les auteurs de [Frigui 2004] proposent une approche qui s'appuie sur le calcul de pondération d'attributs pour représenter la pertinence des attributs. Cette approche effectue le calcul de pondération d'attribut et le regroupement simultanément. Sa nature fait qu'elle est applicable à l'apprentissage non supervisé et aussi applicable à l'apprentissage supervisé. De plus, dans le cas supervisé, elle peut sélectionner un sous-ensemble d'attributs pertinents pour chaque catégorie.

A partir de l'approche SCAD, Frigui et Nasraoui proposent deux algorithmes SCAD1 et SCAD2 qui cherchent à atteindre le même objectif. Cependant, ils minimisent différentes fonctions-objectif (objective function). Dans leurs expérimentations, l'algorithme SCAD2 est meilleur que l'algorithme SCAD1 en terme de distinction des attributs pertinents et des attributs non pertinents. Nous invitons les lecteurs à se référer à l'article de Frigui et Nasraoui (c.f. [Frigui 2004]) pour plus de détails.

Un des points faibles de l'approche SCAD est que le nombre de groupes doit être déterminé à l'avance (information *a priori*). Néanmoins, les auteurs ont proposé une extension pour déterminer le nombre de groupes en utilisant une technique de clustering d'agglomération compétitive (AC) [Frigui 1997]. L'algorithme de regroupement d'agglomération compétitive [23] commence par répartir l'ensemble de données en un grand nombre de petits clusters. Lorsque l'algorithme progresse, les clusters adjacents se disputent les points de données, et les clusters qui perdent dans la compétition disparaissent progressivement. L'algorithme AC minimise la fonction-objectif suivante

$$J_A(B,U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N}(u_{ij})^2 d_{ij}^2 - \alpha \sum_{i=1}^{C}\left[\sum_{j=1}^{N} u_{ij}\right]^2 \tag{1.4.14}$$

sous réserve des contraintes :

$$u_{ij} \in [0,1] \forall i; 0 < \sum_{j=1}^{N} u_{ij} < N \forall i,j; \sum_{i=1}^{C} u_{ij} = 1 \forall j. \tag{1.4.15}$$

où

23. Competitive agglomeration clustering algorithm en anglais.

$U = [u_{ij}]$ est la matrice contrainte floue de C partitions de la taille $C \times N$. u_{ij} est le degré d'appartenance d'un point x_j dans le cluster β_i.

$X = \{x_j | j = 1, ..., N\}$ est un ensemble de N vecteurs d'attributs dans un espace d'attributs n-dimensionnel.

$B = (\beta_1, ..., \beta_C)$ représente un C-uplet [24] de prototypes. Chaque prototype décrit un cluster dans l'ensemble de C clusters. Le nombre de clusters C est mis à jour de façon dynamique dans l'équation 1.4.14.

d_{ij}^2 représente la distance d'un point x_j au prototype β_i.

α est un paramètre d'ajustement. La valeur de α doit être initialement petite pour encourager la formation de petits clusters. Ensuite, elle devrait être augmentée graduellement pour favoriser l'agglomération. Après quelques itérations, lorsque le nombre de clusters devient proche de l'optimum, la valeur de α devrait à nouveau décroître lentement pour permettre à l'algorithme de converger [Frigui 2004].

Dans la fonction-objectif 1.4.14, le premier terme contrôle la forme et la taille des clusters et favorise les partitions en de nombreux clusters, tandis que le second terme pénalise les solutions ayant un grand nombre de clusters et encourage l'agglomération de clusters. Lorsque les deux termes sont combinés et α est choisie d'une manière correcte, la partition finale minimisera la somme des distances intra-cluster [25], tout en partitionnant les données dans le plus petit nombre possible de clusters.

Un autre point faible de l'approche SCAD est qu'elle ne prend pas en compte la corrélation entre les attributs et qu'elle n'est pas robuste au bruit.

1.4.2.3.3 Approche Q-alpha

Les auteurs de [Wolf 2005] présentent une approche algébrique à pondération d'attributs servant à sélectionner les attributs pertinents. Cette approche utilise l'algorithme d'apprentissage comme prédicteur pour guider la sélection des attributs tout en évitant les calculs coûteux associés à l'approche par encapsulation. En effet, elle utilise des propriétés spectrales [26] de la matrice d'affinité des noyaux de clusters [27] pour guider la recherche des

24. En mathématiques, un C-uplet ou C-uple est une collection ordonnée de C objets (C est un entier naturel), appelés "composantes" / "éléments" / "termes" du C-uplet.

25. Minimiser l'inertie intra-classe pour obtenir des clusters/grappes les plus homogènes possibles.

26. Les propriétés spectrales d'une matrice sont les propriétés qui s'appuient sur le spectre de cette matrice.

27. Un noyau d'un cluster est le centre de ce cluster.

attributs pertinents. En d'autres termes, la recherche d'un sous-ensemble d'attributs est effectuée par l'optimisation des propriétés spectrales désirées, plutôt que par des cycles explicites d'apprentissage et de prédiction comme dans l'approche par encapsulation.

La définition de la pertinence d'un attribut est liée à la qualité du regroupement des données dans les clusters. En d'autres termes, Wolf et al. mesurent la qualité d'un sous-ensemble potentiel d'attributs en terme de cohérence de regroupement des k premiers clusters. Dans ce contexte, les auteurs de [Wolf 2005] ont proposé l'algorithme Q-α : la cohérence de regroupement est représentée par les valeurs propres de la matrice standard d'affinité [28] et celles de la matrice Laplacienne. L'approche algébrique servant à pondérer les attributs s'appuie sur la maximisation de la fonction d'optimisation :

$$\max_{Q,\alpha} trace(Q^T A_\alpha^T A_\alpha Q) \tag{1.4.16}$$

où $\sum_{i=1}^{n} \alpha_i^2 = 1$, $Q^T Q = I$ [29].

La matrice orthonormale Q de taille $q \times k$ et le vecteur de poids $\alpha = (\alpha_1, ..., \alpha_n)^T$ sont déterminés au point maximum de la fonction d'optimisation.

M est la matrice d'entrée de taille $n \times q$ tel que les lignes sont $m_1^T, ..., m_n^T$.

A_α est la matrice d'affinité $A_\alpha = \sum_{i=1}^{n} \alpha_i m_i m_i^T$.

Wolf et Shashua ont proposé aussi la méthode Q-α noyau en utilisant les méthodes à noyau avec une fonction noyau Q-α. Nous invitons les lecteurs à se référer à la page 1871-1872 de l'article [Wolf 2005] pour plus de précisions sur cette fonction.

Nous avons présenté en amont quelques méthodes de sélection d'attributs qui sont applicables à l'apprentissage non supervisé. Les chercheurs ont pensé à combiner les deux approches : la sélection et l'extraction des attributs. Par exemple, les auteurs de [Gu 2011] proposent d'intégrer le score de Fisher dans l'Analyse Discriminant Linéaire (ADL). L'objectif de cette approche est de chercher un sous-ensemble d'attributs à partir des attributs originaux et ensuite de transformer ce sous-ensemble en un ensemble de nouveaux attributs en utilisant l'ADL. Cette approche est appelée Linear Discriminant Dimensionality Reduction (LDDR).

28. Une matrice d'affinité A est une matrice qui contient des valeurs où une valeur A_{ij} est calculée par une mesure servant à déterminer la distance ou la similarité entre le point i en ligne et le point j en colonne. c.f. Annexe C.3.

29. I est la matrice identité / unité. Une matrice identité est une matrice carée avec des 1 sur la diagonale et des 0 partout ailleurs.

1.5 Conclusion

Le choix de méthode de réduction de dimension dépend de deux facteurs : le domaine d'application et le contexte.Par exemple, dans le domaine statistique, les méthodes d'extraction d'attributs sont mentionnées sous le nom de la réduction de dimension [30]. Les domaines comme l'apprentissage automatique ou la reconnaissance de formes utilisent les deux types de méthode d'extraction et de sélection en fonction des besoins. Malgré des techniques d'extraction d'attributs mieux adaptées pour compresser les données au lieu de catégoriser les données [31], certains domaines d'application (par exemple : l'apprentissage automatique ou la reconnaissance de formes) utilisent les méthodes d'extraction d'attributs pour leur capacité à réduire la dimension. Néanmoins, le domaine d'application de bio-informatique (plus spécifiquement la sélection génétique lié à l'expression génomique) a besoin de conserver les propriétés qui sont dans les attributs originaux de sorte que les méthodes de sélection des attributs sont utilisées. D'ailleurs, les auteurs de [Kambhatla 1997, Ghahramani 1997, Mishra 2011] proposent d'utiliser les méthodes d'extraction pour sélectionner les attributs. Ou plus récemment, les auteurs de [Tan 2014] proposent d'utiliser les méthodes d'extraction sur les données d'expression génétique.

Le deuxième facteur est le contexte d'utilisation de la méthode de réduction de dimension : à savoir si celle ci est utilisée dans le cas d'apprentissage supervisé ou d'apprentissage non supervisé. Dans le cas supervisé, les algorithmes cherchent les attributs pertinents pour bien distinguer/catégoriser les différentes classes à l'aide des étiquettes [32]. En revanche, dans le cas non supervisé, sans les étiquettes (information *a priori*), les algorithmes visent à trouver les attributs pertinents afin de mieux distinguer les clusters.

Le tableau 1.2 synthétise dix-sept méthodes de réduction reconnues dans la communauté selon quatre propriétés :

- le type de méthode (*i.e.* statistique [33] ou logique [34].) ;
- le type de données d'entrées (*i.e.* continues ou discrètes.) ;
- l'application à quel type d'apprentissage (*i.e.* la nature des méthodes permet leur application dans le cas supervisé ou non-supervisé.) ;
- le moyen de faire la réduction (*i.e.* la nature des méthodes permet leur utilisation dans la sélection ou l'extraction des attributs.).

30. En anglais : dimensionality reduction, dimension reduction.
31. Exemple illustratif de [Fukunaga 1990] est présenté dans la figure 1.3.
32. La vérité terrain. "Groundtruth" en anglais.
33. c.f. La section C.2.1 de l'annexe C.
34. c.f. La section C.2.2 de l'annexe C.

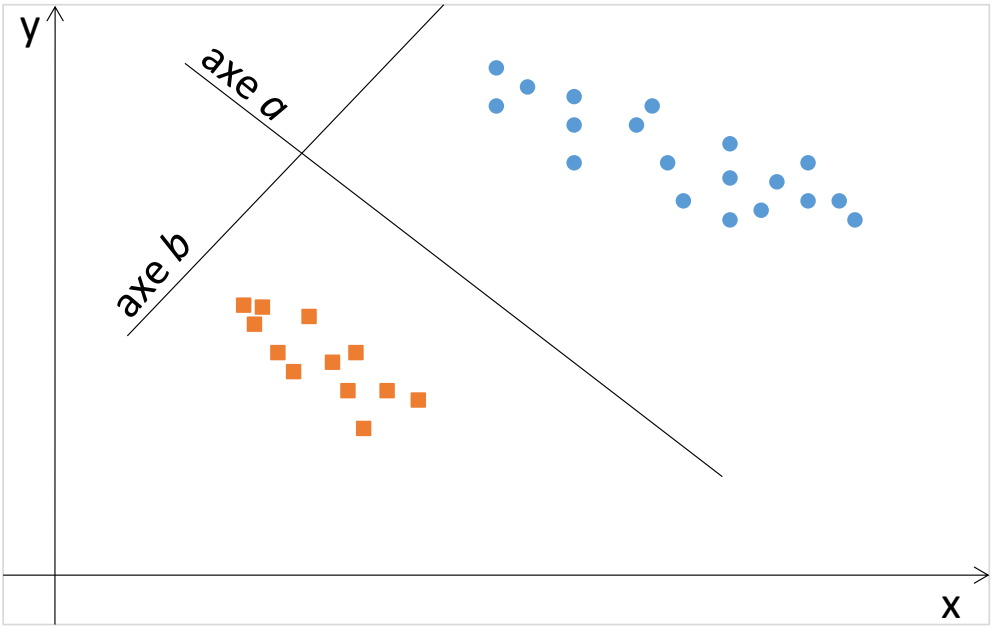


FIGURE 1.3: L'ACP est parfois un discriminateur pauvre. Cette figure est issue du travail de [Fukunaga 1990]. Dans cet exemple, les points carrés oranges et les points ronds bleus représentent deux groupes différents dans un espace de deux dimension x et y. Pour passer de deux dimensions à une dimension, l'ACP choisit la projection qui a la variance la plus haute de sorte que la dimension choisie soit l'axe a. Cependant, l'axe a n'est pas un bon discriminant entre deux groupes. Pour distinguer ces deux groupes, l'axe b est mieux que l'axe a.

Nos méthodes de réduction de dimension sont deux des méthodes logiques/ algébriques qui peuvent supprimer les attributs redondants. Le détail de nos méthodes est présenté dans le chapitre 4. Le chapitre 2 présente les notions et la théorie mathématique sur lesquelles nous nous appuyons afin d'introduire ces méthodes.

Method	Extraction/ Selection (E/S)	Supervised/ Unsupervised (S/U)	Statistics/ Logic (S/L)	Input features
LDA [Fisher 1936]	E	S	S	continuous
PCA [Hotelling 1933] /FA/PP/ICA	E	U	S	continuous
BFA [Belohlavek 2010]	E	U	L	discrete (binary)
FCBF [Yu 2003]	S	S	S	discrete
CFS [Hall 1999]	S	S	S	discrete
MIFS [Battiti 1994]	S	S	S	discrete
MOD Tree [Rakotomalala 2002]	S	S	S	discrete
reliefF Filtering [Kononenko 1994]	S	S	S	continuous/discrete
Laplacian Score (LS) [He 2005]	S	U	S	discrete
CU&COBWEB [Devaney 1997]	S	U	S	discrete
FSS&EM [Dy 2004]	S	U	S	continuous
The Q-alpha approach [Wolf 2005]	S	U	S	continuous
RCE [Elghazel 2013]	S	U	S	continuous
SCAD [Frigui 2004]	S	U	S	continuous

TABLE 1.2: Synthèse des méthodes de réduction de dimension selon quatre propriétés : la nature de ces méthodes, le cadre d'application de ces méthodes, le type de la méthode et le type de données d'entrée de ces méthodes.

Points clés

Positionnement

- ❏ Nous avons précisé les notations que nous utilisons dans ce manuscrit.
- ❏ Nous avons rappelé les types de données et précisé ceux que nous utilisons.
- ❏ Les méthodes de réduction de dimension les plus connues ont été mises en avant et différenciés.

Chapitre 2

Analyse Formelle de Concepts

Sommaire

2.1 Introduction 47
2.2 Notions 48
2.2.1 Contexte formel et treillis des concepts / treillis de Galois . . . 49
2.2.2 Sous-hiérarchie de Galois (AOC-poset) et AC-poset 56
2.2.3 Système de fermeture et treillis des fermés 58
2.3 Discussion 62
Points clés 65

2.1 Introduction

L'Analyse Formelle de Concepts est une méthode de représentation des connaissances qui définit le *treillis des concepts* à partir d'un *contexte*. Cette méthode est utilisée en informatique depuis des années 90. Le terme de *treillis des concepts* a été introduit en 1982 par Wille [Wille 1982]. Cette notion a été apportée depuis 1970 par Barbut [Barbut 1970] sous le nom *treillis de Galois* par référence au terme *correspondance de Galois*, qui lui-même a été introduit en 1944 par Ore [Ore 1944]. Le premier ouvrage de référence sur la théorie des treillis est le livre de Birkhoff de 1940 [Birkhoff 1940]. Dans un treillis, des éléments particuliers, appelés éléments irréductibles, se distinguent des autres. Les

sup-irréductibles ne correspondent pas à une borne supérieure. Les inf-irréductibles ne correspondent pas à une borne inférieure. La table des irréductibles est construite à partir des sup-irréductibles en ligne et des inf-irréductibles en colonne. Selon le théorème de Barbut [Barbut 1970], ne conserver que les irréductibles suffit à maintenir la structure du treillis (*i.e.* les différents combinaisons maximales d'objets possédant des attributs en commun). Ce sont les éléments de base de la théorie des treillis.

En outre, la notion d'*opérateur de fermeture* a été étudié en 1910 par Moore [Moore 1910]. Il est aussi appelé "hull operators" en anglais afin d'éviter la confusion avec la notion d'opérateur de fermeture qui a été étudiée en topologie. La relation entre l'opérateur de fermeture et la correspondance de Galois sera abordée dans le présent chapitre.

La sous-hiérarchie de Galois a été introduite en génie logiciel par Godin et al. [Godin 1993] pour la reconstruction d'une hiérarchie de classes. Cette notion a été introduite sous le nom d'AOC-poset (Attribute/Object Concept poset) par Ganter et Wille [Ganter 1999]. Notez que [Ganter 1999] utilise la relation flèche [1] pour caractériser la relation entre les concepts d'objets et les concepts d'attributs, mais sans faire référence à la sous-hiérarchie de Galois. La sous-hiérarchie de Galois est le sous-ordre du treillis de Galois simplifié qui est défini par Leblanc [Leblanc 2000]. Ce chapitre illustre les relations entre une sous-hiérarchie de Galois (AOC-poset) et une sous-hiérarchie sur l'ensemble des attributs (AC-poset), ainsi que les relations entre un contexte et un système de fermeture, un treillis des concepts et un treillis des fermés.

Ces notions seront présentées dans la section 2.2 et la discussion à ce sujet est dans la section 2.3.

2.2 Notions

Dans cette section, nous présentons formellement les éléments : le contexte formel, le treillis de Galois, l'élément irréductible d'un treillis, le concept d'objet/attribut, l'AOC-poset, le système de fermeture, le treillis des fermés et les relations entre eux qui consolident la réduction des attributs d'un contexte sans perte d'information.

1. La définition de la relation flèche se trouve dans [Ganter 1999].

2.2.1 Contexte formel et treillis des concepts / treillis de Galois

$\alpha\beta$	1	2	3	4	5	6	7	8	9	
a	x		x		x			x		$\alpha(a) = \{1, 3, 5, 8\}$
b	x	x	x			x	x	x		$\alpha(b) = \{1, 2, 3, 6, 7, 8\}$
c	x	x	x			x	x	x		$\alpha(c) = \{1, 2, 3, 6, 7, 8\}$
d	x	x	x		x	x	x	x		$\alpha(d) = \{1, 2, 3, 5, 6, 7, 8\}$
e	x	x	x			x	x	x		$\alpha(e) = \{1, 2, 3, 6, 7, 8\}$
f	x	x	x		x	x	x	x		$\alpha(f) = \{1, 2, 3, 5, 6, 7, 8\}$
g	x	x	x				x	x		$\alpha(g) = \{1, 2, 3, 7, 8\}$
h		x	x		x	x	x	x		$\alpha(h) = \{2, 3, 5, 6, 7, 8\}$
i		x		x				x	x	$\alpha(i) = \{2, 4, 8, 9\}$
j		x		x				x	x	$\alpha(j) = \{2, 4, 8, 9\}$

$\beta(1) = \{a, b, c, d, e, f, g\}$
$\beta(2) = \{b, c, d, e, f, g, h, i, j\}$
$\beta(3) = \{a, b, c, d, e, f, g, h\}$
$\beta(4) = \{i, j\}$
$\beta(5) = \{a, d, f, h\}$
$\beta(6) = \{b, c, d, e, f, h\}$
$\beta(7) = \{b, c, d, e, f, g, h\}$
$\beta(8) = \{a, b, c, d, e, f, g, h, i, j\}$
$\beta(9) = \{i, j\}$

TABLE 2.1: Contexte formel.

Définition 2.2.1 (Contexte)**.** *Un* ***contexte formel*** $C = (\mathcal{O}, \mathcal{A}, R)$ *est une* ***table binaire****, qui se compose d'un ensemble d'objets* $\mathcal{O}$*, d'un ensemble d'attributs* $\mathcal{A}$ *et d'une relation binaire* R *entre* $\mathcal{O}$ *et* $\mathcal{A}$ *. Le contexte* C *pourra être noté* $C = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ *où* (α, β) *est une* ***correspondance de Galois*** *qui se compose de deux fonctions* α *et* β *:*

$$\alpha(X) = \{a \in \mathcal{A} \mid oRa \text{ pour tout } o \in X\}, X \subseteq \mathcal{O}.$$

$$\beta(Y) = \{o \in \mathcal{O} \mid oRa \text{ pour tout } a \in Y\}, Y \subseteq \mathcal{A}.$$

α *est une fonction isotone. i.e. elle satisfait la propriété :*

$$x \leqslant y \text{ implique } \alpha(x) \leqslant \alpha(y)$$

β *est une fonction antitone. i.e. elle satisfait la propriété :*

$$x \leqslant y \text{ implique } \beta(x) \geqslant \beta(y)$$

Exemple : La table 2.1 montre le contexte formel qui servira de fil d'ariane aux différentes structures présentées dans ce chapitre.

Le treillis des concepts est un graphe construit à partir d'un contexte. Un concept est un nœud de ce graphe qui contient un ensemble maximum d'objets et leurs attributs communs. Plus formellement, un concept se définit par :

Définition 2.2.2 (Concept)**.** *Un concept* (X, Y) *du contexte* $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ *est défini par*

$$X \subseteq \mathcal{O}, Y \subseteq \mathcal{A},$$
$$\alpha(X) = Y, \beta(Y) = X$$

où X *est appelé l'**extension** du concept* (X, Y)*, et* Y *est appelé l'**intension** du concept* (X, Y)*.*

Exemple : Soit $X = \{a, b, c, d, e, f, g, h\}, Y = \{1, 3, 8\}$. Nous avons :

$$\alpha(X) = \{1, 3, 8\} = Y \text{ et } \beta(Y) = \{a, b, c, d, e, f, g, h\} = X.$$

Nous en déduisons que $(\{a, b, c, d, e, f, g, h\}, \{1, 3, 8\})$ est un concept du contexte 2.1.

Définition 2.2.3 (Treillis des concepts/Treillis de Galois)**.** *Le treillis des concepts du contexte* $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ *est un couple* $\mathcal{L} = (\mathcal{C}, \leqslant)$ *où* $\mathcal{C}$ *est l'ensemble des concepts du contexte et* $\leqslant$ *est une relation d'ordre partiel sur* $\mathcal{C}$ *définie, pour deux concepts* $(X_1, Y_1), (X_2, Y_2) \in \mathcal{C}$*, par :*

$$(X_1, Y_1) \leqslant (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \Leftrightarrow Y_1 \supseteq Y_2$$

Exemple : Le treillis des concepts du contexte de la table 2.1 est présenté dans la figure 2.1 par son diagramme de Hasse (i.e. sans les arcs de transitivité et de réflexivité qui s'en déduisent).

Un treillis des concepts est une structure qui possède les propriétés algébriques de treillis :

Définition 2.2.4 (Treillis algébrique / Lattice)**.** *Un treillis est un couple* $\mathcal{L} = (\mathcal{S}, \leqslant)$ *où :*

1. $\leqslant$ *est une relation d'ordre sur l'ensemble* S*, i.e. une relation binaire* R *qui vérifie les propriétés suivantes :*

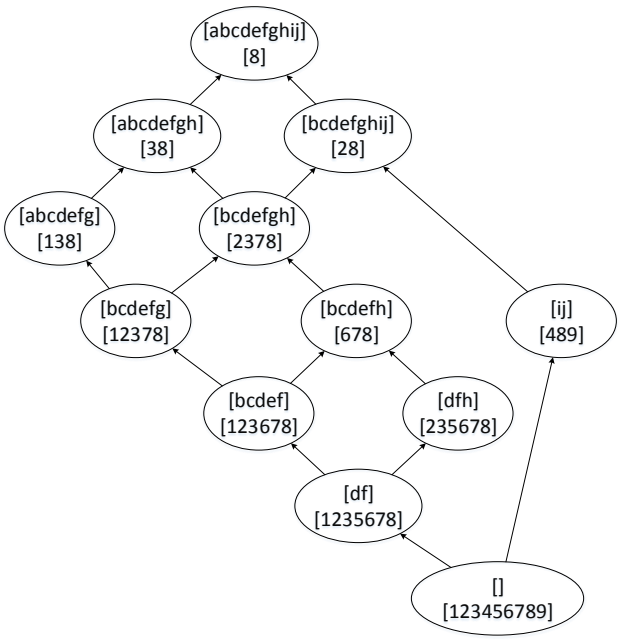


FIGURE 2.1: Treillis des concepts correspondant au contexte 2.1.

réflexivité *: pour tout $x \in S$, on a xRx*

anti-symétrie *: pour tous $x, y \in S, xRy$ et yRx impliquent $x = y$*

transitivité *: pour tous $x, y, z \in S, xRy$ et yRz impliquent xRz*

2. *toute paire d'éléments x, y de S admet à la fois une borne supérieure et une borne inférieure :*

   $x \vee y$ *:* ***la borne supérieure de*** x ***et*** y *(join / supremum), est l'unique élément minimal (plus petit élément) de l'ensemble des successeurs (ou majorants) de x et y (ensemble des éléments $z \in S$ tels que $z \geq x$ et $z \geq y$).*

   $x \wedge y$ *:* ***la borne inférieure de*** x ***et*** y *(meet / infimum), est l'unique élément maximal (plus grand élément) de l'ensemble des prédécesseurs (ou minorants) de x et y (ensemble des éléments $z \in S$ tels que $z \leq x$ et $z \leq y$).*

Dans un treillis, les éléments irréductibles sont définis de la manière suivante :

Définition 2.2.5 (Élément irréductible d'un treillis)**.** *Un élément $x \in \mathcal{S}$ d'un treillis $\mathcal{L} = (\mathcal{S}, \leqslant)$ est* ***supremum-irréductible (join-irréductible / $\vee$- irréductible)*** *s'il n'est pas la borne supérieure d'un sous-ensemble E_x qui ne le contient pas :*

$$x = \vee E_x \text{ implique } x \in E_x$$

L'ensemble des supremum-irréductibles sera noté J.

Un élément $x \in \mathcal{S}$ est ***infimum-irréductible (meet-irréductible / $\wedge$- irréductible)*** *s'il n'est pas la borne inférieure d'un sous-ensemble E_x qui ne le contient pas :*

$$x = \wedge E_x \text{ implique } x \in E_x$$

L'ensemble des infimum-irréductibles sera noté M.

Les sup-irréductibles se caractérisent par un seul arc entrant et les inf-irréductibles ont un seul arc sortant dans le diagramme de Hasse du treillis.

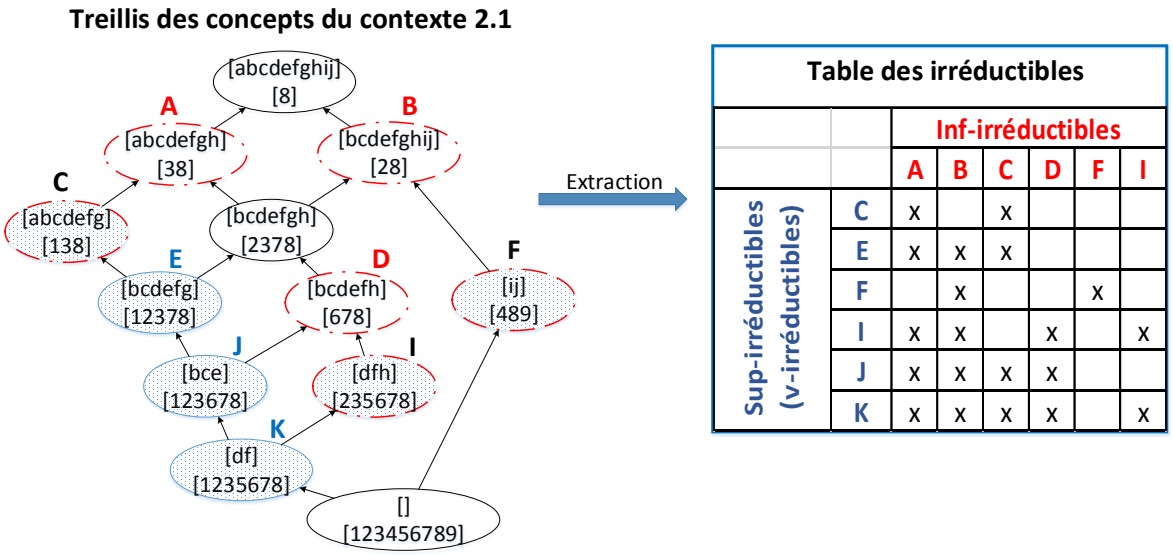


FIGURE 2.2: Table des irréductibles du treillis des concepts du contexte 2.1. Les infimum-irréductibles de ce treillis des concepts sont entourés en pointillés rouges et les supremum-irréductibles de ce treillis des concepts sont remplis en points bleus.

Exemple : Les irréductibles du treillis des concepts du contexte 2.1 sont présentés dans la figure 2.2 où les infimum-irréductibles sont entourés en pointillés rouges et les supremum-irréductibles sont remplis en points bleus.

Définition 2.2.6 (Table des irréductibles)**.** *La table des irréductibles d'un treillis $(\mathcal{S}, \leqslant)$ est un triplet (J, M, R) qui se compose des supremum-irréductibles $j \in J$ en ligne, des infimum-irréductibles $m \in M$ en colonne et d'une relation binaire R où jRm si $j \leqslant m$ dans le treillis.*

Exemple : La table des irréductibles du treillis des concepts correspondant au contexte 2.1 est présentée dans la figure 2.2 avec les supremum-irréductibles : C, E, F, I, J, K et les infimum-irréductibles : A, B, C, D, F, I.

Dans un treillis des concepts, on distingue des concepts particuliers tels que les concept d'objet et les concept d'attribut :

Définition 2.2.7 (Concept d'objet et Concept d'attribut)**.** *Dans un treillis des concepts du contexte* $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ *:*

Pour chaque objet $o \subseteq \mathcal{O}$*, le* ***concept d'objet*** (X_o, Y_o) *est la borne inférieure de l'ensemble des concepts qui contiennent l'objet* o *:*

$$\forall (X_i, Y_i) \text{ tel que } o \in X_i, \text{ nous avons } (X_o, Y_o) \leqslant (X_i, Y_i).$$

Pour chaque attribut $a \subseteq \mathcal{A}$*, le* ***concept d'attribut*** (X_a, Y_a) *est la borne supérieure de l'ensemble des concepts qui contiennent l'attribut* a *:*

$$\forall (X_i, Y_i) \text{ tel que } a \in Y_i, \text{ nous avons } (X_a, Y_a) \geqslant (X_i, Y_i).$$

Exemple : Le concept $(\{bcdefg\}, \{12378\})$ est un concept d'objet qui introduit l'objet g. Le concept $(\{abcdefgh\}, \{38\})$ est un concept d'attribut qui introduit l'attribut 3.

Remarque : Un inf-irréductible est un concept d'attribut et un sup-irréductible est un concept d'objet.

Le théorème fondamental de la théorie des treillis établit que la table des irréductibles est suffisante à sa reconstruction :

Théorème 2.2.1 ([Barbut 1970])**.** Tout treillis est isomorphe au treillis des concepts de sa table des irréductibles.

La réduction d'un contexte est la conservation des éléments irréductibles (objets irréductibles et attributs irréductibles) dans le contexte. Un élément sup-irréductible du treillis est le concept d'objet du treillis des concepts. Un concept d'objet peut être le

concept d'objet d'un objet ou de plusieurs objets. Nous avons besoin de ne conserver qu'un seul objet pour conserver l'élément sup-irréductible du treillis des concepts. Cet objet est un **objet irréductible**. De manière similaire, un élément inf-irréductible du treillis est le concept d'attribut du treillis des concepts. Un concept d'attribut peut être le concept d'attribut d'un attribut ou de plusieurs attributs. Un seul attribut conservé suffit pour conserver l'élément inf-irréductible du treillis des concepts. Cet attribut est un **attribut irréductible**. Un objet dont le concept d'objet n'est pas un élément irréductible du treillis est un **objet réductible**. Un objet dont le concept d'objet est un élément irréductible du treillis peut être un objet réductible s'il existe un autre objet dont le concept d'objet est identique. De manière similaire, un attribut dont le concept d'attribut n'est pas un élément irréductible du treillis est un **attribut réductible**. Un attribut dont le concept d'attribut est un élément irréductible du treillis peut être un attribut réductible s'il existe un autre attribut dont le concept d'attribut est identique. La réduction d'un contexte consiste donc à sélectionner les concepts d'objet et les concepts d'attribut qui sont aussi irréductibles. Le **contexte réduit** est obtenu après la réduction du contexte.

Le **contexte réduit** correspond à la **table des irréductibles** : les objets irréductibles et les attributs irréductibles du contexte réduit correspondent respectivement aux supremum-irréductibles et aux infimum-irréductibles de la table des irréductibles (*i.e.* voir la figure 2.3.). L'ensemble des éléments irréductibles est le sous-ensemble des concepts d'objet et des concepts d'attribut. Le treillis des concepts construit à partir de cette table est isomorphe au treillis des concepts du contexte initial et au treillis des concepts du contexte réduit comme présenté dans la figure 2.3.

Le théorème de Barbut (Théorème 2.2.1) implique que les irréductibles suffisent pour représenter la structure du treillis (l'association entre les objets et les attributs). En effet, le treillis des concepts obtenu à partir du contexte initial est isomorphe au treillis des concepts construit à partir de la table des irréductibles. Ce qui signifie qu'une réduction du contexte initial au contexte réduit maintient la description des données par leurs correspondances maximales objets-attributs sous forme de concepts. Autrement dit, les relations entre les concepts du treillis sont conservées à travers la réduction des objets et attributs réductibles du treillis.

Chaque concept (X, Y) du treillis initial correspond au concept correspondant (X', Y') du treillis des concepts du contexte réduit (*i.e.* $X' = X \wedge J$, $Y' = Y \wedge M$). Nous en déduisons que la relation entre les objets X peut s'exprimer à travers les attributs Y' après réduction sur l'ensemble des attributs.

Exemple : A partir du contexte initial 2.1, les objets réductibles et les attributs réductibles sont déterminés comme suit :

Attribut 7 L'attribut 7 est réductible car son concept d'attribut $([bcdefgh], [2378])$ n'est pas un élément irréductible du treillis des concepts 2.1. Quand l'attribut 7 a été éliminé de l'ensemble des attributs, les concepts qui le contiennent continuent d'exister dans le treillis des concepts du contexte réduit (*i.e.* le treillis en bas à droite de la figure 2.3.).

Attribut 8 L'attribut 8 est réductible car son concept d'attribut $([abcdefghij], [8])$ n'est pas un élément irréductible du treillis des concepts 2.1. De manière similaire, l'attribut 8 a été supprimé de tous les concepts qui le contiennent du treillis des concepts du contexte initial.

Attribut 9 L'attribut 9 est réductible car son concept d'attribut $([ij], [489])$ est aussi le concept d'attribut de l'attribut 4. Nous ne gardons que le premier attribut dans l'ensemble des attributs $4, 9$ et supprimons l'attribut 9.

Objets c **et** e Les objets c et e sont réductibles car ils ont le même concept d'objet $[bcdef], [123678]$ que l'objet b. Le premier objet dans l'ensemble des objets b, c, e a été préservé en tant qu'élément représentatif et les autres ont été supprimés. Quand les objets c et e ont été éliminés de l'ensemble des objets, les concepts qui les contiennent existent toujours dans le treillis des concepts du contexte réduit (*i.e.* le treillis en bas à droite de la figure 2.3.).

Objet f De manière similaire, l'objet f est réductible car le concept $[df], [1235678]$ est le concept d'objet de l'objet d et de l'objet f.

Objet j De manière similaire, l'objet j est réductible car le concept $[ij], [489]$ est le concept d'objet de l'objet i et de l'objet j.

Le contexte réduit du contexte 2.1 (où les objets réductibles c, e, f, j et les attributs réductibles $7, 8, 9$ sont éliminés) est présenté en haut à droite de la figure 2.3. Cette figure illustre la relation entre le contexte 2.1, son treillis des concepts et son contexte réduit. Elle illustre ainsi la relation entre des objets irréductibles et des attributs irréductibles du contexte réduit avec les sup-irréductibles et les inf-irréductibles de la table des irréductibles.

La réduction d'un contexte est le calcul des irréductibles du treillis. Les éléments irréductibles sont les concepts d'objets ou les concepts d'attributs, nous introduisons l'AOC-poset (la sous-hiérarchie de Galois) dans la section suivante. Notre objectif est de réduire les attributs qui représentent les images sans perte d'information qui serve à bien distinguer les images. En d'autres termes, nous sommes intéressés par la réduction des attributs dans un contexte (i.e. calcul des inf-irréductibles). Cette réduction a pour but d'obtenir un contexte attributs-réduits en gardant les informations de la relation entre les objets

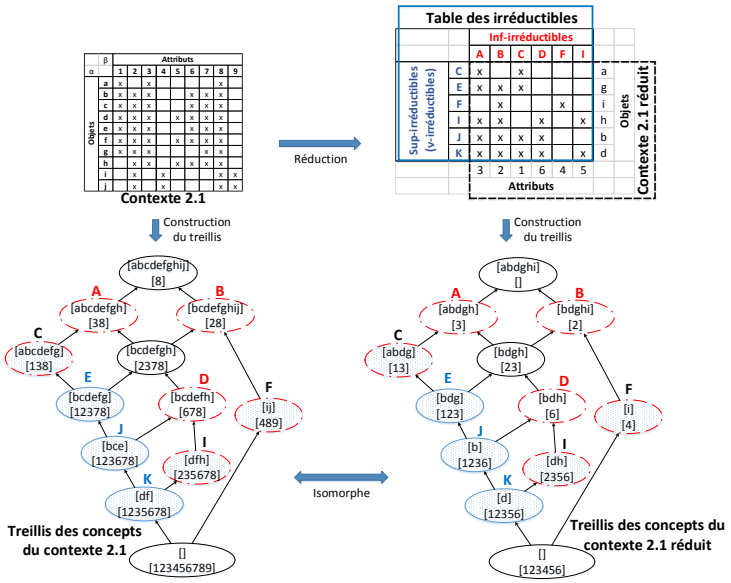


FIGURE 2.3: Illustration du théorème de Barbut en 1970.

dans le contexte initial. En effet, nous cherchons les inf-irréductibles qui sont des concepts d'attribut. La notion de l'AC-poset, sous-ordre de l'AOC-poset, est définie aussi dans la section suivante.

2.2.2 Sous-hiérarchie de Galois (AOC-poset) et AC-poset

Dans la littérature, la sous-hiérarchie de Galois (AOC-poset) est le sous-ordre du treillis des concepts réduit aux concepts d'objet et aux concepts d'attribut.

Définition 2.2.8 (Sous-hiérarchie de Galois/AOC-poset/Pruned concept hierarchy)**.** *La sous-hiérarchie de Galois d'un treillis des concepts* $\mathcal{L} = (\mathcal{C}, \leqslant)$ *est le sous-ordre de ce treillis contenant l'ensemble de ses concepts d'objets et de ses concepts d'attributs.*

Définition 2.2.9 (AC-poset/OC-poset)**.** *L'AC-poset (L'OC-poset) d'un treillis des concepts* $\mathcal{L} = (\mathcal{C}, \leqslant)$ *est le sous-ordre de ce treillis contenant l'ensemble des concepts d'attributs* $\mathcal{C}_A$ *(des concepts d'objets* $\mathcal{C}_O$ *respectivement).*

Par définition, l'AC-poset (l'OC-poset) d'un treillis des concepts est sous-ordre de l'AOC-poset.

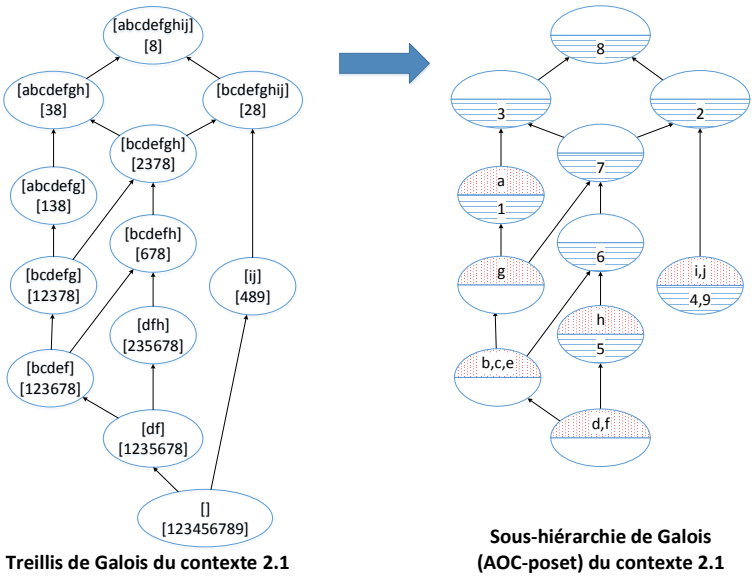


FIGURE 2.4: Sous-hiérarchie de Galois correspondant au treillis des concepts du contexte 2.1.

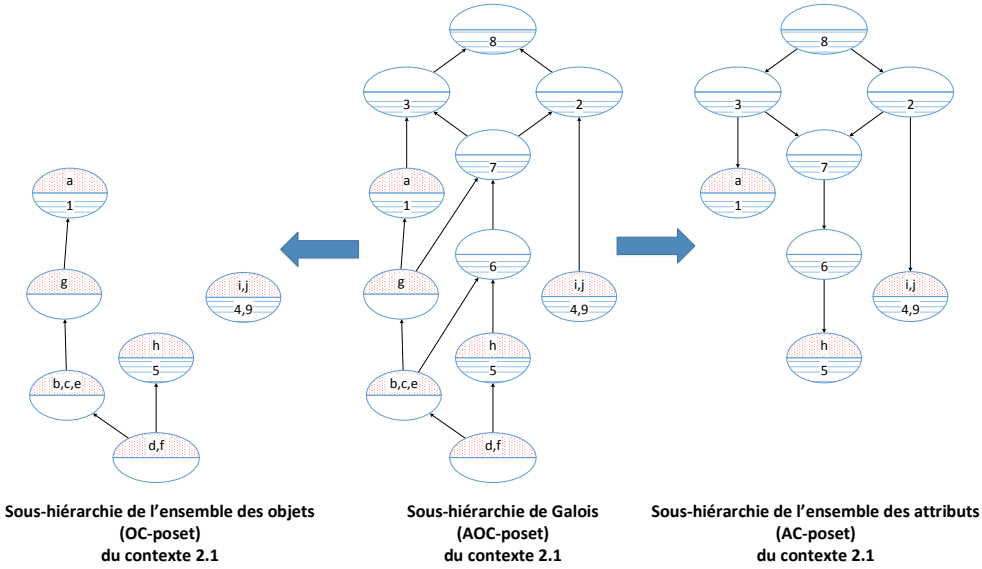


FIGURE 2.5: Sous-hiérarchie de Galois et l'OC-poset, l'AC-poset correspondant au contexte 2.1.

Exemple : La figure 2.5 illustre l'AOC-poset, l'OC-poset et l'AC-poset du treillis des concepts du contexte 2.1. La figure 2.6 illustre l'AOC-poset, l'OC-poset et l'AC-poset du treillis des concepts du contexte 2.1 réduit.

Les inf-irréductibles du treillis des concepts appartiennent à l'AC-poset du contexte initial et à l'AC-poset du contexte réduit. En effet, dans notre exemple, les inf-irréductibles du treillis des concepts (les nœuds sont entourés en pointillé rouge du treillis des concepts dans la figure 2.3) se trouvent dans l'AC-poset du contexte initial (le graphe à droite de la figure 2.5) et dans l'AC-poset du contexte réduit (le graphe à droite de la figure 2.6). Ces concepts d'attribut s'appellent les concepts d'attributs irréductibles. L'AC-poset du contexte réduit (le graphe à droite de la figure 2.6) ne contient que les concepts d'attribut irréductibles. De manière similaire, l'OC-poset du contexte réduit (le graphe à gauche de la figure 2.6) ne contient que les concepts d'objet irréductibles.

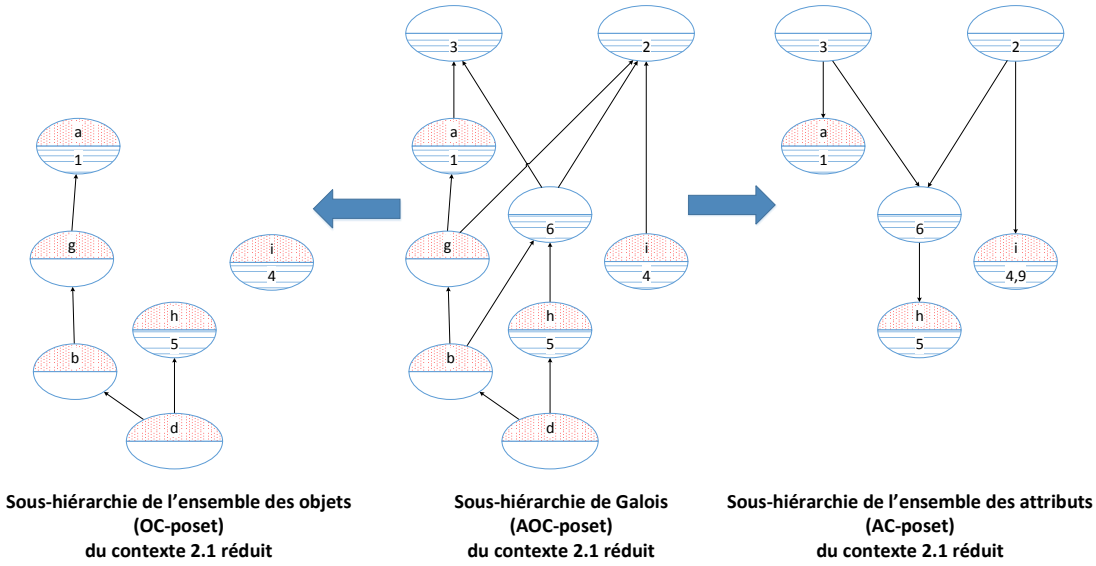


FIGURE 2.6: Sous-hiérarchie de Galois et l'OC-poset, l'AC-poset correspondant au contexte 2.1 réduit.

2.2.3 Système de fermeture et treillis des fermés

Dans cette section, nous définissons le système de fermeture et son treillis des fermés.

Définition 2.2.10 (Système de fermeture/Closure system)**.** *Un système de fermeture* (φ, S) *est défini par un opérateur de fermeture* φ *sur un ensemble fini* S*.*

Définition 2.2.11 (Opérateur de fermeture/Closure operator)**.** *Un opérateur de fermeture est une fonction* φ *définie sur un ensemble fini* S *satisfaisant les trois propriétés suivantes :*

- ***extensive*** *: $X \subseteq \varphi(X)$ où $X \subseteq S$.*
- ***croissante*** *: $X \subseteq Y \Rightarrow \varphi(X) \subseteq \varphi(Y)$ où $X, Y \subseteq S$.*
- ***idempotente*** *: $\varphi(\varphi(X)) = \varphi(X)$ où $X \subseteq S$.*

Définition 2.2.12 (Treillis des fermés/Closure lattice)**.** *Un treillis des fermés du système de fermeture (φ, S) est l'ensemble $\mathcal{F}$, ordonné par inclusion $\subseteq$, de tous les sous-ensembles* ***fermés*** *de l'ensemble S. Un sous-ensemble $X \subseteq S$ est appelé fermé si $\varphi(X) = X$. Un fermé est appelé aussi une fermeture.*

Nous ne rentrons pas plus en détails sur le système de fermeture et les notions le concernant car cela s'écarte du cadre initial de cette thèse. En revanche, nous invitons les lecteurs à se référer à la synthèse de Caspard et Monjardet [Caspard 2003] pour plus de précisions.

La composition $\alpha \circ \beta$ du contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ est monotone [2] et idempotente. Nous avons aussi la propriété $x \leqslant \alpha(\beta(x))$, $\forall x \in \mathcal{A}$. Par conséquent, la composition $\alpha \circ \beta$ est un opérateur de fermeture. Un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ se décompose en deux systèmes de fermeture : le premier, $(\alpha \circ \beta, \mathcal{A})$ défini par un ensemble d'attributs $\mathcal{A}$, avec $\alpha \circ \beta$ comme opérateur de fermeture ; le deuxième, $(\beta \circ \alpha, \mathcal{O})$ défini par un ensemble d'objets $\mathcal{O}$, avec $\beta \circ \alpha$ comme opérateur de fermeture.

Tout treillis des concepts se décompose en deux treillis des fermés : le treillis des fermés sur l'ensemble des objets et le treillis des fermés sur l'ensemble des attributs. Et la relation entre les concepts (X, Y), (X_2, Y_2) est conservée dans la relation entre les fermés Y, Y_2 car $(X, Y) \leqslant (X_2, Y_2) \Leftrightarrow X \subseteq X_2 \Leftrightarrow Y \supseteq Y_2$.

Exemple : Dans la figure 2.7, le treillis des fermés correspondant au système de fermeture de l'ensemble des attributs du contexte 2.1 est présenté à droite, et le treillis des fermés correspondant au système de fermeture de l'ensemble des objets du contexte 2.1 est présenté à gauche. Clairement, le treillis des concepts et ses treillis des fermés sont isomorphes (*i.e.* la figure 2.7).

Une réduction du système de fermeture s'exprime par la notion de système de fermeture réduit.

Propriété 2.2.1 (Système de fermeture réduit)**.** Deux systèmes de fermeture sont équivalents quand leurs treillis des fermés sont isomorphes. Birkhoff [Birkhoff 1940] et Barbut

2. Monotone signifie soit croissante soit décroissante.

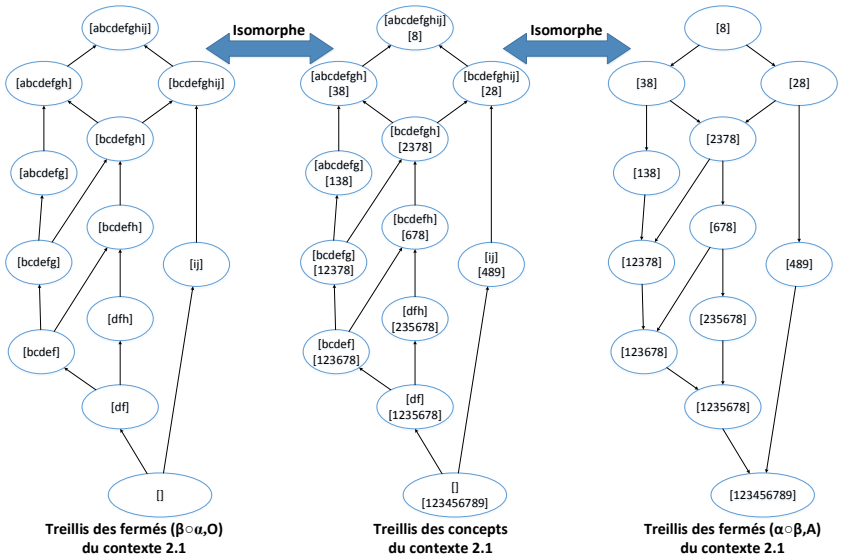


FIGURE 2.7: Treillis des concepts et treillis des fermés correspondant au contexte 2.1.

[Barbut 1970] établissent qu'un système de fermeture est réduit quand, pour chaque $x \in S$, la fermeture $\varphi(x)$ est un join-irréductible dans le treillis des fermés, c.-à-d. :

$$\forall E_x \subseteq S \text{ tel que } x \notin E_x, \varphi(x) \neq \varphi(E_x) \tag{2.2.1}$$

Remarque : Le treillis des fermés du système de fermeture réduit est appelé un *treillis des fermés réduit* ou un *treillis des fermés des irréductibles.*

Les élément irréductibles d'un treillis sont définis dans la définition 2.2.5 en amont. Cette définition est applicable pour un treillis des fermés.

Dans cette partie, pour simplifier la lecture, nous emploierons le terme "treillis des fermés" pour parler de "treillis des fermés sur l'ensemble des attributs".

Dans la section 2.2.1, nous avons présenté l'isomorphisme entre le treillis des concepts d'un contexte et le treillis des concepts de son contexte réduit. Cet isomorphisme s'étend naturellement aux treillis des fermés des irréductibles. Le treillis des concepts est isomorphe au treillis des fermés et au treillis des fermés réduit. Par conséquent, chaque concept (X, Y) correspond au concept irréductible $(X \wedge \mathcal{O}, Y \wedge \mathcal{A})$ et au fermé irréductible $Y' = Y \wedge \mathcal{A}$. La relation entre les objets X dans le concept (X, Y) qui est représentée

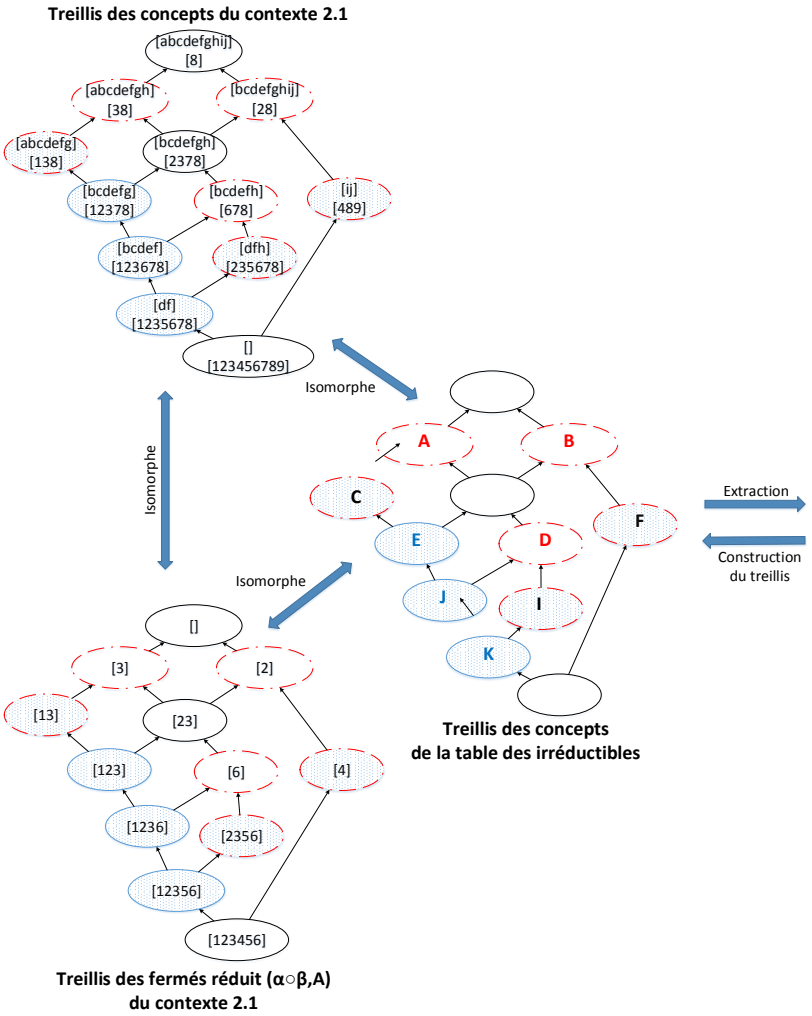


FIGURE 2.8: Application du théorème 2.2.1 sur le treillis des concepts et le treillis des fermés réduits correspondant au contexte 2.1.

par les attributs Y est conservée dans le fermé irréductible correspondant Y'. En d'autres termes, la réduction sur l'ensemble des attributs est faite sans perte d'information pertinente[3]. La figure 2.8 illustre l'isomorphisme entre le treillis des concepts et le treillis des fermés du contexte 2.1.

3. L'information servant à distinguer les objets.

2.3 Discussion

Dans ce chapitre, nous avons introduit les définitions et les propriétés des notions utilisées dans le domaine de l'Analyse Formelle de Concepts. A partir d'un contexte, un

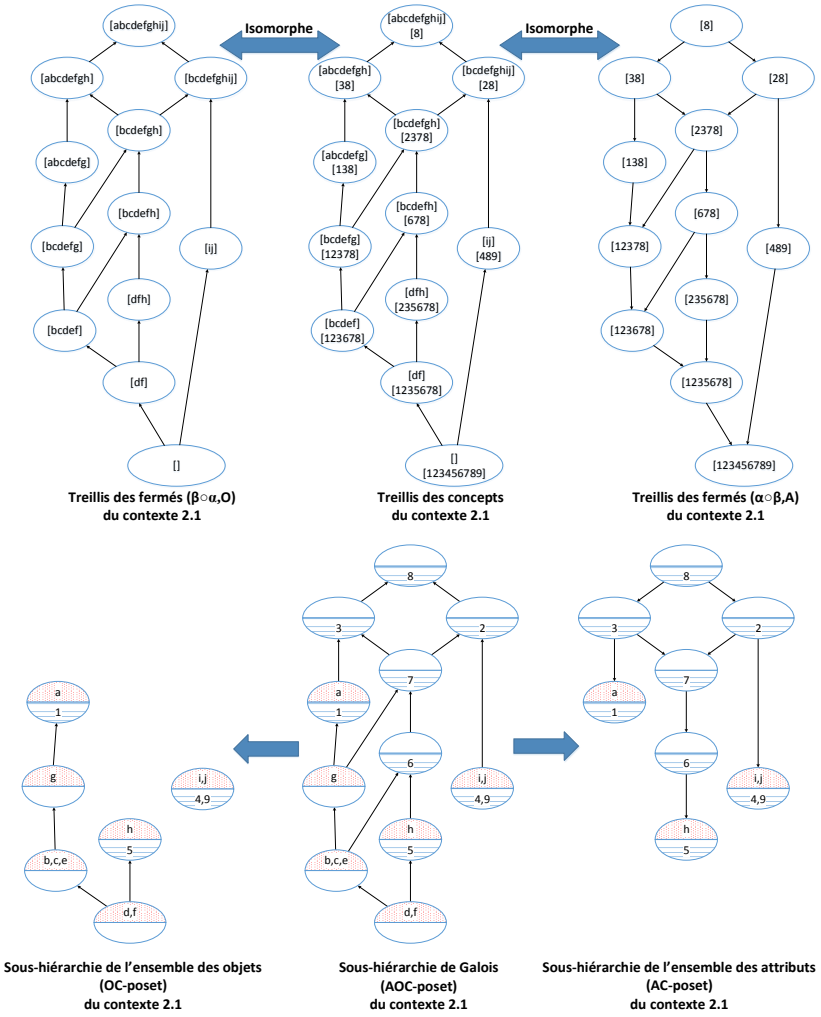


FIGURE 2.9: Synthèse : le treillis des concepts, les treillis des fermés, l'AOC-poset et l'AC-poset, l'OC-poset correspondant au contexte 2.1.

treillis de Galois, qui représente toutes les associations maximales entre les objets et les attributs, peut être construit. De plus, le théorème 2.2.1 de [Barbut 1970] nous permet d'introduire la réduction d'attributs sans perte d'information décrivant les relations entre les objets et les attributs.

En effet, ne conserver que les objets et attributs correspondant aux éléments irréductibles du treillis n'a aucun effet sur la structure du treillis [4], les différentes associations entre les objets et les attributs sont conservées. Un contexte peut se décomposer en deux systèmes de fermetures : système de fermeture sur l'ensemble des objets et système de fermeture sur l'ensemble des attributs. A partir d'un système de fermeture, un treillis des fermés peut être construit. Autrement dit, un treillis de Galois se décompose en deux treillis des fermés (voir la figure 2.9). La figure 2.9 illustre aussi l'OC-poset et l'AC-poset correspond à l'AOC-poset du contexte initial. Le treillis des concepts et le treillis des fermés du contexte 2.1 sont présentés côte à côte en haut de la figure 2.10. Et les treillis du contexte 2.1 attributs-réduits sont présentés en bas de la figure 2.10. Cette figure illustre ainsi l'AC-poset de l'ensemble initial d'attributs et de l'ensemble réduit d'attributs du contexte 2.1.

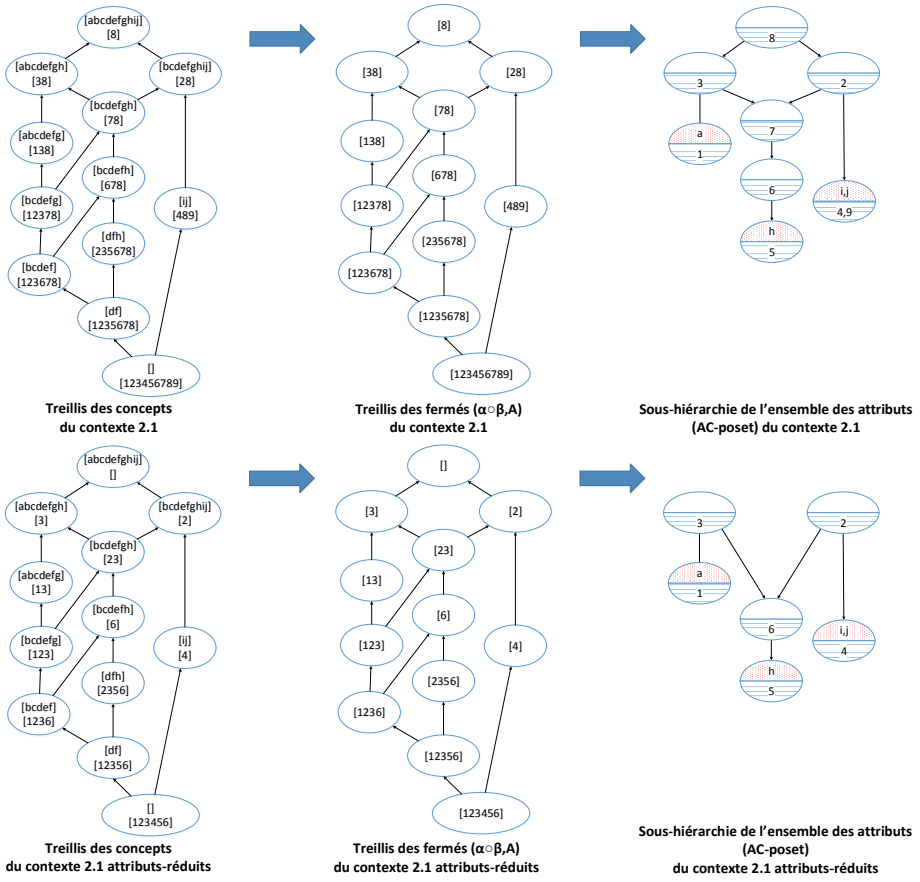


FIGURE 2.10: Synthèse : le treillis des concepts, le treillis des fermés et l'AC-poset correspondant au contexte 2.1 et au contexte attributs-réduits.

4. Le treillis avant et le treillis après réduction sont isomorphes.

Nous sommes intéressés par la réduction des attributs aux inf-irréductibles. Une telle réduction permet de conserver les informations pertinentes qui distinguent les objets. Cependant, la réduction d'un contexte est appliquée sur l'ensemble des objets et l'ensemble des attributs. La réduction d'un système de fermeture peut être appliquée seulement sur l'ensemble des objets ou sur l'ensemble des attributs. L'approche la plus simple pour faire la réduction serait de construire le treillis des concepts ou le treillis des fermés puis d'en extraire ses inf-irréductibles. Cependant, en théorie, la construction de ces treillis est exponentielle dans le pire des cas selon la taille du contexte.

Pour éviter de construire ces treillis, nous nous proposons de construire un graphe de précédence, proche de l'AC-poset, défini sur l'ensemble des attributs. La définition et l'utilisation de ce graphe pour réduire les attributs est détaillée dans la section 4.3 du chapitre 4.

Dans le but d'améliorer la réduction en acceptant une légère perte d'information, le graphe de précédence a l'avantage d'être étendu au cas flou. Nous avons proposé un algorithme de réduction correspondant. Cet algorithme est présenté dans la section 4.4 du chapitre 4.

Points clés

Positionnement

- Nous avons présenté les notations importantes de la théorie des treillis avant de nous intéresser plus précisément aux treillis et sous-hiérarchies de Galois ainsi qu'au système de fermeture.

Contributions

- Nous avons précisé les notations que nous utilisons dans ce manuscrit afin d'obtenir la réduction des attributs s'appuyant sur le treillis de Galois et le système de fermeture.

Chapitre 3

Modèle de sac de mots visuels

Sommaire

3.1 Introduction 67
3.2 Extraction des caractéristiques d'images 69
3.2.1 Scale-Invariant Feature Transform (SIFT) 70
3.2.2 Color Moment Invariants (CMI) 76
3.2.3 Détecteur de Harris-Laplace 77
3.3 Construction d'un dictionnaire des mots visuels 78
3.4 Encodage des caractéristiques d'une image dans un descripteur d'images (sac de mots visuels) 79
3.5 Conclusion 80
Points clés 81

3.1 Introduction

Le **modèle de sac de mots** est une représentation de documents textuels qui est utilisée couramment dans les domaines du traitement du langage naturel, et de la recherche d'information [Salton 1975]. La notion de **sac de mots** a été introduite en 1954 [Harris 1954] dans le contexte de la linguistique. Dans ce modèle, un document de texte est représenté sous forme d'un ensemble (aussi appelé "sac") de fréquences d'occurrence de

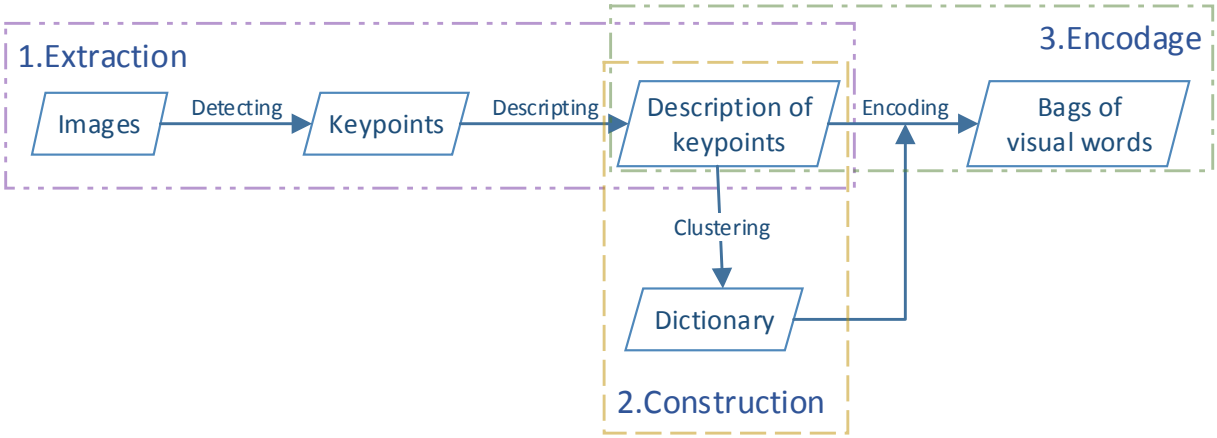


FIGURE 3.1: Approche classique permettant de construire le modèle de sac de mots visuels.

mots. Ces mots sont prédéterminés et l'ensemble de ces mots s'appelle un dictionnaire ou un vocabulaire. Un sac de mots permet de représenter et de catégoriser chaque document.

Ce principe a également été adopté et appliqué pour faire de la reconnaissance d'objets dans une vidéo par Sivic et al. [J. Sivic 2003]. Il a aussi été utilisé pour la représentation d'images dans plusieurs domaines comme la vision par ordinateur et l'apprentissage automatique : par exemple dans des applications liées à la recherche d'images par le contenu [Maillot 2006], la classification de la scène [Rasiwasia 2008], la reconnaissance d'objets [J. Sivic 2003, Bosch 2006]. Contrairement au traitement du texte, où les mots sont des vrais mots au sens littéral du terme, ici les mots visuels sont des clusters de caractéristiques locales. Il y a plusieurs dénominations comme **sac de mots visuels** [Zhang 2011, Chatfield 2011], **sac de visterms** [Maillot 2006, Quelhas 2007], **sac de caractéristiques** [Jiang 2007, Cao 2010], ou **sac de points d'intérêts** [Bolovinou 2012].

La figure 3.1 représente une chaîne de traitement classique permettant d'obtenir un modèle de sac de mots visuels, composée des étapes suivantes :

1. **Extraction** des caractéristiques d'images : cette étape est en charge de détecter et de décrire les caractéristiques dans une image. Classiquement, la **détection** identifie les caractéristiques locales qui s'appellent les points d'intérêts. Comme son nom l'indique, la **description** décrit chaque point d'intérêt en un descripteur local (un ensemble de valeurs).

2. Construction d'un **dictionnaire des mots visuels** à l'aide d'une méthode de clustering afin de regrouper les caractéristiques (i.e. descripteurs locaux) qui se ressemblent au sein d'un même **cluster**. Un cluster représente un **mot visuel**. Un **dictionnaire** est l'ensemble des mots visuels qui existent dans le lot d'images de la base de données.
3. **Encodage** des caractéristiques d'une image dans un descripteur d'images (sac de mots visuels). Cette étape crée un vecteur de mots visuels qui représente l'image. Ce vecteur s'appuie sur la quantification des descripteurs locaux dans le dictionnaire des mots visuels. Le **sac de mots visuels** et le **dictionnaire des mots visuels** [1] sont appelés aussi respectivement **codevector** et **codebook**.

Afin d'améliorer le modèle de sac de mots visuels, plusieurs suggestions ont été proposées : au lieu de quantifier les descripteurs locaux dans un seul cluster (hard-assignment), les auteurs de [Farquhar 2005, Philbin 2008, Van Gemert 2010] ont proposés un encodage "doux" (soft-assignment), pour lequel un descripteur local peut être représenté dans plusieurs clusters. Les auteurs de [Lazebnik 2006, Zhou 2010] proposent par ailleurs l'intégration de la position des descripteurs locaux dans les images dans le modèle de sac de mots visuels en utilisant une représentation spatiale pyramidale.

Dans ce chapitre, nous présentons le schéma classique que nous avons utilisé dans le chapitre 5.

3.2 Extraction des caractéristiques d'images

Dans le domaine du traitement d'images, l'extraction des caractéristiques vise à représenter l'essentiel de l'information portée par une image sous une forme réduite (vecteur de valeurs caractéristiques). Avant manipulation, les pré-traitements classiques permettent d'enlever le bruit, de changer la taille, de convertir l'espace de couleurs, etc.

Nous pouvons distinguer les caractéristiques globales et les caractéristiques locales. Les caractéristiques globales s'appuient sur les propriétés telles que la couleur, la texture ou la forme sur la totalité de l'image. Ces informations peuvent être extraites de l'image considérée dans son intégralité en utilisant les approches suivantes :

1. Aussi appelé le vocabulaire des mots visuels.

- L'histogramme de couleurs [van de Sande 2010], le moment de couleurs [Stricker 1995, Mindru 2004] pour la **couleur** ;
- Les matrices de co-occurrences en niveaux de gris [Haralick 1973], les filtres de Gabor [Weldon 1996], l'histogramme local de Fourier (Local Fourier Histogram - LFH) [Zhou 2001] pour la **texture** ;
- L'espace de courbure multi-échelle (Curvature Scale Space - CSS) [Mokhtarian 1998], les descripteurs de Fourier [Zahn 1972], les moments d'image [Hu 1962, Khotanzad 1990] pour la **forme**...

Les caractéristiques locales sont calculées en un nombre réduit de pixels par rapport à l'image entière, par exemple autour de points ou de régions d'intérêt. Afin de déterminer ces caractéristiques, deux phases de traitement sont utilisées : la détection et la description. Ces points/régions peuvent être détectés grâce à des :

- **détecteur de coins** (corner) (Harris [Harris 1988], FAST [Rosten 2006]), Curvature Scale Space [Mokhtarian 1998] ;
- **détecteur de contours** (edge) (Prewitt [Prewitt 1970], Canny [Canny 1986], Sobel [Boyle 1988]) ;
- **détecteur de régions (blob)** (Laplacian de Gaussian (LoG) [Lindeberg 1993], Différence de Gaussians (DoG) [Lowe 2004], determinant de Hessian (DoH) [Bay 2008]).

Une fois que les points/régions d'intérêts ont été détectés, un algorithme de description sera utilisé afin de décrire chaque point/région d'intérêt. Différents types d'algorithmes de description peuvent être utilisés : l'histogramme des orientations, l'histogramme de couleurs ou les dérivées partielles d'ordre N (N-jets). La nature des méthodes SIFT [Lowe 2004] ou SURF [Bay 2008] leur permet de s'appliquer dans les deux phases de détection et de description.

Nous détaillons ensuite les méthodes utilisées lors des expérimentations du chapitre 5 de ce manuscrit.

3.2.1 Scale-Invariant Feature Transform (SIFT)

La méthode SIFT a été proposée par Lowe [Lowe 2004] et contient quatre étapes principales :

3.2.1.1 Détection des extrema dans l'espace des échelles

Le Laplacien de Gaussiennes (LoG) détermine les extrema locaux en fonction de l'échelle et la localisation à l'aide du paramètre d'échelle σ et des coordonnées cartésiennes x et y. Ces extrema locaux sont des points d'intérêts potentiels. Le LoG est le résultat de la convolution d'une image $\mathcal{I}$ par un filtre gaussien $\mathcal{G}$ de paramètre σ qui est calculée selon l'équation suivante :

$$\mathcal{L}(x, y, \sigma) = \mathcal{G}(x, y, \sigma) * \mathcal{I}(x, y), \tag{3.2.1}$$

où

$$\mathcal{G}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{3.2.2}$$

Le LoG est coûteux en temps de calcul, c'est pourquoi l'algorithme SIFT calcule la Différence de Gaussiennes (DoG), qui est l'approximation du LoG. La DoG est définie par :

$$\mathcal{D}(x, y, \sigma) = \mathcal{L}(x, y, k\sigma) - \mathcal{L}(x, y, \sigma) \tag{3.2.3}$$

où k est une constante de l'algorithme. L'erreur entre DoG et LoG est proche de 0 quand k approche 1. La DoG d'une image est calculée entre deux images avec différents niveaux de flou (blurring) (*i.e.* σ et $k\sigma$).

Dans son article, Lowe a utilisé les valeurs des paramètres suivantes : $\sigma = 1.6$ et $k = \sqrt{2}$, 4 octaves [2] et 5 niveaux d'échelle (5 images floues) pour chaque octave. Le calcul de la DoG est répété pour chaque paire d'images floues dans une octave et pour chaque octave dans la pyramide de gaussiennes (*i.e.* la colonne à gauche de la figure 3.2). L'ensemble des images de la DoG construit la pyramide des différences de gaussiennes (DoG) (*i.e.* la colonne à droite de la figure 3.2).

Une fois que ces DoG sont calculées, les extrema locaux de l'image initiale sont recherchés à travers les échelles et spatialement dans l'image des DoG. Pour ce faire, un pixel est comparé avec ses 8 voisins dans la même image, et avec les 9 pixels à la même position dans l'image de l'échelle suivante ainsi qu'avec les 9 pixels à la même position dans l'image de l'échelle précédente (*i.e.* voir la figure 3.3). Si ce pixel est un extremum local, alors il est un point d'intérêt potentiel.

2. Un niveau donné - une octave par analogie avec la musique.

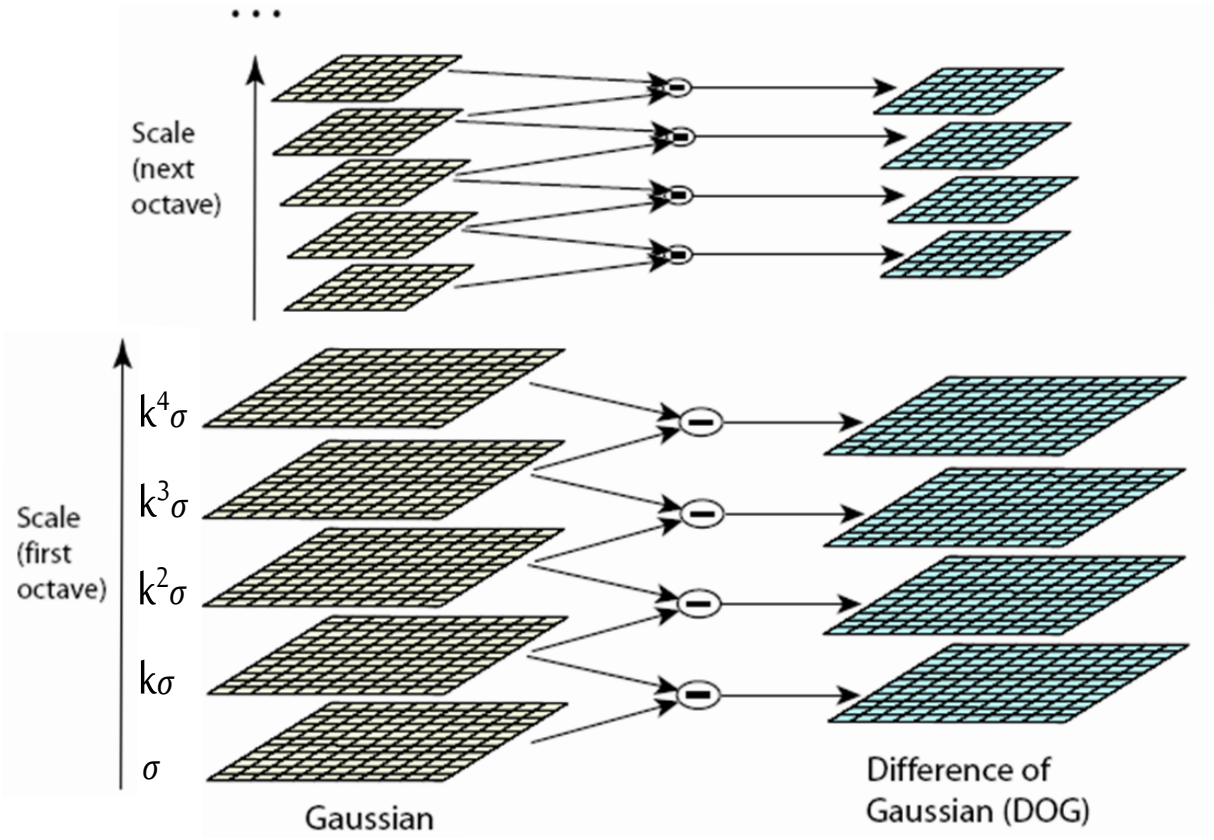


FIGURE 3.2: Pour chaque octave de l'espace d'échelle, la convolution de l'image par un filtre gaussien de paramètre σ est répétée et produit des images filtrées par une gaussienne qui sont présentées à gauche. Les images filtrées par une gaussienne côte à côte sont soustraites comme dans l'équation 3.2.1.1 pour produire l'image de DoG à droite. A chaque octave, l'image filtrée par une gaussienne est sous échantillonnée d'un facteur 2, et le processus est répété jusqu'au traitement de toutes les octaves. Cette figure est issue de la figure de l'article de Lowe [Lowe 2004].

3.2.1.2 Localisation des points d'intérêt

L'étape de détection des extrema détecte un nombre important de points d'intérêts potentiels, dont les points proches sont instables pour les petites perturbations de l'image. De plus, la localisation des points qui sont aux octaves supérieures de la pyramide (*i.e.* de résolution plus faible) est approximative.

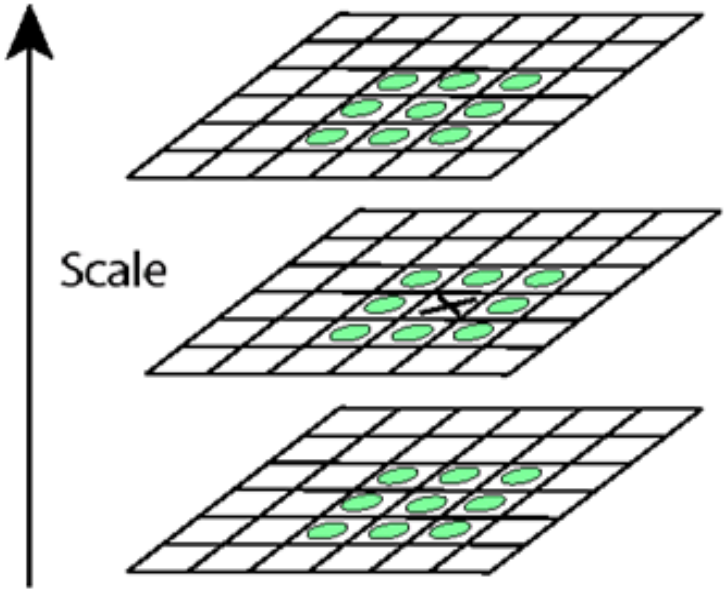


FIGURE 3.3: Comparaison d'un pixel (marqué avec X) avec ses 26 voisins (marqué avec des cercles) dans une zone de 3 échelles × 3 espaces servant à détecter les maximums et les minimums locaux. Cette figure est issue du travail de Lowe [Lowe 2004].

Pour préciser la position des points d'intérêts (i.e. x, y et σ), Lowe utilise le *développement en série de Taylor* $D(x, y, \sigma)$ à l'ordre deux de la fonction DoG, en prenant les coordonnées du point d'intérêt potentiel comme origine. La valeur de la fonction en ce point [3], $D(\hat{\mathbf{x}})$, est calculée de la manière suivante :

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{3.2.4}$$

où $\mathbf{x} = (x, y, \sigma)^T$ est le delta (offset) par rapport à ce point, et $\hat{\mathbf{x}}$ est la localisation de ce point. Le point d'intérêt potentiel est éliminé si l'intensité [4] de ce point est inférieure à 0.03 (une constante de la méthode proposée par Lowe).

La DoG considère un nombre important de points potentiels au niveau des contours. De ce fait, ces points sont instables et très sensibles au bruit. Un point instable situé sur un contour a une large courbure principale sur le contour et une faible courbure principale

3. extremum.
4. Aussi appelé le contraste. C'est la valeur de $|D(\hat{\mathbf{x}})|$.

dans la direction perpendiculaire. Lowe utilise la *matrice Hessienne 2×2* H pour calculer les courbures principales. Les valeurs propres de H sont proportionnelles aux courbures principales de D. Si elle est détectée comme un contour (i.e. une valeur propre est plus grande que l'autre), autrement dit, quand le ratio entre les 2 valeurs propres est plus grand qu'un seuil, alors ce point potentiel est rejeté. Ce seuil est un paramètre de la méthode fixé à 10 dans le travail de Lowe.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}. \tag{3.2.5}$$

3.2.1.3 Affectation d'orientation

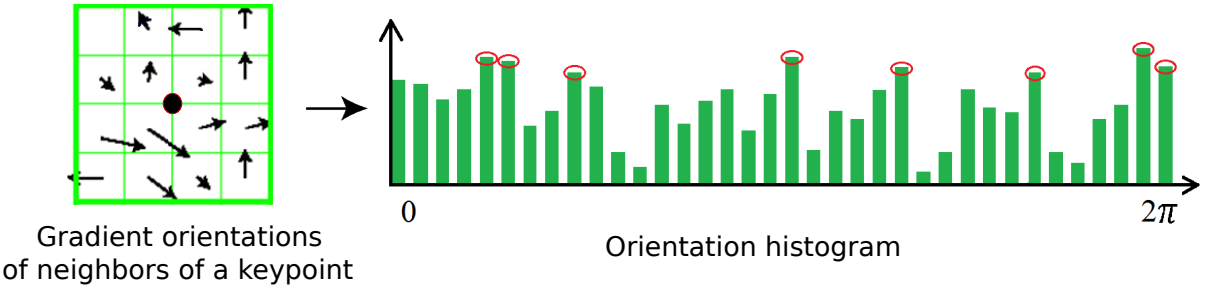


FIGURE 3.4: Histogramme d'orientations avec 36 boîtes contenant 360°(2π). Cette figure est inspirée du travail de Lowe [Lowe 2004].

Cette étape permet d'obtenir l'invariance du point d'intérêt à la rotation de l'image. Autrement dit, les mêmes descriptions doivent être obtenues à partir d'une même image, quelle que soit l'orientation de l'image. Afin d'atteindre cet objectif, il faut calculer la magnitude des gradients et l'orientation des voisinages du point d'intérêt (x_0, y_0, σ_0) en question. L'échelle du point d'intérêt σ_0 a été utilisée pour choisir la transformée Gaussienne L de l'image. De ce fait, tous les calculs se font sur cette image de manière invariante à l'échelle. Pour chaque voisin $L(x, y)$, à l'échelle σ_0, la magnitude du gradient $m(x, y)$ et d'orientation $\theta(x, y)$ sont calculées comme suit :

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{3.2.6}$$

$$\theta(x,y) = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{3.2.7}$$

Un histogramme d'orientation contenant 36 boîtes correspondant à 360°d'orientation est formé à partir des orientations du gradient des voisins dans le voisinage du point d'intérêt

en question. Chaque voisin ajoute dans l'histogramme la valeur de la magnitude du gradient et une fenêtre circulaire gaussienne avec le paramètre $\sigma = 1.5 \times \sigma_0$. Chaque boîte correspond à une orientation couvrant 10°d'angle. Toutes les orientations dominantes [5] qui atteignent 80% de la valeur de l'orientation dominante maximale sont choisies [6] pour créer des points d'intérêts. Ils ont la même valeur de localisation et d'échelle, mais différentes orientations. Ils sont définis donc par quatre paramètres (x, y, σ, θ). Ces nouveaux points d'intérêts contribuent à une meilleure stabilité lors de la comparaison des images.

3.2.1.4 Description des points d'intérêt

Pour chaque point d'intérêt, un voisinage de 16×16 pixels (valeur optimale fixée par défaut) autour de celui-ci est considéré. Cette zone est divisée en 16 sous-blocs de taille 4×4. Pour chaque sous-bloc, un histogramme d'orientation de 8 boîtes est crée. Ce qui donne un descripteur SIFT composé de 128 valeurs (4×4×8).

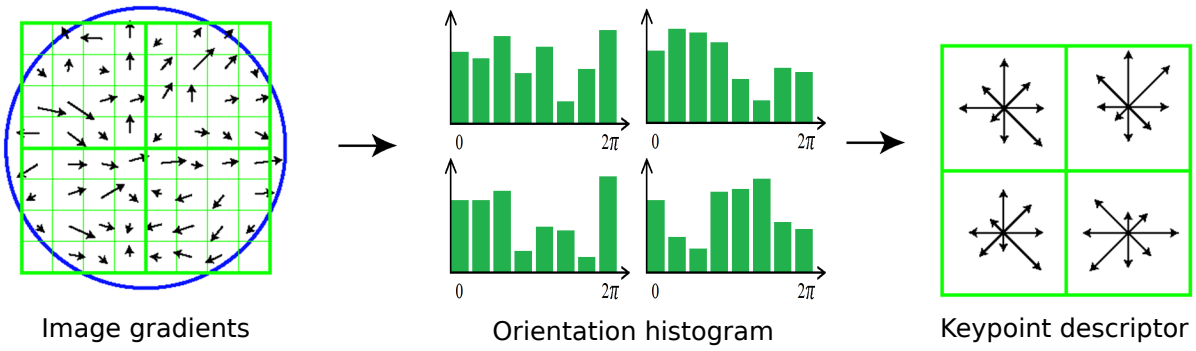


FIGURE 3.5: Un descripteur de points d'intérêt est créé en calculant la magnitude du gradient et l'orientation des voisins autour de la position du point d'intérêt, comme présenté à gauche. La fenêtre circulaire Gaussienne est représentée par le cercle bleu superposé. Les valeurs de l'orientation du gradient et la fenêtre circulaire gaussienne des voisins dans une sous-région de la taille 4×4 sont cumulées dans un histogramme d'orientation représenté au milieu. Cette figure illustre un ensemble de descripteurs de point d'intérêt de taille 2×2 (4 sous-régions) à partir d'un ensemble de 8×8 voisins. Dans son article, Lowe utilise un ensemble de descripteurs de taille 4×4 à partir d'un ensemble de 16×16 voisins. Cette figure est inspirée du travail de Lowe [Lowe 2004].

5. Les boîtes les plus grandes dans l'histogramme.
6. *i.e.* les boîtes avec un cercle rouge dans la figure 3.4.

La figure 3.5 illustre à gauche une région de 8×8 voisins [7] autour d'un point d'intérêt et un ensemble de descripteurs de taille 2×2 à droite, tandis que Lowe considère un voisinage de 16×16 voisins autour d'un point d'intérêt servant à créer un ensemble de descripteurs de taille 4×4. Tout d'abord, SIFT calcule la magnitude du gradient et l'orientation de chaque voisin dans l'ensemble des 8×8 voisins. Le cercle bleu superposé à la matrice est la fenêtre circulaire Gaussienne pour 4 sous-régions de taille 4×4. Dans chaque sous-région, les valeurs de magnitude du gradient ayant une direction proche sont additionnées dans l'histogramme d'orientation correspondant, présenté au milieu et à droite.

Comme l'indique son nom, SIFT est une méthode d'extraction de caractéristiques qui n'est pas affectée par l'échelle de l'image. De plus, l'étape *Affectation d'orientation* fait qu'elle est insensible à la rotation de l'image. Lowe montre que SIFT est stable indépendamment de la localisation, de l'échelle ou de l'orientation, et ce pour différents niveaux de bruit dans l'image [Lowe 2004].

3.2.2 Color Moment Invariants (CMI)

L'information de couleur dans les images s'avère très utile dans le domaine de la recherche d'images par le contenu [Swain 1991, Gevers 1996]. L'histogramme de couleurs est souvent utilisé afin de représenter la distribution de la couleur dans l'image [van de Sande 2010]. Cependant, l'histogramme de couleurs (et même le moment de couleur de Stricker et Orengo [Stricker 1995]) n'exploitent pas l'aménagement de l'espace couleur (*i.e.* RGB, Lab, YUV.). De ce fait, certaines informations essentielles peuvent être perdues. Mindru [Mindru 2004] définit une méthode utilisant des moments de couleur incluant l'information spatiale qui s'appelle *les moments généralisés de couleurs* [8]. En supposant que l'image est définie dans l'espace de couleur RVB, alors le moment généralisé de couleur M_{pq}^{abc} est défini par :

$$M_{pq}^{abc} = \int\int x^p y^q [R(x,y)]^a [V(x,y)]^b [B(x,y)]^c dxdy \tag{3.2.8}$$

M_{pq}^{abc} est un moment d'ordre $p+q$ et de degré $a+b+c$. $R(x,y), V(x,y), B(x,y)$ sont respectivement la valeur de R, V, B du pixel à la position (x,y) de l'image. Dans le cas de l'utilisation d'un autre espace de couleur, les moments peuvent être calculés d'une

7. Échantillons.
8. Ces moments s'appellent "generalized color moments" en anglais.

manière semblable. Ces moments peuvent être créés en fonction de la combinaison de deux paramètres : l'ordre et le degré. Par exemple, avec les moments au premier ordre M_{00}^{abc}, M_{10}^{abc}, M_{01}^{abc} et les moments au deuxième degré M_{pq}^{000}, M_{pq}^{100}, M_{pq}^{010}, M_{pq}^{001}, M_{pq}^{110}, M_{pq}^{011}, M_{pq}^{101}, M_{pq}^{200}, M_{pq}^{020}, M_{pq}^{020}, il existe 30 combinaisons possibles. Avec des valeurs différentes d'ordre et de degré, en multipliant les possibilités de combinaison, un grand nombre de moments peuvent être créés. En outre, plus la valeur de l'ordre ou du degré est élevée, plus le moment a une faible résistance au bruit. Par conséquence, Mindru et al. proposent de n'utiliser que les moments au premier ordre et au second degré. On peut ensuite utiliser la bonne combinaison [9] de moments et la normaliser pour contrer les changements photométriques. Ces combinaisons s'appellent les moments invariants de couleur [Mindru 2004].

Les moments invariants de couleur peuvent être classifiés en fonction de trois paramètres : l'ordre, le degré et le nombre de bandes de couleur des moments concernés. Il faut noter que le moment d'ordre 0 ne contient pas les informations spatiales, alors que le moment de degré 0 ne contient pas les informations photométriques. Le moment d'ordre 0 est invariant en rotation tandis que les ordres supérieurs ne le sont pas. Dans l'ensemble des moments, les moments d'ordre le plus bas qui fournissent les invariants ont été ajoutés, et ensuite l'ordre est augmenté afin d'élargir l'ensemble des invariants en fonction du besoin. En fonction du type de transformation photométrique (*i.e.* transformation de déformation et de diagonale, transformation d'échelle et de décalage, transformation linéaire suivie d'une translation), la séparation des bandes de couleur est possible. En outre, 2-bandes invariantes sont générées à partir d'1-bande invariante en l'appliquant à chacune des deux bandes de couleur. De manière similaire, la même propriété est vraie pour 2-bandes invariantes, qui fait partie de 3-bandes invariantes. Mindru et al. ont montré les invariants dans les cas de déformations géométriques/photométriques. Pour plus de détails sur les moments invariants de couleur dans chaque cas, voir [Mindru 2004].

3.2.3 Détecteur de Harris-Laplace

Le détecteur de Harris-Laplace [Mikolajczyk 2001] est invariant à l'échelle et est robuste à la transformation sans changement de point de vue (*i.e.* le décalage ou la transformation linéaire suivie d'une translation. Il se compose de deux étapes :

– La détection de points à différentes échelles en combinant la trace et le déterminant

9. Mindru et al. montrent les approches permettent de choisir les bonnes combinaisons dans [Mindru 2004], page 9 - 14.

de la matrice des moments du deuxième ordre.
– La sélection itérative de l'échelle et de l'emplacement jusqu'à la convergence vers le même point d'intérêt.

Dans un premier temps, le détecteur de Harris et Stephens [Harris 1988] est utilisé pour déterminer la localisation des points d'intérêts à chaque échelle de l'image. Dans un deuxième temps, le point d'intérêt sélectionné est déterminé en s'appuyant sur la sélection d'échelle tel que proposé par [Lindeberg 1993, Lindeberg 1998]. En général, pour chaque point d'intérêt choisi dans l'étape précédente, un algorithme itératif est appliqué afin de détecter la localisation et l'échelle du point d'intérêt. Les points d'intérêts sont rejetés dans le cas où le résultat du Laplacien de Gaussienne ne possède aucun extremum, ou dans le cas où ce résultat est inférieur à un seuil [Mikolajczyk 2004].

3.3 Construction d'un dictionnaire des mots visuels

Chaque descripteur de point d'intérêt peut être considéré comme un mot visuel. Néanmoins, le nombre de mots visuels est trop important et rend difficile les traitements sur cet ensemble (*i.e.* le temps de calcul, ou la précision du résultat obtenu). De plus, le nombre de points d'intérêts est différent pour chaque image. Certaines méthodes de classification ou de regroupement requièrent des vecteurs [10] de dimension fixe comme entrée. De ce fait, l'utilisation de vecteurs de taille différente pour ces méthodes est impossible. Pour ces raisons, l'idée de regrouper les descripteurs locaux des points d'intérêts dans des groupes dont la structure interne peut être négligée ou paramétrée apporte une solution. Une méthode de regroupement est donc utilisée pour regrouper les descripteurs dans des régions informatives dont les structures internes sont distantes entre elles. Chaque région correspond à un cluster et est appelée un **mot visuel**. L'ensemble des mots visuels s'appelle un **dictionnaire** ou un **vocabulaire visuel**.

Une des méthodes les plus utilisées pour construire le dictionnaire (vocabulaire) est la méthode de regroupement K-moyennes (K-means) car cette méthode permet de fixer la taille du dictionnaire désiré. La méthode de regroupement K-moyennes classique ne garantit ni l'optimum global, ni un temps de calcul polynomial [Arthur 2009]. Étant donné un ensemble $x_1, ..., x_N \in R^D$ de N descripteurs, K-means cherche K vecteurs $\mu_1, ..., \mu_K \in R^D$ et la distance entre chaque descripteur et les clusters μ_i, $q_1, ..., q_N \in \{1, ..., K\}$ telle que l'erreur d'approximation cumulée $\sum_{i=1}^{N} \|x_i - \mu_{q_i}\|^2$ soit minimisée. Avec l'utilisation d'heuristiques telles que l'algorithme de Lloyd [Lloyd 1982], la méthode K-moyennes est

10. L'ensemble de mots visuels d'une image est le vecteur de mots visuels.

assez facile à mettre en œuvre et à appliquer. Elle cherche les meilleures moyennes de groupes (clusters) $\mu_k = avg\{x_i : q_i = k\}$ et elle cherche aussi le meilleur groupe donnant la plus petite distance entre le descripteur et les groupes $q_{ki} = argmin_k \|x_i - \mu_k\|^2$. Pour les grandes bases de données, une version approximée de la version de Lloyd (Approximate Nearest Neighbour algorithm - ANN) est proposée par Muja [Muja 2009].

Perronnin et al. [Perronnin 2010] proposent une autre méthode de regroupement, le Gaussian Mixture Model clustering (GMM), pour construire le dictionnaire des mots visuels. Le calcul de la distance entre les descripteurs locaux et les groupes s'appuie alors sur la densité de probabilité de x.

3.4 Encodage des caractéristiques d'une image dans un descripteur d'images (sac de mots visuels)

Cette étape encode les descripteurs locaux dans un sac de mots visuels pour la création d'une signature de l'image. L'approche la plus simple est de quantifier la fréquence des descripteurs locaux dans un histogramme. Plusieurs chercheurs ont proposé différents types d'encodage qui permettent de construire le dictionnaire d'une manière souple : par exemple l'encodage de Fisher (Fisher encoding) [Perronnin 2010], le kernel codebook encoding [Van Gemert 2008], le super-vector encoding [Zhou 2010], le locality-constrained linear encoding [Wang 2010] dans le but d'améliorer le modèle par sac de mots visuels.

L'encodage de l'histogramme commence par le calcul de la distance entre un descripteur local et les groupes par une mesure de distance. La distance la plus courte permet de sélectionner le groupe auquel le descripteur de points d'intérêt appartient. L'histogramme de l'ensemble des descripteurs de points d'intérêts est un vecteur non-négatif [11] $f_{hist} \in R^K$. La librairie OpenCV [Itseez 2015] propose l'encodage FLANN-based (Fast Library for Approximate Nearest Neighbors) [12] qui contient une collection d'algorithmes optimisés pour la recherche rapide du plus proche voisin dans les grandes bases de données.

11. Un vecteur non-négatif est un vecteur dont les valeurs sont positives ou nul.
12. http ://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html#flann-based-matcher

3.5 Conclusion

Dans ce chapitre, nous avons présenté quelques méthodes utilisées dans le schéma classique. Dans la suite de ce manuscrit, nous utilisons une chaîne de traitement pour laquelle nous n'avons pas d'application spécifique (*e.g.* la reconnaissance de visage). Nous n'avons ainsi pas de demande spécifique concernant les méthodes de détection, de description, de regroupement et d'encodage. Nous choisissons donc la méthode SIFT [Lowe 2004] pour la détection et la description, la méthode K-moyennes [Lloyd 1982] pour le regroupement et la méthode FLANN-based [Itseez 2014, Muja 2013] pour l'encodage comme présenté dans la figure 5.16 du chapitre 5. La capacité de la méthode SIFT pour trouver des points d'intérêts qui sont invariants à la localisation, la rotation, le changement d'échelle et la robustesse aux transformations affines (distorsions) ainsi qu'aux changements de luminosité en font un bon choix pour notre expérimentation dans cette section. La méthode de regroupement K-moyennes a été choisie car elle permet de fixer la taille du dictionnaire désiré en fixant le nombre de clusters. L'encodage FLANN-based a été choisi car il est optimisé pour les grandes bases de données et des dimensions élevées de l'espace de caractéristiques.

Points clés

Positionnement

- ❑ Nous avons décrit la chaîne de traitement permettant d'obtenir les sacs de mots visuels, de manière traditionnelle ainsi qu'avec des approches plus nouvelles.
- ❑ Nous avons présenté les méthodes liées à l'obtention des sacs de mots visuels. Ces méthodes sont utilisées au chapitre 5.

Réduction de dimension

« Sometimes the questions are complicated and the answers are simple. » "

(Dr. Seuss)

Chapitre 4

Réduction de dimension

Sommaire

4.1 Introduction 85
4.2 Normalisation et binarisation 86
4.2.1 Normalisation 87
4.2.2 Binarisation 90
4.3 Réduction exacte des attributs 91
4.3.1 Introduction 91
4.3.2 Graphe de précédence 93
4.3.3 Algorithme RedAttsSansPerte 97
4.4 Réduction floue des attributs 103
4.4.1 Graphe de précédence flou 104
4.4.2 Algorithme RedAttsfloue 109
4.5 Discussion 120
Points clés 121

4.1 Introduction

L'intérêt de la réduction de dimension en analyse d'images est de réduire au maximum le nombre d'attributs (la dimension) tout en gardant le maximum d'informations perti-

nentes contenues dans les attributs avec plusieurs buts possibles, par exemple pour séparer les observations (objets/images) dans le cadre d'une classification ou d'un regroupement (clustering). Dans la littérature sur le traitement d'images, il existe des méthodes de réduction de dimensions statistiques ou probabilistes ; certaines d'entre elles sont présentées dans le chapitre 1. Néanmoins, à notre connaissance, très peu de méthodes algébriques ont été utilisées pour réduire le nombre d'attributs.

Nous proposons d'étudier une méthode algébrique de l'AFC et d'évaluer la possibilité d'appliquer cette méthode au domaine du traitement d'images. Cette réduction exploite la propriété d'isomorphisme entre le treillis des concepts d'une matrice de données objet/attribut (un contexte) avant et après réduction des attributs. L'ensemble des objets dans les concepts n'est pas modifié par la réduction des attributs. En d'autres termes, les correspondances entre ces objets et les attributs ne changent pas après application de ce traitement de réduction. Nous supposons que la classification ou le regroupement ne devraient pas en être trop affectés.

Nous décrivons dans la section 4.3 l'algorithme de réduction d'attributs s'appuyant sur le graphe de précédence qui s'appelle RedAttsSansPerte. La section 4.4 présente l'extension de cet algorithme au cas flou, en utilisant le graphe de précédence flou. Ces algorithmes peuvent aussi s'appliquer dans le cas de réduction d'images en utilisant la fonction monotone α de la correspondance de Galois. Néanmoins, ces algorithmes ne s'appliquent que pour les données binaires. En conséquent, la section 4.2 introduit les méthodes de normalisation et binarisation utilisées afin d'obtenir des données binaires.

4.2 Normalisation et binarisation

Plusieurs types de données existent dans la littérature (section 1.2 du chapitre 1). Le modèle de sac de mots visuels, avec un "hard-assignment" [1], utilise des données quantitatives discrètes. L'algorithme de réduction nécessite des données binaires. Un pré-traitement de normalisation et de binarisation est donc nécessaire. Il s'agit de transformer les données M_{nm} en contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ où M_{nm} est la matrice de données avec n images et m attributs ($i = [1, n]$, $j = [1, m]$), $\mathcal{O}$ est l'ensemble des images et $\mathcal{A}$ est l'ensemble des attributs. La table 4.1 illustre d'une transformation possible de M_{nm} en $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$.

1. Un "hard-assignment" est le fait qu'un descripteur local soit assigné pour un seul cluster.

	a_1	a_j	a_m
img_1	1	50	0
img_i	10	v_{ij}	7
img_n	99	5	2

(a) Données initiales M_{nm}.

α/β		$\mathcal{A}$		
		a_1	a_j	a_m
$\mathcal{O}$	img_1	x		x
	img_i			x
	img_n	x	x	x

(b) Contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$.

TABLE 4.1: Illustration d'une transformation de M_{nm} en $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$.

4.2.1 Normalisation

Nous avons des données quantitatives discrètes. Nous voulons obtenir des données binaires. Pour faciliter la binarisation, la normalisation des données est nécessaire. Son but est de recaler l'ensemble des valeurs dans un nouvel intervalle de données, par exemple [a-b]. Pour chaque valeur $v_{ij} \in M_{nm}$ d'un attribut a_j ($a_j \in \mathcal{A}$) prise par une image img_i ($img_i \in \mathcal{O}$), v_{ij} est transformée en $v_{ij}^{'}$ selon la formule :

$$v_{ij}^{'} = \frac{v_{ij}(b-a)}{f} + a \tag{4.2.1}$$

où f est un paramètre qui change en fonction du type de normalisation.

Le tableau 4.2 donne un exemple de données initiales. Les tableaux 4.3, 4.4, 4.5 illustrent le résultat après application de différents types de normalisation.

4.2.1.1 Normalisation par ligne (max)

Dans ce travail, nous voulons représenter une relation entre les valeurs normalisées des attributs dans une même image. Ce type de normalisation est appelée **NormLigneMax**.

	a_1	a_2	a_3	a_4	a_5	a_6
img_1	1	50	5	0	0	0
img_2	10	9	1	8	5	0
img_3	99	5	10	1	2	0
img_4	7	0	0	0	7	0

TABLE 4.2: Données initiales.

Pour l'exemple présenté en table 4.2, le résultat de cette normalisation se retrouve dans la table 4.3. Nous utilisons l'intervalle [0,1] et f est la valeur maximum des attributs dans une img_i (une ligne) : $\max_{j=[1,m]}(v_{ij})$. Par conséquent, il n'existe pas réellement de relation entre les valeurs normalisées des images de même attribut. Dans le cas où nous voulons ajouter des images dans l'ensemble des données au cours du traitement, il n'est pas nécessaire de recalculer toutes les valeurs normalisées. Cependant, les valeurs normalisées ne représentent pas l'échelle de rapport du même attribut entre les images dans l'ensemble de données.

$$v_{ij}^{'} = \frac{v_{ij}(b-a)}{\max\limits_{j=[1,m]}(v_{ij})} + a \tag{4.2.2}$$

	a_1	a_2	a_3	a_4	a_5	a_6
img_1	0.02	**1**	0.10	0	0	0
img_2	**1**	0.9	0.1	0.8	0.5	0
img_3	**1**	0.05	0.10	0.01	0.02	0
img_4	**1**	0	0	0	**1**	0

TABLE 4.3: Après normalisation par NormLigneMax.

4.2.1.2 Normalisation par colonne

Dans ce type de normalisation, nous voulons représenter une relation entre les valeurs normalisées des images d'un même attribut. Il est appelé **NormColonne** avec f la valeur maximale de la fréquence pour chaque attribut a_j dans l'ensemble de données : $\max_{i=[1,n]}(v_{ij})$. Avec cette approche, la correspondance entre les images dans la base de données est prise en compte. Pour l'exemple présenté en table 4.2, le résultat de cette normalisation se retrouve dans la table 4.4. L'inconvénient majeur est qu'à chaque insertion d'une nouvelle image dans l'ensemble de données, les valeurs normalisées doivent être recalculées.

$$v_{ij}^{'} = \frac{v_{ij}(b-a)}{\max\limits_{i=[1,n]}(v_{ij})} + a \tag{4.2.3}$$

	$\boldsymbol{a_1}$	$\boldsymbol{a_2}$	$\boldsymbol{a_3}$	$\boldsymbol{a_4}$	$\boldsymbol{a_5}$	$\boldsymbol{a_6}$
$\boldsymbol{img_1}$	0.01	**1**	0.50	0	0	0
$\boldsymbol{img_2}$	0.10	0.18	0.10	**1**	0.71	0
$\boldsymbol{img_3}$	**1**	0.10	**1**	0.13	0.29	0
$\boldsymbol{img_4}$	0.07	0	0	0	**1**	0

TABLE 4.4: Après normalisation par NormColonne.

4.2.1.3 Normalisation par ligne (somme)

Nous proposons la normalisation **NormLigneSomme** qui calcule le ratio entre la fréquence d'un attribut et la somme des fréquences des attributs contenus dans une image (*i.e.* une ligne.). La formule est écrit de la manière suivante :

$$v_{ij}^{'} = \frac{v_{ij}(b-a)}{\sum_{j=1}^{m} v_{ij}} + a \tag{4.2.4}$$

La normalisation NormLigneSomme apporte une relation entre tous les attributs dans une image alors que la normalisation NormLigneMax produit le ratio entre chaque attribut et la valeur maximale des attributs dans une image. NormLigneSomme fournit donc un point de vue plus global de la relation entre les attributs dans une image que la normalisation NormLigneMax. Pour l'exemple présenté en table 4.2, le résultat de cette normalisation se retrouve dans la table 4.5.

	a_1	a_2	a_3	a_4	a_5	a_6
img_1	0.02	0.89	0.09	0	0	0
img_2	0.30	0.27	0.03	0.24	0.15	0
img_3	0.85	0.04	0.08	0.01	0.02	0
img_4	0.5	0	0	0	0.5	0

TABLE 4.5: Après normalisation par NormLigneSomme.

Conclusion

Nous avons commencé par l'utilisation de la normalisation NormLigneMax et NormColonne sur plusieurs ensembles de données différents. Nous avons remarqué que la relation entre les attributs dans une image est plus importante pour effectuer la réduction des attributs que la relation entre les valeurs d'un attribut de plusieurs images. Ensuite, afin de représenter au mieux la relation entre tous les attributs dans une image, nous avons utilisé la normalisation NormLigneSomme.

4.2.2 Binarisation

Une fois les données normalisées, nous pouvons réaliser la binarisation. Cette phase de binarisation se paramètre en fonction d'une fonction de binarisation. Dans le cas le plus simple, cette fonction peut être un seuil de binarisation θ. Avec ce seuil, les données binaires peuvent être obtenues par la formule :

	a_1	a_2	a_3	a_4	a_5	a_6
img_1	0.02	0.89	0.09	0	0	0
img_2	0.30	0.27	0.03	0.24	0.15	0
img_3	0.85	0.04	0.08	0.01	0.02	0
img_4	0.5	0	0	0	0.5	0

(a) Après normalisation par ligne somme.

	a_1	a_2	a_3	a_4	a_5	a_6
img_1	1	1	1	0	0	0
img_2	1	1	1	1	1	0
img_3	1	1	1	1	1	0
img_4	1	0	0	0	1	0

(b) Après binarisation avec le seuil = 0 .

TABLE 4.6: Illustration pour la binarisation.

$$T[i,j] = \begin{cases} 1 & \text{si } v_{i,j} \geqslant \theta \\ 0 & \text{si } v_{i,j} < \theta \end{cases} \tag{4.2.5}$$

L'exemple de binarisation est présenté dans le tableau 4.6.

4.3 Réduction exacte des attributs

4.3.1 Introduction

Comme nous l'avons expliqué en début de ce manuscrit, nous nous intéressons à la réduction des attributs et particulièrement à la notion de contexte attributs-réduits car il s'agit du plus petit ensemble d'attributs garantissant la même structure de treillis de concepts (voir la section 2.2.1 du chapitre 2). Le contexte attributs-réduits contient les attributs irréductibles, qui correspondent aux infimum-irréductibles du treillis de concepts (cf. chapitre 2). En conséquence, la réduction des attributs consiste à calculer les infimum-irréductibles du treillis de concepts.

Une approche naïve pour atteindre ce but est de générer l'ensemble des concepts et/ou son treillis de concepts ou l'ensemble des fermés de $(\alpha \circ \beta, \mathcal{A})$, puis d'en extraire ses irréductibles. Les versions originales des algorithmes de [Bordat 1986, Godin 1995, Lindig 2000, Nourine 1999] génèrent l'ensemble des concepts et son treillis en même temps. Les algorithmes de [Ganter 1984, Fu 2004, Krajca 2008, Andrews 2009] ne génèrent que l'ensemble des concepts. Il suffit ensuite d'ordonner ces concepts pour obtenir le treillis. En général, cette approche est chronophage : les algorithmes de génération d'un treillis des concepts sont exponentiels en fonction de la taille du contexte dans le scénario le plus pessimiste car le nombre de concepts peut être exponentiel. Par exemple, l'algorithme NextClosure proposé par Ganter en 1984 [Ganter 1984] calcule tous les concepts à partir d'un contexte avec une complexité en $O(|\mathcal{O}|, |\mathcal{A}|, |\mathcal{C}|)$ où $\mathcal{O}$ est l'ensemble des objets, $\mathcal{A}$ est l'ensemble des attributs et $\mathcal{C}$ est l'ensemble des concepts formels. Afin d'améliorer le volume de mémoire utilisé dans cet algorithme, Baklouti a proposé une nouvelle version de complexité $O(|\mathcal{O}|^2, |\mathcal{A}|, |\mathcal{C}|)$ [Baklouti 2005]. L'algorithme de Bordat de 1986 [Bordat 1986] et l'algorithme de Lindig [Lindig 2000] construisent le diagramme de Hasse du treillis des concepts en calculant récursivement les successeurs immédiats d'un concept, à partir du bottom du treillis. Ces deux algorithmes ont une complexité en $O(|\mathcal{O}|, |\mathcal{A}|^2, |\mathcal{C}|)$. L'algorithme de [Vychodil 2008] génère le treillis des fermés. Cependant, le coût de génération d'un treillis des fermés est similaire au coût de génération d'un treillis des concepts.

Il est possible de déterminer les attributs irréductibles, sans calculer le treillis, en calculant la fermeture $\alpha \circ \beta$ de chaque attribut, puis en testant si cette fermeture est irréductible ou non. Par exemple, l'algorithme Co-closure utilise la décomposition récursive pour calculer l'ensemble des inf-irréductibles [Colomb 2011], Gély quant à lui, utilise un algorithme générique de type *diviser pour régner* (generic divide and conquer algorithm) pour générer l'ensemble des irréductibles du contexte formel [Gély 2005]. Nous pouvons aussi déterminer les attributs-irréductibles à partir de la sous-hiérarchie de Galois du contexte. Les algorithmes de génération des sous-hiérarchies de Galois (AOC-poset) comme ARES [Dicky 1995], CERES [Leblanc 2000], PLUTON [Berry 2005], HERMES [Berry 2014] génèrent l'ensemble des concepts d'objets et des concepts d'attributs et la sous-hiérarchie de Galois (AOC-poset [2]). Ces algorithmes ont un coût plus faible que le coût de génération d'un treillis des concepts car le nombre de fermetures à calculer correspond au nombre d'objets et d'attributs du contexte. Pour déterminer les infimum-irréductibles du treillis des concepts à partir de son contexte initial, nous proposons de calculer la fermeture de chaque attribut du contexte initial, puis de construire le graphe de précédence [Bertet 2012] qui est proche de l'AC-poset [3]. L'algorithme de réduction supprime les attributs qui ne correspondent pas aux éléments irréductibles, le graphe de précédence ainsi

2. L'AOC-poset est définie dans la définition 2.2.8 du chapitre 2.
3. La similarité entre ces deux graphes a été présentée dans la sous-section 4.3.2.

obtenu ne contient que les infimum-irréductibles. Ce graphe est l'AC-poset [4] du contexte attributs-réduits. Il est ainsi le sous-ordre de l'AC-poset du contexte initial. La définition et les propriétés du graphe de précédence sont présentées dans la sous-section suivante.

4.3.2 Graphe de précédence

Le graphe de précédence se définit pour un système de fermeture de la manière suivante :

α/β	1	2	3	4	5	6	7	8	9
a	x		x		x			x	
b	x	x	x			x	x	x	
c	x	x	x			x	x	x	
d	x	x	x		x	x	x	x	
e	x	x	x			x	x	x	
f	x	x	x		x	x	x	x	
g	x	x	x				x	x	
h		x	x		x	x	x	x	
i		x		x				x	x
j		x		x				x	x

TABLE 4.7: Contexte formel (exemple 2.1 du chapitre 2).

Définition 4.3.1 (graphe de précédence d'un système de fermeture)**.** *Le graphe de précédence du système de fermeture (φ, S) est un graphe orienté $G(S, E)$. Il se compose de l'ensemble fini des sommets S et de l'ensemble des arcs E qui satisfait la condition suivante :*

$$\forall x, y \in S, \exists (x, y) \in E \text{ ssi } \varphi(x) \subseteq \varphi(y) \tag{4.3.1}$$

Propriété 4.3.1. Pour un contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$, nous avons un système de fermeture $(\varphi, \mathcal{A})$ où la fermeture $\varphi = \alpha \circ \beta$. Par conséquent, le graphe de précédence $G(\mathcal{A}, E)$ vérifie $\forall (X, Y) \in \mathcal{A}^2, \varphi(X) \subseteq \varphi(Y) \Leftrightarrow \beta(X) \supseteq \beta(Y)$

Démonstration. Soit le contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ qui contient les concepts $(\alpha(\beta(X)), \beta(X))$, $(\alpha(\beta(Y)), \beta(Y))$ où $X, Y \in \mathcal{A}$.

4. L'AC-poset est définie dans la définition 2.2.9 du chapitre 2.

Reprenons la relation entre deux concepts $(\alpha(\beta(X)), \beta(X))$ et $(\alpha(\beta(Y)), \beta(Y))$:

$$(\alpha(\beta(X)), \beta(X)) \leqslant (\alpha(\beta(Y)), \beta(Y)) \Leftrightarrow \alpha(\beta(X)) \subseteq \alpha(\beta(Y)) \Leftrightarrow \beta(X) \supseteq \beta(Y)$$

On déduit que $\varphi(X) \subseteq \varphi(Y) \Leftrightarrow \beta(X) \supseteq \beta(Y)$.

Le coût de calcul d'une fermeture est important : pour calculer une fermeture $\alpha \circ \beta$ sur un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$, nous devons appliquer la fonction α et la fonction β d'où un coût en $O(|\mathcal{A}| * |\mathcal{O}|)$. Dans notre contexte, le nombre d'objets est communément plus grand que le nombre d'attributs [5]. D'après la propriété 4.3.1, nous pouvons obtenir le graphe de précédence $G(\mathcal{A}, E)$ en utilisant la fonction β au lieu d'utiliser la fermeture d'où un coût en $O(|\mathcal{A}|$. Le graphe de précédence d'un contexte se redéfinit alors par :

Définition 4.3.2 (graphe de précédence d'un contexte)**.** *Pour un contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$, le graphe de précédence $G(\mathcal{A}, E_A)$ est un graphe orienté du système de fermeture $(\alpha \circ \beta, \mathcal{A})$ sur l'ensemble des attributs $\mathcal{A}$ où l'ensemble des arcs E_A satisfait la condition suivante :*

$$\forall x, y \in \mathcal{A}, \exists (x, y) \in E_A \text{ ssi } \beta(x) \supseteq \beta(y) \tag{4.3.2}$$

Exemple : Avec le système de fermeture $(\alpha \circ \beta, \mathcal{A})$ du contexte 4.7, nous avons

$$\varphi(3) = \alpha(\beta(3)) = \{3, 8\}, \beta(3) = \{a, b, c, d, e, f, g, h\}$$
$$\varphi(5) = \alpha(\beta(5)) = \{3, 5, 8\}, \beta(5) = \{a, d, f, h\}$$
$$\varphi(3) \subset \varphi(5) \Leftrightarrow \beta(3) \supset \beta(5) \Rightarrow \text{Il existe un arc } (3, 5) \in E.$$

A noter qu'il est aussi possible de construire le graphe de précédence sur l'ensemble des objets $G(\mathcal{O}, E_O)$ à partir du système de fermeture $(\beta \circ \alpha, \mathcal{O})$ du contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ pour réduire les objets aux sup-irréductibles du treillis des concepts. La relation entre les sommets du graphe de précédence dans ce cas-ci est alors équivalent à $\alpha(x) \subseteq \alpha(y)$. Dans le cas où l'on s'intéresse à la réduction d'objets, nous pouvons aussi construire le graphe de précédence sur l'ensemble des objets $G(\mathcal{O}, E_O)$ à partir du système de fermeture $(\beta \circ \alpha, \mathcal{O})$ du contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ avec la condition $\alpha(x) \supseteq \alpha(y)$. Par exemple, avec le système de fermeture $(\beta \circ \alpha, \mathcal{O})$ du contexte 4.7, nous avons

$$\varphi(b) = \beta(\alpha(b)) = \{b, c, d, e, f\}, \alpha(b) = \{1, 2, 3, 6, 7, 8\}$$
$$\varphi(g) = \beta(\alpha(g)) = \{b, c, d, e, f, g\}, \alpha(g) = \{1, 2, 3, 7, 8\}$$
$$\varphi(b) \subset \varphi(g) \Leftrightarrow \alpha(b) \supset \alpha(g) \Rightarrow \text{Il existe un arc } (b, g) \in E.$$

5. *i.e.* le tableau 5.2 du chapitre 5.

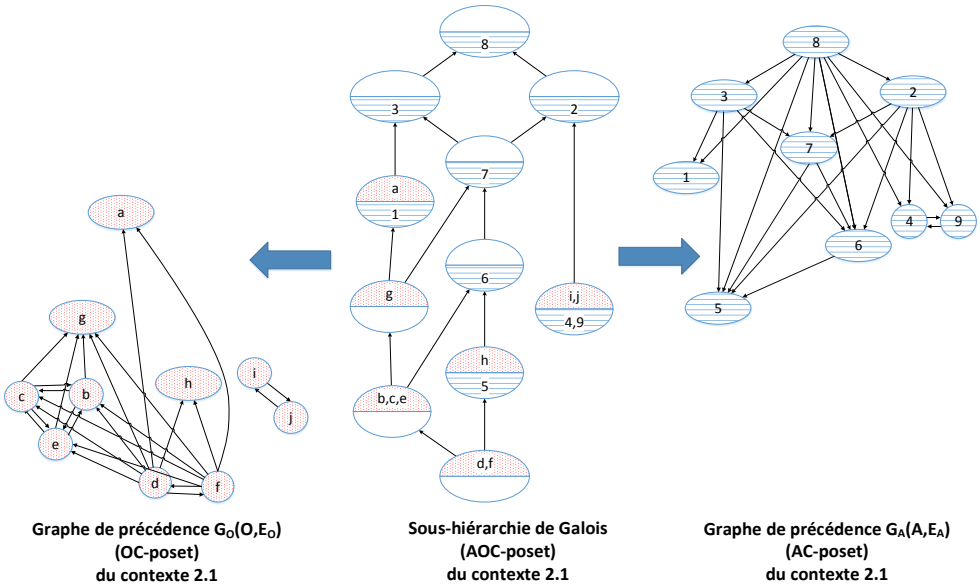


FIGURE 4.1: Sous-hiérarchie de Galois et les graphes de précédences correspondant au contexte 4.7 (contexte 2.1 du chapitre 2).

La figure 4.1 montre le lien entre les graphes de précédence $G_A(\mathcal{A}, E_A)$, $G_O(\mathcal{O}, E_O)$ et la sous-hiérarchie de Galois du contexte 4.7. Nous intéressons à la réduction des attributs, par conséquent, nous ne parlons que du graphe de précédence $G_A(\mathcal{A}, E_A)$ sur l'ensemble des attributs $\mathcal{A}$ du contexte $\mathcal{C} = (\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ dans le reste de ce chapitre. Toutes les propriétés du graphe de précédence $G_A(\mathcal{A}, E_A)$ sont applicables symétriquement sur le graphe de précédence $G_O(\mathcal{O}, E_O)$.

Propriété 4.3.2. La relation binaire sur l'ensemble des sommets $\mathcal{A}$ du graphe de précédence $G_A(\mathcal{A}, E_A)$ vérifie les propriétés suivantes :
- **Réflexivité** : $\forall x \in \mathcal{A}$, on a xRx.
- **Transitivité** : $\forall (x, y, z) \in \mathcal{A}^3$, xRy et yRz impliquent xRz.

Soit $x \in \mathcal{A}$, $y \in \mathcal{A}$ tel que $(x, y) \in E$; P_x et P_y sont respectivement l'ensemble des prédécesseurs de sommet x et de sommet y ; S_x et S_y sont respectivement l'ensemble des successeurs de sommet x et de sommet y. Nous avons les propriétés suivantes :

Propriété 4.3.3.

$$\begin{cases} x \in P_y \\ y \in S_x \\ P_x \subseteq P_y \\ S_y \subseteq S_x \end{cases} \tag{4.3.3}$$

Propriété 4.3.4. Deux attributs x et y sont dits équivalents si les deux arcs (x, y) et (y, x) existent. Dans ce cas :

$$\begin{cases} P_x = P_y \\ S_y = S_x \end{cases} \tag{4.3.4}$$

Propriété 4.3.5. Dans le graphe de précédence $G(\mathcal{A}, E)$, l'ensemble des attributs équivalents forme une clique [6].

Définition 4.3.3 (Graphe de précédence simplifié)**.** *Le graphe de précédence simplifié $G_s(\mathcal{A}', E_A)$ est le graphe de précédence $G_A(\mathcal{A}, E_A)$ où chaque clique est remplacée par un seul élément représentatif.*

Propriété 4.3.6. La relation binaire sur l'ensemble des sommets $\mathcal{A}'$ du graphe de précédence simplifié $G_s(\mathcal{A}', E_A)$ vérifie les propriétés suivantes :
- **Réflexivité** : $\forall x \in \mathcal{A}$, on a xRx.
- **Transitivité** : $\forall (x, y, z) \in \mathcal{A}^3$, xRy et yRz impliquent xRz.
- **Anti-symétrie** : $\forall (x, y) \in \mathcal{A}^2$, xRy et yRx impliquent $x = y$.

C'est donc une relation d'ordre.

Propriété 4.3.7. Le graphe de précédence simplifié $G_s(\mathcal{A}', E_A)$ est isomorphe à l'AC-poset. [7]

Noter que le graphe de précédence simplifié $G_s(\mathcal{A}', E_A)$ correspond au quotient de la relation d'équivalence entre les attributs.

A partir de l'ensemble des attributs du contexte initial, nous voulons obtenir l'ensemble des attributs irréductibles. Le contexte initial contient les attributs irréductibles et les attributs réductibles. Lorsque le contexte est un contexte attributs-réduits, il ne contient que les attributs irréductibles. Le graphe de précédence du contexte initial est défini sur l'ensemble des attributs initiaux alors que le graphe de précédence du contexte d'attributs-réduits est défini sur l'ensemble des attributs irréductibles [8]. Lorsque le contexte est attributs-réduits, son graphe de précédence est son AC-poset. Ces deux graphes sont isomorphes au sous-ordre des infimum-irréductibles du treillis des concepts. Par conséquent, nous cherchons à obtenir l'AC-poset du contexte attributs-réduits à partir du graphe de précédence du contexte initial. Autrement dit, nous voulons supprimer les attributs réductibles du contexte initial.

6. Rappelons qu'**une clique** est un ensemble de sommets C_s tels que $\forall x, y \in C_s$, il existe un arc reliant x et y.

7. *i.e.* la figure 4.2.

8. Un attribut irréductible est un élément infimum-irréductible du treillis des concepts.

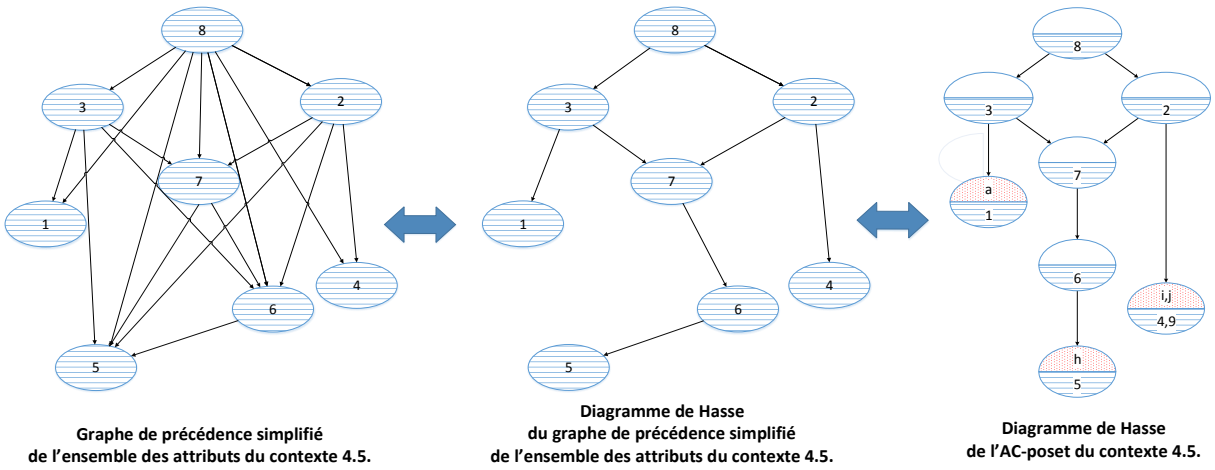


FIGURE 4.2: Isomorphisme entre l'AC-poset et le graphe de précédence sur l'ensemble des attributs correspondant au contexte 4.7 (contexte 2.1 du chapitre 2).

4.3.3 Algorithme RedAttsSansPerte

La figure 4.3 montre le treillis des concepts du contexte 2.1 et le treillis des concepts du contexte attributs-réduits du contexte 2.1. Le théorème 2.2.1 du chapitre 2 établit la propriété d'isomorphisme[9] entre ces deux treillis des concepts. La réduction aux inf-irréductibles permet donc de conserver la description de données et de supprimer les attributs redondants.

L'algorithme **RedAttsSansPerte** (algorithme 1) [Dao 2014] que nous proposons dans cette section considère trois cas, trois étapes selon la taille de l'ensemble des attributs équivalents E_x pour chaque attribut réductible $x \in \mathcal{A}$. E_x est un sous-ensemble des attributs ($E_x \subseteq \mathcal{A}$) correspondant à chaque attribut réductible $x \in \mathcal{A}$ tel que $\beta(E_x) = \beta(x)$.

L'entrée de l'algorithme est un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$. Cet algorithme est composé de la construction du graphe de précédence et trois étapes de réduction selon la cardinalité de l'ensemble des attributs équivalents E_x. La première étape, la clarification ($|E_x| = 1$), est présentée en sous-section 4.3.3.1. La seconde étape, la standardisation ($|E_x| = 0$), est

9. La définition d'isomorphisme du graphe dans l'annexe C.1.

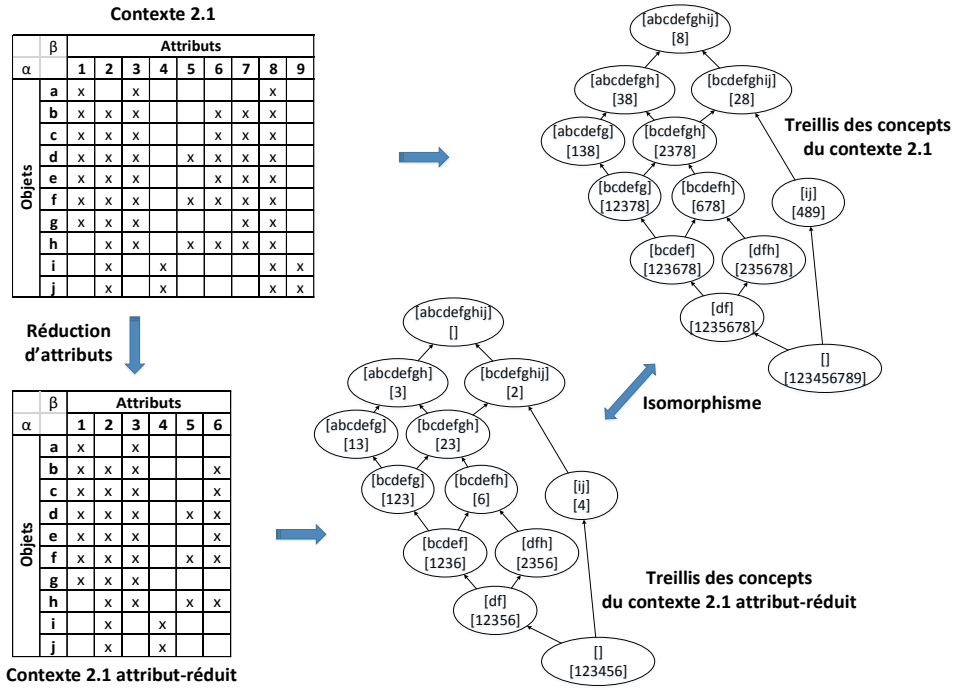


FIGURE 4.3: La relation entre un contexte, son treillis des concepts et son contexte attributs-réduits.

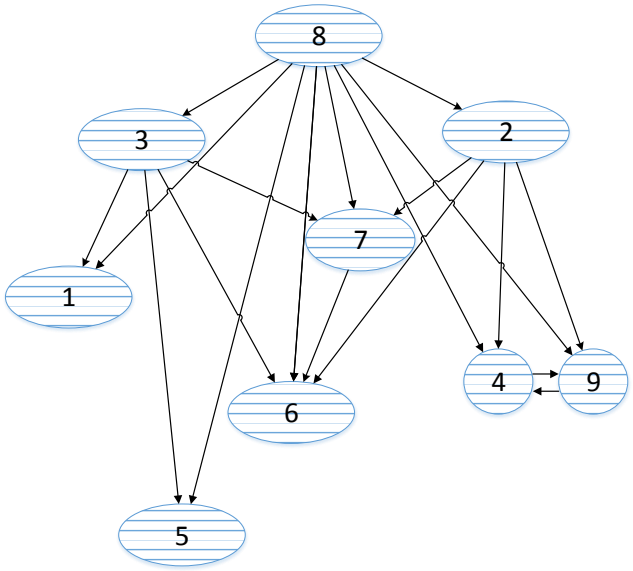


FIGURE 4.4: Graphe de précédences $G_A(\mathcal{A}, E_A)$ correspondant au contexte 4.7.

présentée en sous-section 4.3.3.2. La dernière étape, la réduction ($|E_x| > 1$), est présentée en sous-section 4.3.3.3. La sortie de l'algorithme est l'ensemble des attributs réductibles $X \subset \mathcal{A}$ et l'ensemble des attributs équivalents E_x pour chaque attribut réductible $x \in X$.

4.3.3.1 Clarification exacte

Lorsqu'il existe un attribut x tel qu'il existe un ensemble E_x avec $|E_x| = 1$, ceci signifie que x est équivalent à un attribut y, avec $E_x = \{y\}$. x et y sont des attributs identiques dans le contexte (être présenter dans les même objets), et n'apporte pas d'information supplémentaire lors du traitement de classification ou regroupement d'objets.

Dans le treillis des concepts (ou le treillis des fermés), un concept (ou un fermé) contenant x contient aussi y et vice versa ($\wedge(\alpha \circ \beta)(x) = \wedge(\alpha \circ \beta)(y)$). La suppression d'un de ces attributs ne modifie donc pas la structure du treillis.

Dans le graphe de précédence, des sommets équivalents appartiennent à la même clique C_{clq} car $\beta(x) = \beta(y)$. Il peut exister plusieurs attributs équivalents qui correspondant donc à une clique du graphe. Chaque sommet [10] d'une clique peut être utilisé comme attribut représentatif des autres sommets. La réduction des attributs redondants revient à déterminer les cliques et à éliminer les sommets de chaque clique sauf un élément représentatif. Cependant, la recherche des cliques dans un graphe est un problème NP-complet. Le graphe de précédence a la propriété de transitivité et ses cliques sont donc ses Composantes Fortement Connexes (CFC). Cette propriété nous permet de chercher les CFC du graphe au lieu de chercher les cliques. Nous en gardons alors un arbitrairement et supprimons les autres. Pour simplifier l'algorithme, nous supprimons x du graphe (ajout de x dans l'ensemble des attributs réductibles X) et nous ajoutons l'élément représentatif y dans l'ensemble des attributs équivalents E_x.

Après cette étape, le graphe de précédence devient acyclique. Il correspond au graphe de précédence simplifié et il est isomorphe à l'AC-poset.

Il serait aussi possible de vérifier si $\beta(x) = \beta(y)$ pour chaque nœud x et y simultanément à la construction du graphe de précédence. Mais dans ce cas, la complexité est $O(|\mathcal{A}|^2)$ plutôt que $O(|\mathcal{A}|)$ comme lorsque l'on utilise CFC.

10. Un sommet représente un attribut.

Exemple : Dans le contexte 4.7, les attributs 4 et 9 appartiennent à la même clique étant donné $\beta(4) = \beta(9) = \{i, j\}$. L'attribut 9 est supprimé. L'attribut 4 est conservé comme attribut représentatif.

4.3.3.2 Standardisation exacte ($|E_x| = 0$)

Lorsqu'il existe un attribut x tel que $|E_x| = 0$. Ceci signifie que $E_x = \emptyset$ et x est équivalent à $\emptyset$. Dans le contexte formel, ceci signifie que cet attribut appartient à tous les objets. Il n'y a donc pas d'intérêt à le conserver puisqu'il n'est pas un facteur de différenciation entre les objets et alors son élimination n'influence pas la capacité à catégoriser ou regrouper les objets selon leurs attributs.

Dans le treillis, cet attribut apparaîtra dans tous les concepts (ou fermés) ($\wedge(\alpha \circ \beta)(x) = \wedge(\alpha \circ \beta)(\emptyset)$) et de ce fait, la suppression de cet attribut ne change pas la structure du treillis.

Dans le graphe de précédence, lors de l'étape de clarification, nous avons déjà réduit à une occurrence les attributs d'une même clique. L'attribut x (s'il existe) correspond à la seule source du graphe de précédence (la source du graphe étant le sommet n'ayant que les arcs sortants). En effet, si le graphe possède une seule source x, ceci implique que pour tout $y \in \mathcal{A}$, nous avons $\beta(y)$ strictement inclus dans $\beta(x)$ qui contient tous les objets et $\beta(x) = \beta(\emptyset)$. Nous ajoutons x à X, et $\emptyset$ à E_x.

Exemple : Dans le contexte 4.7, $\beta(8) = \beta(\emptyset) = \{a, b, c, d, e, f, g, h, i, j\}$. L'ensemble β de tous les autres attributs est strictement inclus dans $\beta(8)$ comme nous le montrons dans la figure 4.4. En conclusion, l'attribut 8 est la seule source du graphe de précédence et il est ajouté à X.

4.3.3.3 Réduction exacte ($|E_x| > 1$)

Lorsqu'il existe un attribut x tel qu'il existe E_x avec $|E_x| > 1$, ceci signifie que x est équivalent à la conjonction des attributs de $E_x = \{x_1, x_2, ..., x_n\}$, $n > 1$. Dans un contexte formel, l'extension de l'attribut x est égale à l'intersection des extensions des autres at-

x	$P_x(\|P_x\| > 1)$	$\beta(x)$	$\beta(P_x)$	$\beta(x) = \beta(P_x)$? Réduction ?
7	2,3	b,c,d,e,f,g,h	b,c,d,e,f,g,h	oui
6	2,3,7	b,c,d,e,f,h	b,c,d,e,f,g,h	non

TABLE 4.8: Exemple pour la réduction d'attributs avec le graphe de précédence exact 4.4 (étape de réduction).

tributs $x_1, x_2, ..., x_n$, $n > 1$. Par conséquent, l'attribut x ne contient pas d'information supplémentaire qui aide au regroupement des objets et nous pouvons donc le supprimer.

Dans le treillis des concepts (ou des fermés), l'ensemble des attributs équivalents E_x et l'attribut x apparaissent toujours ensemble dans les concepts (les fermés) ($\wedge(\alpha \circ \beta)(x) = \wedge(\alpha \circ \beta)(E_x)$). Les treillis des fermés avant et après réduction des attributs réductibles sont ainsi isomorphes [Barbut 1970].

Dans le graphe de précédence, l'attribut réductible x qui est équivalent à plusieurs autres attributs est un nœud du graphe qui satisfait la condition $\beta(x) = \beta(P_x)$ où P_x est l'ensemble des prédécesseurs immédiats de x. En conséquence, nous devons vérifier pour l'ensemble des prédécesseurs immédiats P_x de chaque attribut $x \in \mathcal{A}$ s'il satisfait les conditions $|P_x| > 1$ et $\beta(P_x) = \beta(x)$. Si P_x satisfait ces conditions, x est un attribut réductible. Nous ajoutons alors x dans l'ensemble des attributs réductibles X et P_x est ajouté à l'ensemble des attributs équivalents E_x.

Exemple : Dans le contexte 4.7, nous avons $\beta(7) = \beta(2, 3)$ où les sommets $2, 3$ sont les prédécesseurs immédiats du sommet 7. L'attribut 7 est ainsi ajouté à l'ensemble des attributs réductibles X et l'ensemble $\{2, 3\}$ est ajouté à l'ensemble des attributs équivalents E_x. Le tableau 4.8 montre tous les couples (x, E_x) existant dans notre exemple et la satisfaction de la condition permettant de supprimer l'attribut.

Finalement, l'algorithme RedAttsSansPerte complet s'écrit dans l'algorithme 1.

Algorithme 1 : Algorithme de réduction des attributs utilisant le graphe de précédence.

Name : `RedAttsSansPerte.`
Input : un contexte $(O, A, (\alpha, \beta))$
Output : L'ensemble $X \subset A$ des éléments réductibles, et l'ensemble E_x des éléments équivalents pour chaque $x \in X$
initialiser les ensembles $X = \emptyset, E_x = \emptyset$;
initialiser un graphe $G(\mathcal{A}, E)$ avec $\mathcal{A}$ ensemble des sommets;
\\ Construction du graphe de précédence;
foreach $(x, y) \in A \times A$ **do**
  if $\beta(y) \subseteq \beta(x)$ **then** ajouter l'arc (x, y) dans E;
end
\\ Étape (1) : Clarification;
calculer l'ensemble CFC des composantes fortement connexes de G;
foreach $C \in CFC$ **do**
  choisir $y \in C$;
  foreach $x \in C$ *tel que* $x \neq y$ **do**
    ajouter x dans X avec $E_x = \{y\}$; supprimer x à partir du graphe G;
  end
end
soit src l'ensemble des sources du graphe G;
\\ Étape (2) : Standardisation;
if $|src| = 1$ *et* $\beta(src) = \beta(\emptyset)$ **then**
  ajouter src dans X avec $E_x = \{\emptyset\}$; supprimer src à partir du graphe G;
end
\\ Étape (3) : Réduction;
foreach $x \in G$ **do**
  soit P_x l'ensemble des prédécesseurs immédiats x dans le graphe G;
  if $|P_x| > 1$ *et* $\beta(x) = \beta(P_x)$ **then**
    ajouter x dans X avec $E_x = P_x$; supprimer x à partir du graphe G;
  end
end
return X et E_x;

4.3.3.3.1 Complexité de l'algorithme 1

La complexité de l'algorithme 1 se décompose suivant les étapes de l'algorithme : construction du graphe de précédence, clarification, standardisation, réduction.

La complexité de la construction du graphe de précédence inclut le coût d'une génération de fonction monotone $\beta(x)$ pour chaque sommet $x : c_\beta = O(|\mathcal{A}| * |\mathcal{O}|)$. Ensuite, ces valeurs sont comparées en $|\mathcal{A}|^2 * \log|\mathcal{A}|$ pour le calcul des arcs du graphe. Ainsi le coût de calcul total de cette étape est $O(|\mathcal{A}|^2 * (|\mathcal{O}| + \log|\mathcal{A}|))$. L'ensemble des prédécesseurs et l'ensemble des successeurs de chaque attribut sont des informations secondaires qui sont stockées à cette étape afin d'éviter de refaire le calcul dans les étapes ultérieures.

Le coût de la clarification est $O(|\mathcal{A}| + |E|)$. La standardisation parcourt une seule fois tous les sommets du graphe en $O(|\mathcal{A}|)$. La complexité de la réduction est $O(|\mathcal{A}|^2 * |\mathcal{O}|)$.

En conséquence, l'estimation de **la complexité de l'algorithme 1** est $O(|\mathcal{A}|^2 * (|\mathcal{O}| + \log|\mathcal{A}|) + |E|)$.

4.3.3.3.2 Extension à un système de fermeture

L'algorithme **RedAttsSansPerte** (algorithme 1) que nous proposons est un algorithme qui peut s'étendre à tous les systèmes de fermeture (φ, S). Les modifications à faire sont :

- Changement d'entrée de l'algorithme : un système de fermeture (φ, S) au lieu d'un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$.
- Construction du graphe : pour construire le graphe de précédence, il faut utiliser la fermeture φ au lieu de la fonction monotone β.
- Complexité de l'algorithme : la complexité sera calculée en fonction de c_φ où c_φ est le coût d'une génération de fermeture φ du système de fermeture (φ, S). La complexité de l'algorithme 1 pour le système de fermeture (φ, S) est $O(|S|c_\varphi + |S|^2 \log|S| + |E|)$.

Nous retrouvons l'application de l'algorithme générique sur un contexte avec le système de fermeture $(\alpha \circ \beta, \mathcal{A})$ ou le système de fermeture $(\beta \circ \alpha, \mathcal{O})$.

4.4 Réduction floue des attributs

L'algorithme de réduction RedAttsSansPerte exploite l'isomorphisme du treillis des concepts avant et après réduction pour une réduction sans perte d'information, et des taux de classification similaires dans les deux cas (voir la sous-section 5.2.2 du chapitre 5). Cependant, l'obtention d'un taux de réduction significatif n'est pas garanti par ce

traitement car il dépend de la base de données utilisée. L'analyse des facteurs influençant la réduction est détaillée dans le chapitre 5.

Dans le but d'améliorer la réduction des attributs en autorisant une baisse acceptable de la précision des traitements à suivre (*i.e.* classification, clustering), nous proposons une extension floue de notre algorithme : l'algorithme de réduction **RedAttsFloue**. Alors que l'algorithme RedAttsSansPerte supprime des attributs réductibles, l'algorithme RedAttsFloue supprime des attributs irréductibles qui n'apportent pas beaucoup d'information complémentaire, où l'information complémentaire apportée est définie par la similarité entre les extensions de ces attributs [11]. Cet algorithme flou repose sur une extension du graphe de précédence qui permet d'introduire cette notion de similarité entre attributs.

4.4.1 Graphe de précédence flou

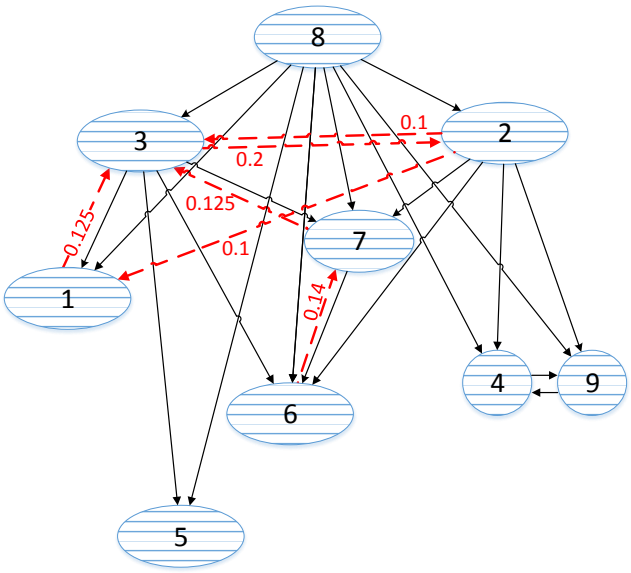


FIGURE 4.5: Graphe de précédence flou $\tilde{G}_A(A, \tilde{E}_A)$ avec $\delta = 0.2$ correspondant au contexte 4.7 (contexte 2.1 du chapitre 2).

Définition 4.4.1 (Graphe de précédence flou d'un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$)**.** *Le graphe de précédence flou* $\tilde{G} = (\mathcal{A}, f)$ *d'un contexte* $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ *est un graphe orienté flou avec la*

11. Une extension d'un attribut est le nombre d'objets qui le partagent.

fonction de flexibilité f définie par :

$$\begin{aligned} f :& \mathcal{A} \times \mathcal{A} \longmapsto [0,1] \\ & (x,y) \longmapsto f(x,y) = \frac{|\beta(y) \backslash \beta(x)|}{|\beta(x) \cup \beta(y)|} \end{aligned} \tag{4.4.1}$$

f reflète la perte d'information du graphe de précédence flou. En effet, f mesure de combien l'extension de y ($\beta(y)$) n'est pas contenue dans l'extension de x ($\beta(x)$), ce qui permet la relaxation de la contrainte d'inclusion de la définition 4.3.2. Quand nous considérons le graphe de précédence flou avec tous les arcs où $f(x,y) \leqslant 1$, nous avons un graphe complet avec 100% de perte d'information. Par ailleurs, lorsque nous limitons le graphe aux arcs où $f(x,y) = 0$, nous retrouvons la relation d'inclusion entre les sommets de ce graphe car $|\beta(y) \backslash \beta(x)| = 0$. Ce graphe correspond alors au graphe de précédence dans le cas exact avec 0% de perte d'informations.

En pratique, le graphe de précédence flou est limité par le seuil de flexibilité δ où la valeur de la fonction de flexibilité f est inférieure ou égale au seuil δ. Nous noterons le graphe de précédence flou $\tilde{G} = (\mathcal{A}, f, \delta)$. Ce graphe flou peut aussi se définir par $\tilde{G} = (\mathcal{A}, \tilde{E}, f)$ où $\tilde{E}$ est l'ensemble des arcs du graphe de précédence flou. Les arcs du graphe de précédence flou où $f(x,y) = 0$ sont appelés les arcs exacts. Les arcs du graphe de précédence flou où $f(x,y) > 0$ sont appelés les arcs flous. L'ensemble des prédécesseurs du sommet y est noté $\tilde{P}_y$. L'ensemble des successeurs du sommet x est noté $\tilde{S}_x$.

Exemple : La figure 4.5 montre le graphe de précédence flou du système de fermeture $(\alpha \circ \beta, A)$ du contexte 4.7 avec un seuil de flexibilité $\delta = 0.15$ où les arcs flous sont en pointillés avec la valeur de f et les arcs exacts sont pleins et sans valeur de f (car la valeur de f d'un arc exact est toujours égale à zero). Par exemple, l'arc de l'attribut 2 vers l'attribut 6 est un arc exact car

$$f(2,6) = \frac{|\beta(6) \backslash \beta(2)|}{|\beta(2) \cup \beta(6)|} = \frac{|\{b,c,d,e,f,h\} \backslash \{b,c,d,e,f,g,h,i,j\}|}{|\{b,c,d,e,f,g,h,i,j\} \cup \{b,c,d,e,f,h\}|} = \frac{0}{9} = 0.$$

On vérifie bien que $\beta(6) \subset \beta(2)$. L'arc de l'attribut 2 vers l'attribut 3 est un arc flou avec la valeur de f est égale à 0.1 :

$$f(2,3) = \frac{|\beta(3) \backslash \beta(2)|}{|\beta(2) \cup \beta(3)|} = \frac{|\{a,b,c,d,e,f,g,h\} \backslash \{b,c,d,e,f,g,h,i,j\}|}{|\{b,c,d,e,f,g,h,i,j\} \cup \{a,b,c,d,e,f,g,h\}|} = \frac{1}{10} = 0.1.$$

Propriété 4.4.1. L'équation de la fonction f peut se reécrire :

$$\begin{aligned} f(x,y) = \frac{|\beta(y)\backslash\beta(x)|}{|\beta(x)\cup\beta(y)|} &= \frac{|(\beta(y)\cup\beta(x))\backslash\beta(x)|}{|\beta(x)\cup\beta(y)|} \\ &= \frac{|\beta(y)\cup\beta(x)| - |\beta(x)|}{|\beta(x)\cup\beta(y)|} \text{ puisque } \beta(x)\subseteq\beta(y)\cup\beta(x) \\ &= 1 - \frac{|\beta(x)|}{|\beta(x)\cup\beta(y)|} \end{aligned} \tag{4.4.2}$$

Le graphe de précédence exact est transitif car il est défini par la relation d'inclusion. Cette propriété n'est pas maintenue dans le graphe de précédence flou. Par exemple, dans la figure 4.5, avec un seuil de flexibilité $\delta = 0.15$, il y a un arc flou de l'attribut 6 vers attribut 7 ($f(6,7) = 0.14$) et un arc flou de l'attribut 7 vers l'attribut 3 ($f(7,3) = 0.125$) mais l'arc flou de l'attribut 6 vers l'attribut 3 n'existe pas car le degré de flou $f(6,3)$ de cet arc est 0.25. Cependant, nous avons des propriétés de pseudo-transitivité intéressantes que nous exploitons dans la réduction à la sous-section 4.4.2.1.

Propriété 4.4.2. Dans un graphe de précédence flou $\tilde{G} = (\mathcal{A}, f, \delta)$, nous avons :

$$\forall (x,y)\in\mathcal{A}^2, \delta \geqslant f(x,y) > 0 \Rightarrow \begin{cases} P_x \subseteq \tilde{P}_y \\ S_y \subseteq \tilde{S}_x \end{cases} \tag{4.4.3}$$

Démonstration.

Montrons que $P_x \subseteq \tilde{P}_y$ est vrai.

Soit un arc du sommet x vers le sommet y tel que $0 \leqslant f(x,y) \leqslant \delta$

$$0 \leqslant \frac{|\beta(y)\backslash\beta(x)|}{|\beta(x)\cup\beta(y)|} \leqslant \delta \tag{1}$$

Soit $a \in P_x$ alors il y a un arc de a vers x, et $\beta(x)\subseteq\beta(a)$. On déduit :

$$\begin{aligned} &\Leftrightarrow \backslash\beta(x) \supseteq \backslash\beta(a) \\ &\Leftrightarrow \beta(y)\backslash\beta(x) \supseteq \beta(y)\backslash\beta(a) \\ &\Leftrightarrow |\beta(y)\backslash\beta(x)| \geqslant |\beta(y)\backslash\beta(a)| \end{aligned} \tag{2}$$

On déduit également de $\beta(x) \subseteq \beta(a)$: $\beta(x) \cup \beta(y) \subseteq \beta(a) \cup \beta(y)$

$$\Rightarrow 0 \leqslant |\beta(x) \cup \beta(y)| \leqslant |\beta(a) \cup \beta(y)|$$

$$\Rightarrow \frac{1}{|\beta(x) \cup \beta(y)|} \geqslant \frac{1}{|\beta(a) \cup \beta(y)|} \geqslant 0 \quad (3)$$

Puisque (2) et (3)
$$\Rightarrow \frac{|\beta(y) \backslash \beta(x)|}{|\beta(x) \cup \beta(y)|} \geqslant \frac{|\beta(y) \backslash \beta(a)|}{|\beta(a) \cup \beta(y)|} \geqslant 0 \quad (4)$$

Puisque (1) et (4)
$$\Rightarrow \delta \geqslant \frac{|\beta(y) \backslash \beta(a)|}{|\beta(a) \cup \beta(y)|} \geqslant 0 \quad (5)$$

On en déduit donc qu'il y a un arc de a vers y dans $\tilde{G} = (\mathcal{A}, f, \delta)$. Et donc que $a \in \tilde{P}_y$. Ce qui nous permet de conclure que $P_x \subseteq \tilde{P}_y$. □

Montrons que $S_y \subseteq \tilde{S}_x$ est vrai.

Soit un arc du sommet x au sommet y tel que $0 \leqslant f(x, y) \leqslant \delta$

Avec la propriété 4.4.1, on a $0 \leqslant 1 - \frac{|\beta(x)|}{|\beta(x) \cup \beta(y)|} \leqslant \delta \quad (1)$

Soit $a \in S_y$ alors il y a un arc de y vers a tel que $f_{ya} = 0$, donc on a $\beta(a) \subseteq \beta(y)$

$$\Rightarrow \beta(a) \cup \beta(x) \subseteq \beta(y) \cup \beta(x)$$

$$\Rightarrow |\beta(a) \cup \beta(x)| \leqslant |\beta(y) \cup \beta(x)|$$

$$\Rightarrow \frac{-1}{|\beta(a) \cup \beta(x)|} \leqslant \frac{-1}{|\beta(y) \cup \beta(x)|}$$

$$\Rightarrow \frac{-|\beta(x)|}{|\beta(a) \cup \beta(x)|} \leqslant \frac{-|\beta(x)|}{|\beta(y) \cup \beta(x)|}$$

$$\Rightarrow 0 \leqslant 1 - \frac{|\beta(x)|}{|\beta(a) \cup \beta(x)|} \leqslant 1 - \frac{|\beta(x)|}{|\beta(y) \cup \beta(x)|} \quad (2)$$

Puisque (1) et (2)
$$\Rightarrow 0 \leqslant 1 - \frac{|\beta(x)|}{|\beta(a) \cup \beta(x)|} \leqslant \delta \quad (3)$$

On en déduit donc qu'il y a un arc de x vers a. Et donc que $a \in \tilde{S}_x$. Ce qui nous permet de conclure que $S_y \subseteq \tilde{S}_x$. □

Propriété 4.4.3. Dans un graphe de précédence flou $\tilde{G} = (\mathcal{A}, f, \delta)$, nous avons :

$$\forall (x, y) \in \mathcal{A}^2, f(x, y) = 0 \Rightarrow \begin{cases} \tilde{P}_x \subseteq \tilde{P}_y \\ \tilde{S}_y \subseteq \tilde{S}_x \end{cases} \quad (4.4.4)$$

Démonstration.

Montrons que $\tilde{P}_x \subseteq \tilde{P}_y$ est vrai.

Soit $a \in \tilde{P}_x$ alors il y a un arc de a vers x tel que $0 \leqslant f(a,x) \leqslant \delta$

Avec la propriété 4.4.1, on a $\quad 0 \leqslant 1 - \dfrac{|\beta(a)|}{|\beta(x) \cup \beta(a)|} \leqslant \delta \qquad (1)$

Soit un arc du sommet x vers le sommet y tel que $f(x,y) = 0 \Rightarrow \beta(y) \subseteq \beta(x)$

$$\begin{aligned} &\Rightarrow \beta(y) \cup \beta(a) \subseteq \beta(x) \cup \beta(a) \\ &\Rightarrow |\beta(y) \cup \beta(a)| \leqslant |\beta(x) \cup \beta(a)| \qquad (2) \\ &\Rightarrow \frac{-1}{|\beta(y) \cup \beta(a)|} \leqslant \frac{-1}{|\beta(x) \cup \beta(a)|} \end{aligned}$$

$$\begin{aligned} &\Rightarrow \frac{-|\beta(a)|}{|\beta(y) \cup \beta(a)|} \leqslant \frac{-|\beta(a)|}{|\beta(x) \cup \beta(a)|} \\ &\Rightarrow 1 - \frac{|\beta(a)|}{|\beta(y) \cup \beta(a)|} \leqslant 1 - \frac{|\beta(a)|}{|\beta(x) \cup \beta(a)|} \qquad (2) \end{aligned}$$

$$\text{Puisque (1) et (2)} \Rightarrow 0 \leqslant 1 - \frac{|\beta(a)|}{|\beta(a) \cup \beta(y)|} \leqslant \delta \qquad (3)$$

On en déduit donc qu'il y a un arc de a vers y. Et donc que $a \in \tilde{P}_y$. Ce qui nous permet de conclure que $\tilde{P}_x \subseteq \tilde{P}_y$. □

Montrons que $\tilde{S}_y \subseteq \tilde{S}_x$ est vrai.

Soit $a \in \tilde{S}_y$ alors il y a un arc de y vers a, $\quad$ donc $0 \leqslant f(a,y) \leqslant \delta$

$$\Rightarrow 0 \leqslant \frac{|\beta(a) \backslash \beta(y)|}{|\beta(y) \cup \beta(a)|} \leqslant \delta \qquad (1)$$

Soit un arc de sommet x vers sommet y tel que $f(x,y) = 0 \Rightarrow \beta(y) \subseteq \beta(x)$

$$\begin{aligned} &\Leftrightarrow \backslash\beta(y) \supseteq \backslash\beta(x) \\ &\Leftrightarrow \beta(a) \backslash \beta(y) \supseteq \beta(a) \backslash \beta(x) \\ &\Leftrightarrow |\beta(a) \backslash \beta(y)| \geqslant |\beta(a) \backslash \beta(x)| \qquad (2) \end{aligned}$$

$\beta(y) \subseteq \beta(x)$ $\Rightarrow \beta(y) \cup \beta(a) \subseteq \beta(x) \cup \beta(a)$

$\Rightarrow 0 \leqslant |\beta(y) \cup \beta(a)| \leqslant |\beta(x) \cup \beta(a)|$

$$\Rightarrow \frac{1}{|\beta(y) \cup \beta(a)|} \geqslant \frac{1}{|\beta(x) \cup \beta(a)|} \geqslant 0 \quad (3)$$

Puisque (2) et (3)
$$\Rightarrow \frac{|\beta(a) \backslash \beta(y)|}{|\beta(y) \cup \beta(a)|} \geqslant \frac{|\beta(a) \backslash \beta(x)|}{|\beta(x) \cup \beta(a)|} \geqslant 0 \quad (4)$$

Puisque (1) et (4)
$$\Rightarrow \delta \geqslant \frac{|\beta(a) \backslash \beta(x)|}{|\beta(x) \cup \beta(a)|} \geqslant 0 \quad (5)$$

On en déduit donc qu'il y a un arc de x vers a. Et donc que $a \in \tilde{S}_x$. Ce qui nous permet de conclure que $\tilde{S}_y \subseteq \tilde{S}_x$. □

4.4.2 Algorithme RedAttsfloue

L'algorithme RedAttsSansPerte ne s'étend pas directement au cas flou. Une adaptation des traitements des étapes de clarification et de réduction est nécessaire. En revanche, l'étape de standardisation flou est dispensable car elle ne dépend pas des arcs flous mais seulement des sources du graphe de précédence flou.

4.4.2.1 Clarification floue

Rappelons que dans le cas exact, la clarification consiste à remplacer chaque clique[12] du graphe par un unique représentant. Plus précisément, nous avons $P_x = P_y$ et $S_x = S_y$ lorsque x et y sont équivalents ($\beta(x) = \beta(y)$) (Propriété 4.3.4 dans la sous-section 4.3.2). Cette propriété est une conséquence directe du fait que le graphe exact est une relation transitive.

Dans le cas flou, si la transitivité était maintenue, cette notion d'équivalence deviendrait $\tilde{P}_x = \tilde{P}_y$ et $\tilde{S}_x = \tilde{S}_y$ lorsqu'il existe un arc de x vers y et un arc de y vers x. Cependant, la propriété de transitivité n'est pas maintenue dans le cas flou. Il s'agit alors de vérifier que la suppression d'un des deux sommets permet de conserver la connexion entre son voisinage immédiat et l'autre sommet. Cette connexion assure au minimum la transitivité

12. Une clique contient un ensemble de sommets C tel que $\forall x, y \in C$, il existe un arc reliant x et y.

entre les voisinages immédiats d'un sommet et l'autre sommet. En d'autres termes, il s'agit de vérifier l'existence d'une telle **"pseudo-transitivité"**, l'existence d'arcs entre un sommet x et les prédécesseurs et successeurs immédiats d'un sommet y ou vice versa, pour supprimer l'attribut qui rapporte peu d'information complémentaire.

Dans le cas où les deux arcs (x, y) et (y, x) sont exacts, $\tilde{P}_x = \tilde{P}_y$ et $\tilde{S}_x = \tilde{S}_y$ car la transitivité entre eux deux est maintenue. Un traitement adapté est nécessaire dans les deux autres cas :

- L'arc (x, y) est exact et l'arc (y, x) est flou. (section 4.4.2.1.1)
- Les arcs (x, y) et (y, x) sont flous. (section 4.4.2.1.2)

Dans ces deux cas, il s'agit de déterminer si un des deux attributs peut être supprimé tout en maintenant la connexion entre l'autre attribut et le voisinage immédiat de l'attribut supprimé (*i.e.* assouplir l'égalité à une inclusion entre $\tilde{P}_x \subseteq \tilde{P}_y$ et $\tilde{S}_x \subseteq \tilde{S}_y$ ou $\tilde{P}_y \subseteq \tilde{P}_x$ et $\tilde{S}_y \subseteq \tilde{S}_x$). Lorsque les deux attributs peuvent être supprimés (c'est-à-dire $\tilde{P}_x = \tilde{P}_y$ et $\tilde{S}_x = \tilde{S}_y$), nous supprimerons celui qui a la valeur de flexibilité maximum

$$max(\gamma_x, \gamma_y) \tag{4.4.5}$$

Où γ_x est la valeur de flexibilité d'un sommet x, calculée par la formule suivante qui prend en compte les valeurs de flexibilité des successeurs et prédécesseurs de x et y :

$$\gamma_x = \Sigma_{p_y \in P_y}^{s_y \in S_y}(f(x, s_y) + f(p_y, x)) \tag{4.4.6}$$

4.4.2.1.1 Le cas où l'arc (x,y) est exact et l'arc (y,x) est flou

L'arc (x, y) est exact, donc, par la propriété 4.4.3, nous avons $\tilde{P}_x \subseteq \tilde{P}_y$ et $\tilde{S}_y \subseteq \tilde{S}_x$. La figure 4.6 illustre toutes les relations possibles entre x et y et leurs prédécesseurs et leurs successeurs.

Afin de déterminer si un des deux attributs peut être supprimé, nous devons vérifier si $\tilde{P}_y \subseteq \tilde{P}_x$ ou $\tilde{S}_x \subseteq \tilde{S}_y$. Quatre situations peuvent se produire :

a) $\boldsymbol{\tilde{P}_y \subseteq \tilde{P}_x}$ **et** $\boldsymbol{\tilde{S}_x \subseteq \tilde{S}_y}$. Dans ce cas, il existe des arcs flous entre x et les prédécesseurs (successeurs) de y et réciproquement. Autrement dit, les deux attributs x, y sont équivalents dans le cas flou. Nous pouvons donc éliminer un des deux. Nous calculons la valeur de flexibilité de chaque sommet par la formule 4.4.6 et supprimons celui qui a la valeur maximum (*c.f.* figure 4.7a).

b) $\boldsymbol{\tilde{P}_y \subseteq \tilde{P}_x}$ **et** $\boldsymbol{\tilde{S}_x \not\subseteq \tilde{S}_y}$. Dans ce cas, tous les arcs flous entre x et les prédécesseurs (successeurs) de y existent et il manque au moins un des arcs flous entre y et les

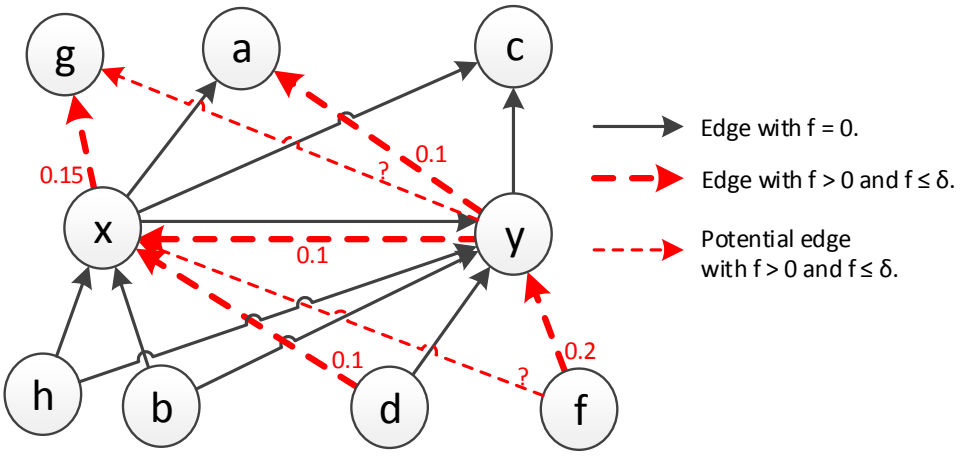


FIGURE 4.6: Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs.

prédécesseurs (successeurs) de x. Dans ce cas, x est donc conservé et y est supprimé afin de conserver la relation entre x et les prédécesseurs (successeurs) de y (*c.f.* figure 4.7b).

c) $\tilde{\boldsymbol{P}}_{\boldsymbol{y}} \not\subseteq \tilde{\boldsymbol{P}}_{\boldsymbol{x}}$ **et** $\tilde{\boldsymbol{S}}_{\boldsymbol{x}} \subseteq \tilde{\boldsymbol{S}}_{\boldsymbol{y}}$. C'est le cas dual du précédent. Dans ce cas, y est donc conservé et x est supprimé (*c.f.* figure 4.7c).

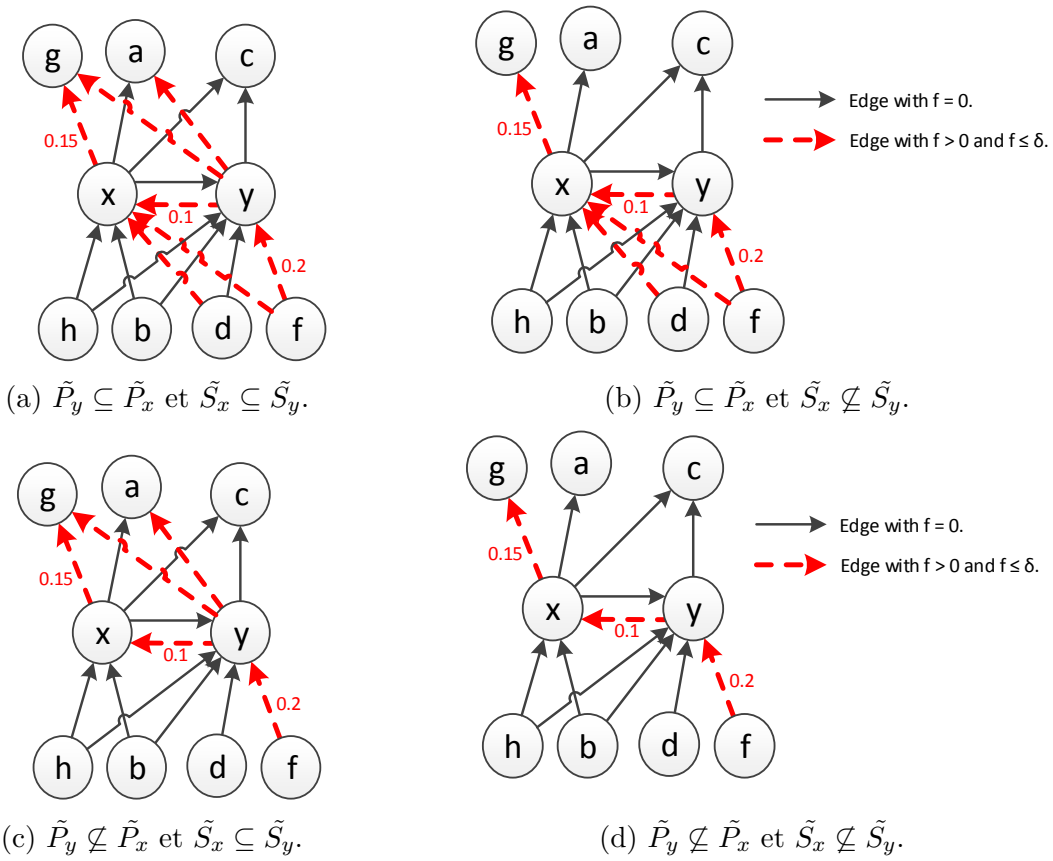
d) $\tilde{\boldsymbol{P}}_{\boldsymbol{y}} \not\subseteq \tilde{\boldsymbol{P}}_{\boldsymbol{x}}$ **et** $\tilde{\boldsymbol{S}}_{\boldsymbol{x}} \not\subseteq \tilde{\boldsymbol{S}}_{\boldsymbol{y}}$. Dans ce cas, il n'existe pas de relation entre x et les prédécesseurs (successeurs) de y et vice versa et les deux sommets x et y sont conservés (*c.f.* figure 4.7d).

4.4.2.1.2 Le cas où les arcs (x,y) et (y,x) sont flous

Lorsque les deux arcs (x, y) et (y, x) sont flous, par la propriété 4.4.2 nous avons $P_x \subseteq \tilde{P}_y$, $S_y \subseteq \tilde{S}_x$ et $P_y \subseteq \tilde{P}_x$, $S_x \subseteq \tilde{S}_y$. La figure 4.8 illustre toutes les relations possibles entre x et y, leurs prédécesseurs et leurs successeurs. Afin de vérifier l'équivalence entre ces deux sommets x et y, nous devons vérifier si $\tilde{P}_x = \tilde{P}_y$ et $\tilde{S}_x = \tilde{S}_y$. Les situations suivantes peuvent se produire :

a) $\tilde{\boldsymbol{P}}_{\boldsymbol{y}} = \tilde{\boldsymbol{P}}_{\boldsymbol{x}}$ **et** $\tilde{\boldsymbol{S}}_{\boldsymbol{x}} = \tilde{\boldsymbol{S}}_{\boldsymbol{y}}$. Dans ce cas, les deux sommets x et y sont équivalents. Nous supprimons le sommet qui a la valeur maximale $max(\gamma_x, \gamma_y)$. Où γ_x (resp. γ_y) est la valeur de flexibilité du sommet x (resp. y) (formule 4.4.6) (*c.f.* figure 4.9a).

b) $\tilde{\boldsymbol{P}}_{\boldsymbol{x}} \subseteq \tilde{\boldsymbol{P}}_{\boldsymbol{y}}$ **et** $\tilde{\boldsymbol{S}}_{\boldsymbol{x}} \subseteq \tilde{\boldsymbol{S}}_{\boldsymbol{y}}$. Dans ce cas, le sommet x est supprimé et le sommet y est conservé afin de retenir la relation entre y et les prédécesseurs flous et successeurs



(a) $\tilde{P}_y \subseteq \tilde{P}_x$ et $\tilde{S}_x \subseteq \tilde{S}_y$.

(b) $\tilde{P}_y \subseteq \tilde{P}_x$ et $\tilde{S}_x \not\subseteq \tilde{S}_y$.

(c) $\tilde{P}_y \not\subseteq \tilde{P}_x$ et $\tilde{S}_x \subseteq \tilde{S}_y$.

(d) $\tilde{P}_y \not\subseteq \tilde{P}_x$ et $\tilde{S}_x \not\subseteq \tilde{S}_y$.

FIGURE 4.7: Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs.

flous de x (*c.f.* figure 4.9b).

c) $\tilde{\boldsymbol{P}}_{\boldsymbol{y}} \subseteq \tilde{\boldsymbol{P}}_{\boldsymbol{x}}$ **et** $\tilde{\boldsymbol{S}}_{\boldsymbol{y}} \subseteq \tilde{\boldsymbol{S}}_{\boldsymbol{x}}$. Dans ce cas, le sommet y est supprimé et le sommet x est conservé afin de retenir la relation entre x et les prédécesseurs flous et successeurs flous de y (*c.f.* figure 4.9c).

d) **Toutes les autres situations**. Dans ce cas, les sommets x et y ne sont pas équivalents dans le cas flou selon la définition présentée et ne sont donc pas supprimés (*c.f.* figure 4.9d).

Afin de vérifier la transitivité locale entre deux sommets x et y tel que les arcs (x, y) et (y, x) existent, nous proposons l'algorithme **VerifTransLocale** (Algorithme 2). L'entrée de cet algorithme est un graphe de précédence flou $\tilde{G}(\mathcal{A}, f, \delta)$. La sortie est l'ensemble $X_t \subset \mathcal{A}$ des éléments supprimés, et l'ensemble E_{t_x} des éléments équivalents pour chaque $x \in X_t$. Pour chaque arc (y, x) du graphe :

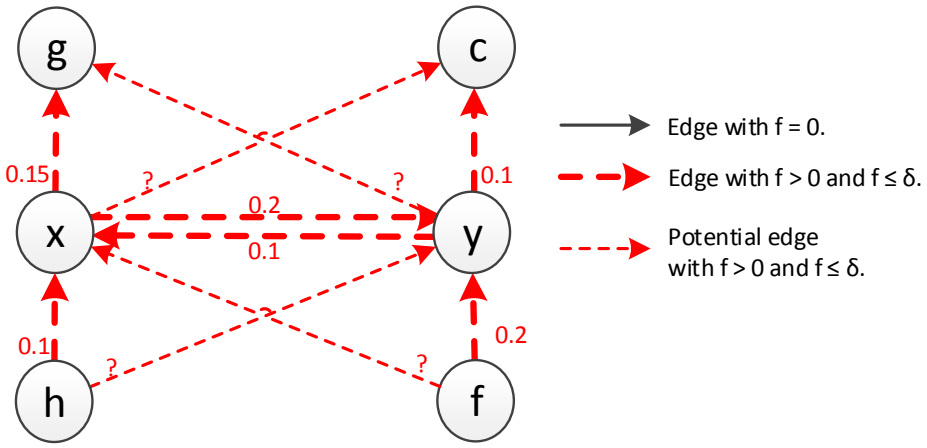
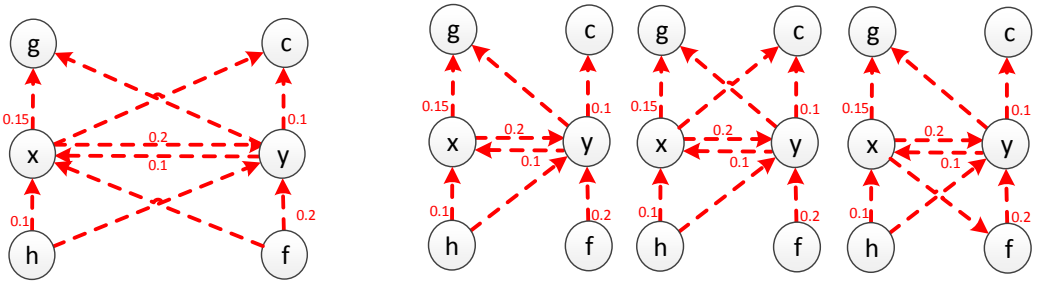


FIGURE 4.8: Relations entre le sommet x et le sommet y, leurs prédécesseurs et leurs successeurs.

- si (x, y) existe et la propriété de transitivité est conservée ($\tilde{P}_x = \tilde{P}_y$ et $\tilde{S}_x = \tilde{S}_y$) : alors nous supprimons l'attribut qui possède la valeur de flexibilité maximum.
- si (x, y) existe et est exact ($f(x, y) = 0$) : alors nous vérifions si les prédécesseurs de x sont les prédécesseurs de y et alors nous supprimons x. S'il s'agit du cas dual, si les successeurs de y sont les successeurs de x alors nous supprimons y.
- si (x, y) existe et est flou ($f(x, y) > 0$) : alors nous vérifions si les prédécesseurs et les successeurs de x sont les prédécesseurs et les successeurs de y. Si cette contrainte est vérifiée alors nous pouvons supprimer x en gardant la transitivité locale entre y et les prédécesseurs, les successeurs de x. De façon similaire, nous supprimons y si $P_y \subseteq P_x$ et $S_y \subseteq S_x$.

4.4.2.2 Réduction floue

Plusieurs approches sont possibles pour effectuer la réduction avec le graphe de précédence flou. Compte tenu de la formulation avec l'opérateur de fermeture, nous pourrions essayer de vérifier si $\beta(x) = \beta(\tilde{P}_x)$. Une autre possibilité serait de vérifier si $f(\tilde{P}_x, x) \leqslant \delta$. Cependant, ces approches peuvent être inappropriées car elles appliquent la fonction de flexibilité à un ensemble $\tilde{P}_x$ qui est déjà flou. Nous avons donc décidé de vérifier si $f(P_x, x) \leqslant \delta$. Ceci est cohérent avec le fait que nous acceptons la même flexibilité sur $\beta(x)$ et $\beta(P_x)$ comme nous l'avons fait lors de la construction du graphe de précédence flou.



(a) $\tilde{P}_y = \tilde{P}_x$ et $\tilde{S}_x = \tilde{S}_y$.

(b) $\tilde{P}_x \subseteq \tilde{P}_y$ et $\tilde{S}_x \subseteq \tilde{S}_y$.

(c) $\tilde{P}_y \subseteq \tilde{P}_x$ et $\tilde{S}_y \subseteq \tilde{S}_x$.

(d) Autres situations.

FIGURE 4.9: Relations entre les sommets x et y, leurs prédécesseurs et leurs successeurs.

Notons que cette approche a l'avantage de ne pas utiliser les arcs flous et par conséquent nous pouvons effectuer cette étape avant l'ajout des arcs flous, et ainsi réduire le nombre de sommets concernés par le calcul.

Algorithme 2 : Vérification de la propriété de pseudo-transitivité entre chaque paire de sommets dans un graphe de précédence flou.

Name : `VerifTransLocale()`.
Input : un graphe de précédence flou $\tilde{G}(\mathcal{A}, f, \delta)$.
Output : l'ensemble $X_t \subset \mathcal{A}$ des éléments réductibles, et l'ensemble E_{t_x} des éléments équivalents pour chaque $x \in X_t$.

begin
  Initialiser $X_t = \emptyset$, $E_{t_x} = \emptyset$;
  foreach $(y, x), \delta \geqslant f(y, x) > 0$ **do**
    Initialiser $red = null$, $eq = null$;
    if $\exists (x, y)$ *et* $x, y \notin X_t$ **then**
      Calculer $\tilde{P}_x, \tilde{P}_y, \tilde{S}_x, \tilde{S}_y$;
      if $\tilde{P}_x = \tilde{P}_y$ *et* $\tilde{S}_x = \tilde{S}_y$ **then**
        Calculer $|\tilde{P}_x|, |\tilde{P}_y|, |\tilde{S}_x|, |\tilde{S}_y|, \gamma_x$ et γ_y;
        if $\gamma_x > \gamma_y$ **then**
          $red = x$; $eq = y$;
        else
          $red = y$; $eq = x$;
        end
      else
        if $f(x, y) = 0$ **then**
          if $\tilde{P}_y \subseteq \tilde{P}_x$ **then** $red = y$; $eq = x$;
          if $\tilde{S}_x \subseteq \tilde{S}_y$ **then** $red = x$; $eq = y$;
        else
          if $\tilde{P}_x \subseteq \tilde{P}_y$ *et* $\tilde{S}_x \subseteq \tilde{S}_y$ **then** $red = x$; $eq = y$;
          if $\tilde{P}_y \subseteq \tilde{P}_x$ *et* $\tilde{S}_y \subseteq \tilde{S}_x$ **then** $red = y$; $eq = x$;
        end
      end
    end
    Ajouter red dans X_t avec $E_{t_x} = \{eq\}$;
    Supprimer red du graphe $\tilde{G}$;
  end
  return X_t et l'ensemble E_{t_x};
end

4.4.2.3 Algorithme RedAttsFloue

L'algorithme **RedAttsFloue** [Dao 2016] proposé dans cette section (Algorithme 4) est un algorithme de réduction de dimension qui prend en entrée un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$. La sortie de l'algorithme est l'ensemble des attributs supprimés $X \subset \mathcal{A}$ et l'ensemble des attributs équivalents E_x pour chaque attribut supprimé $x \in X$.

Cet algorithme est une extension de l'algorithme RedAttsSansPerte. Il se compose des étapes de construction du graphe de précédence (exact et flou - algorithme 3) et de cinq étapes : Les trois premières étapes (clarification, standardisation et réduction) sont les étapes de l'algorithme RedAttsSansPerte. Ensuite, le graphe de précédence flou est construit. Les deux dernières étapes sont la réduction floue et la clarification floue. Dans la dernière étape, nous vérifions la propriété de transitivité locale grâce à l'algorithme VerifTransLocale (algorithme 2).

Étape 1 : Clarification exacte. L'étape 1 de la section 4.3.

Étape 2 : Standardisation exacte. L'étape 2 de la section 4.3.

Étape 3 : Réduction exacte. L'étape 3 de la section 4.3.

Étape 4 : Réduction floue. Pour chaque sommet x, nous vérifions si $f(P_x, x) \leqslant \delta$ avec P_x l'ensemble des prédécesseurs de x. Si c'est le cas, alors x est ajouté dans X et P_x est ajouté dans E_x.

Étape 5 : Clarification floue. Nous utilisons l'algorithme 2 pour vérifier la propriété de pseudo-transitivité entre chaque paire de sommets du graphe de précédence flou $\tilde{G}(\mathcal{A}, f, \delta)$.

Exemple : Dans le contexte 4.7 de ce chapitre (le contexte 2.1 du chapitre 2) :

Étape 1 : L'attribut 4 et 9 appartiennent à la même clique étant donné que $\beta(4) = \beta(9) = \{i, j\}$ donc nous supprimons l'attribut 9.

Étape 2 : Nous avons $\beta(8) = \beta(\emptyset) = \{a, b, c, d, e, f, g, h, i, j\}$. Par conséquent, l'attribut 8 est supprimé.

Étape 3 : Nous avons $\beta(7) = \beta(2, 3)$ où les sommets $2, 3$ sont les prédécesseurs immédiats du sommet 7. Nous supprimons donc l'attribut 7.

Étape 4 : Avec un seuil de flexibilité $\delta = 0.2$ nous avons

$$f(P_6, 6) = \frac{|\beta(6) \backslash \beta(2, 3, 7)|}{|\beta(6) \cup \beta(2, 3, 7)|} = \frac{|-1|}{|7|} = 0.14 \leqslant \delta.$$

Par conséquent, l'attribut 6 est supprimé.

Étape 5 : Avec un seuil de flexibilité $\delta = 0.2$, nous avons la liste des arcs flous ordonnés en ordre croissant selon leur valeur de flexibilité

$$\{f(2,1), f(2,3), f(1,3), f(7,3), f(6,7), f(3,2)\}.$$

Nous vérifions :

- l'arc flou $(2,1)$: il n'existe pas d'arc $(1,2)$, la condition de l'algorithme 2 n'est pas satisfaite donc pas de suppression.
- l'arc flou $(2,3)$: il existe un arc flou $(3,2)$. Nous avons $\tilde{P}_2 = \{2,3,8\}$, $\tilde{P}_3 = \{1,2,3,7,8\}$, $\tilde{S}_2 = \{1,2,3,4,6,7,9\}$, $\tilde{S}_3 = \{1,2,3,5,6,7\}$. $\tilde{P}_2 \subseteq \tilde{P}_3$ et $\tilde{S}_2 \not\subseteq \tilde{S}_3$ donc nous ne pouvons pas supprimer l'attribut 2. De manière similaire, nous ne pouvons pas supprimer l'attribut 3.
- l'arc flou $(1,3)$: il existe un arc exact $(3,1)$. Nous avons $\tilde{P}_1 = \{1,2,3,8\}$, $\tilde{P}_3 = \{1,2,3,7,8\}$, $\tilde{S}_1 = \{3,6\}$, $\tilde{S}_3 = \{1,2,3,5,6,7\}$. Nous avons $\tilde{P}_1 \subset \tilde{P}_3$ donc nous supprimons l'attribut 1.
- l'arc flou $(6,7)$: il existe un arc exact de l'attribut 7 à l'attribut 6. Nous avons $\tilde{P}_6 = \{1,2,3,6,7,8\}$, $\tilde{P}_7 = \{2,3,6,7,8\}$, $\tilde{S}_6 = \{6,7\}$, $\tilde{S}_7 = \{3,6,7\}$. $\tilde{P}_6 \not\subseteq \tilde{P}_7$ et $\tilde{S}_7 \not\subseteq \tilde{S}_6$, la condition de l'algorithme 2 n'est pas satisfaite donc pas de suppression.
- l'arc flou $(7,3)$: il existe un arc exact de l'attribut 3 à l'attribut 7. Nous avons $\tilde{P}_3 = \{1,2,3,7,8\}$, $\tilde{P}_7 = \{2,3,6,7,8\}$, $\tilde{S}_3 = \{1,2,3,5,6,7\}$, $\tilde{S}_7 = \{3,6,7\}$. $\tilde{P}_7 \not\subseteq \tilde{P}_3$ et $\tilde{S}_3 \not\subseteq \tilde{S}_7$, la condition de l'algorithme 2 n'est pas satisfaite donc pas de suppression.

Algorithme 3 : Construction d'un graphe de précédence flou.

Name : `ConstrGraFlou()`.
Input : un contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ et un seuil de flexibilité δ.
Output : un graphe de précédence flou $\tilde{G}(\mathcal{A}, \tilde{E}, f)$.
begin
  initialiser un graphe $\tilde{G}(\mathcal{A}, \tilde{E}, f)$ avec $\mathcal{A}$ l'ensemble des sommets;
  initialiser $\tilde{E} = \emptyset$;
  foreach $(x, y) \in \mathcal{A} \times \mathcal{A}$ *tel que* $x \neq y$ **do**
    Calculer $f(x, y)$;
    if $f(x, y) \leqslant \delta$ **then**
      Ajouter l'arc (x, y) dans $\tilde{E}$;
    end
  end
  return $\tilde{G}(\mathcal{A}, \tilde{E}, f)$;
end

Algorithme 4 : Algorithme de réduction des sommets dans un graphe de précédence flou.

Name : `RedAttsFloue()`.
Input : le contexte $(\mathcal{O}, \mathcal{A}, (\alpha, \beta))$ avec un seuil de flexibilité δ.
Output : l'ensemble $X \subset A$ des éléments réductibles, et l'ensemble E_x des éléments équivalents pour chaque $x \in X$.
Initialiser $X = \emptyset, E_x = \emptyset$;
Construire le graphe de précédence exact $G(V, E)$;
\\ Étape (1) : Clarification;
Calculer l'ensemble CFC des composantes fortement connexes de G;
foreach $C \in CFC$ **do**
  Choisir $y \in C$;
  foreach $x \in C$ *telle que* $x \neq y$ **do**
    Ajouter x dans X avec $E_x = \{y\}$; Supprimer x du graphe G ;
  end
end
\\ Étape (2) : Standardisation;
Soit s l'ensemble des sources du graphe G;
if $|s| = 1$ *et* $\beta(s) = \beta(\emptyset)$ **then**
  Ajouter s dans X avec $E_s = \emptyset$; Supprimer s du graphe G ;
end
\\ Étape (3) : Réduction;
foreach $x \in G$ **do**
  Soit P_x l'ensemble des prédécesseurs immédiats de x dans le graphe G;
  if $|P_x| > 1$ *et* $\beta(x) = \beta(P_x)$ **then**
    Ajouter x dans X avec $E_x = P_x$; Supprimer x du graphe G ;
  end
end
\\ Étape (4) : Réduction floue;
foreach $x \in G$ **do**
  Calculer P_x l'ensemble des prédécesseurs immédiats x dans le graphe G;
  if $|P_x| \neq 1$ *et* $f(P_x, x) \leqslant \delta$ **then**
    Ajouter x dans X avec $E_x = P_x$; Supprimer x du graphe G ;
  end
end
\\ Étape (5) : Standardisation floue;
$\tilde{G}(\mathcal{A}, f, \delta) = \text{constrGraFlou}((\mathcal{O}, \mathcal{A}, (\alpha, \beta)), \delta)$;
$(X_t, E_{t_x}) = \text{verify\_localTransitivity}(\tilde{G}(\mathcal{A}, f, \delta))$;
Ajouter X_t dans X;
Ajouter E_{t_x} dans E_x;
return X et l'ensemble E_x;

L'approche de réduction présentée dans l'algorithme 4 est une extension de l'algorithme 1. Nous pouvons, en termes de réduction, obtenir des résultats équivalents au cas exact, voire supérieurs dans certains cas.

4.4.2.3.1 Complexité de l'algorithme RedAttsFloue (algorithme 4)

La complexité algorithmique de l'algorithme 4 se décompose selon les étapes suivantes : construction du graphe de précédence exact, clarification, standardisation, réduction, réduction floue, construction du graphe de précédence flou et clarification floue.

Le coût de la construction du graphe de précédence exact, de la clarification, de la standardisation et de la réduction est le même que dans le cas exact puisque l'algorithme est le même. La complexité de l'algorithme 4 est donc égale à la complexité de l'algorithme 1 plus le coût de la réduction floue, de la construction du graphe de précédence flou et de la clarification floue.

La complexité de l'algorithme 1 est $O((|\mathcal{A}|^2 * (|\mathcal{O}| + \log|\mathcal{A}|)) + |E|)$.

Ensuite, la complexité de la réduction floue est $O(|\mathcal{A}|^2 * (|\mathcal{O}|)$.

La complexité de la génération du graphe de précédence flou inclut le calcul de la flexibilité $O(|\mathcal{A}|^2 * |\mathcal{O}|^2)$ pour chaque paire d'attributs, d'où une complexité totale de cette étape de $O(|\mathcal{A}|^2 * (|\mathcal{A}|^2 * |\mathcal{O}|^2)) = O(|\mathcal{A}|^4 * |\mathcal{O}|^2)$. Cependant, la valeur de la fonction β de chaque attribut a été calculée lors de l'étape de construction du graphe de précédence exact. Avec une bonne implémentation de l'algorithme, en sauvegardant ces valeurs de β de tous les attributs, le calcul de la flexibilité pour chaque paire d'attributs est $O(|\mathcal{A}|)$. Nous pouvons obtenir un coût plus faible pour cette étape : $O(|\mathcal{A}|^3)$.

Enfin, le coût pour vérifier la pseudo-transitivité entre les sommets dans le graphe de précédence flou est $O(|\tilde{E}|)$ ($\tilde{E}$ est l'ensemble des arcs avec $\delta \geqslant f > 0$ du graphe $\tilde{G}(\mathcal{A}, f, \delta)$), alors la complexité de l'étape de clarification floue (algorithme 2) est $O(|\tilde{E}|)$.

Finalement, **la complexité de l'algorithme** 4 est $O((|\mathcal{A}|^2 * (|\mathcal{O}| + \log|\mathcal{A}| + |A|)) + |E| + |\tilde{E}|)$.

4.5 Discussion

Dans ce chapitre, nous avons d'abord présenté les pré-traitements nécessaires afin d'obtenir les données binaires.

Ensuite, la définition formelle du graphe de précédence et ses propriétés ont été présentées dans la sous-section 4.3.2. Par ailleurs, nous avons montré le lien existant entre ce graphe et l'AC-poset, plus précisément ces deux graphes sont isomorphes quand le graphe de précédence est acyclique.

Puis, nous avons présenté l'algorithme de réduction des attributs RedAttsSansPerte qui garde les attributs irréductibles. Cet algorithme repose sur le théorème fondamental de la théorie des treillis qui stipule que les attributs irréductibles garantissent le maintien de la structure de treillis des concepts, et donc des correspondances maximum d'objets et leurs attributs communs. Il n'y a donc pas de perte d'information. Les attributs "réductibles" sont déterminés à partir du graphe de précédence entre attributs en trois étapes : clarification, standardisation et réduction. Il s'agit là d'une méthode de sélection d'attributs par filtrage et applicable dans le cas de l'apprentissage non supervisé. Elle n'utilise pas le résultat de classification ou clustering comme certaines autres méthodes de sélection des attributs (*c.f.* l'approche par encapsulation ou l'approche par embarquement du chapitre 1).

Cependant, l'obtention d'un taux de réduction significatif dépend des données. Dans le but d'améliorer la réduction des attributs, nous proposons une extension floue de l'algorithme existant. L'algorithme RedAttsFloue va supprimer des attributs irréductibles qui apportent peu d'informations complémentaires. Il repose sur une extension floue du graphe de précédence qui met en relation des attributs "similaires" selon un seuil de flexibilité δ de ce graphe. Nous avons présenté sa définition formelle et ses propriétés dans la sous-section 4.4.1. Cet algorithme (sous-section 4.4.2.3) apporte un taux de réduction plus élevé que l'algorithme original avec cependant une perte d'information. Le niveau de perte d'information dépend du seuil de flexibilité δ. Les expérimentations de ces algorithmes sont présentées dans le chapitre suivant (chapitre 5).

Points clés

Positionnement

- ❏ Nous avons rappelé les méthodes de pré-traitement afin d'obtenir le contexte formel à partir de données numériques.
- ❏ Après avoir défini formellement le graphe de précédence, nous avons présenté ses propriétés et ses liens avec une sous-ordre de sous-hiérarchie de Galois (AC-poset).
- ❏ Nous avons présenté l'algorithme RedAttsSansPerte qui utilise le graphe de précédence comme outil.
- ❏ Après avoir défini formellement le graphe de précédence flou, nous avons proposé une extension de l'algorithme de réduction (l'algorithme RedAttsFloue) qui utilise le graphe de précédence flou comme outil.

Contributions

- ❏ Nous avons présenté le lien entre le graphe de précédence et l'AC-poset.
- ❏ Nous avons défini formellement le graphe de précédence flou et ses propriétés concernant la transitivité.
- ❏ Nous avons proposé l'extension de l'algorithme de réduction (l'algorithme RedAttsFloue), qui est une méthode de sélection d'attributs par filtrage.

Chapitre 5

Expérimentations et évaluations

Sommaire

5.1 Contexte 123
5.1.1 Méthode d'évaluation 124
5.1.2 Description des données 126
5.2 Evaluation de l'algorithme RedAttsSansPerte 132
5.2.1 Point de vue quantitatif 132
5.2.2 Point de vue qualitatif 147
5.3 Evaluation de l'algorithme RedAttsFloue 151
5.3.1 Point de vue quantitatif 151
5.3.2 Point de vue qualitatif 155
5.4 Conclusion 158
Points clés 160

5.1 Contexte

Ce chapitre 5 se propose d'évaluer les performances des algorithmes présentés dans le chapitre 4 et, plus particulièrement, de poser la question de leur pertinence pour une utilisation dans le domaine de la reconnaissance d'images par le contenu (CBIR - Content Based Image Retrieval).

Deux études seront menées pour évaluer les apports des deux contributions majeures de cette thèse :

- Dans un premier temps (section 5.2), nous étudierons l'efficacité de l'algorithme de réduction exacte **RedAttsSansPerte** [1] sur différentes bases de données, incluant des bases d'images et une base du domaine de la sélection de caractéristiques : NIPS 2003 [Guyon 2003].
- Dans un deuxième temps (section 5.3), pour améliorer la capacité de réduction, nous étudierons l'efficacité de l'algorithme de réduction floue **RedAttsFloue** [2] sur la base NIPS 2003.

5.1.1 Méthode d'évaluation

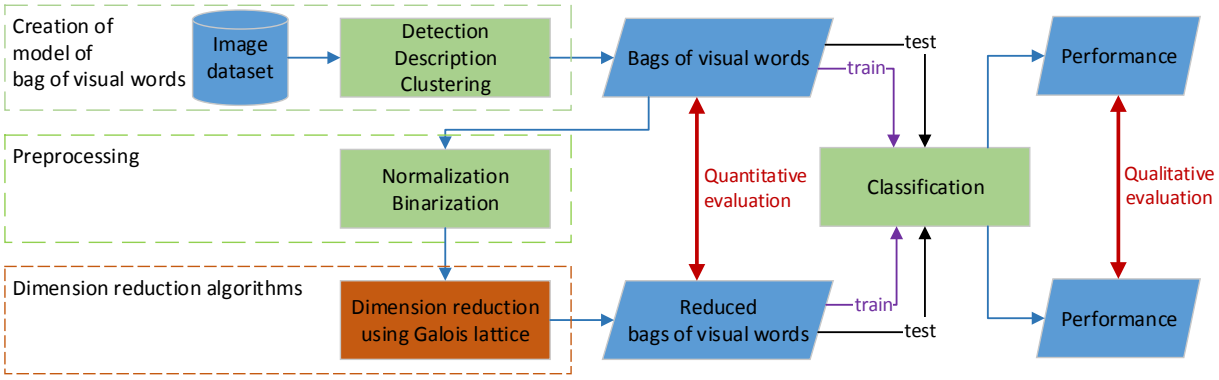


FIGURE 5.1: Chaîne de traitement finale réalisée par nos algorithmes en partant de la base d'images et en allant vers le sac de mots visuels réduit.

La figure 5.1 présente la chaîne de traitement finale réalisée par nos algorithmes en partant de la base d'images et en allant vers le sac de mots visuels réduit. Deux approches sont utilisées pour évaluer le résultat obtenu :

Quantitativement : en comparant le nombre d'attributs réduits par rapport au nombre d'attributs originaux.

Qualitativement : en comparant les performances de classification sur l'ensemble de données original et sur l'ensemble de données réduit.

Cela revient en fait à répondre aux deux questions suivantes :

- Est-ce que l'algorithme réduit effectivement le nombre d'attributs ?

1. Cet algorithme est présenté dans la section 4.3 du chapitre 4.
2. Cet algorithme est présenté dans la section 4.4 du chapitre 4.

– Avec un ensemble d'attributs moins important après l'étape de réduction d'attributs, la classification utilisant cet ensemble d'attributs garde t-elle les mêmes performances ?

Pour répondre à ces questions, nous proposons dans un premier temps de comparer le nombre d'attributs avant et après avoir appliqué nos deux méthodes de réduction en utilisant la mesure Fraction d'Attributs réduits (FA - rapport du nombre d'attributs après réduction sur le nombre d'attributs avant la réduction). Nous proposons aussi de comparer la FA après chaque étape de réduction. L'algorithme RedAttsSansPerte comprend trois étapes : (1) clarification, (2) standardisation et (3) réduction. L'algorithme RedAttsFloue contient cinq étapes : (1) clarification, (2) standardisation, (3) réduction, (4) réduction floue, (5) clarification floue.

$$FA_{ij} = \frac{\Delta N_{att_{ij}}}{N_{attInit_i}} \text{ avec } \Delta N_{att_{ij}} = N_{attInit_i} - N_{attInit_j} \tag{5.1.1}$$

Où $\Delta N_{att_{ij}}$ est le nombre d'attributs réduits entre l'étape i et l'étape $j-1$ tel que $i < j$; $N_{attInit_i}$ est le nombre d'attributs avant la réduction par l'étape i de l'algorithme ; $N_{attInit_j}$ est le nombre d'attributs avant la réduction par l'étape j de l'algorithme. Autrement dit, $N_{attInit_j}$ est le nombre d'attributs après la réduction par l'étape $j-1$ de l'algorithme.

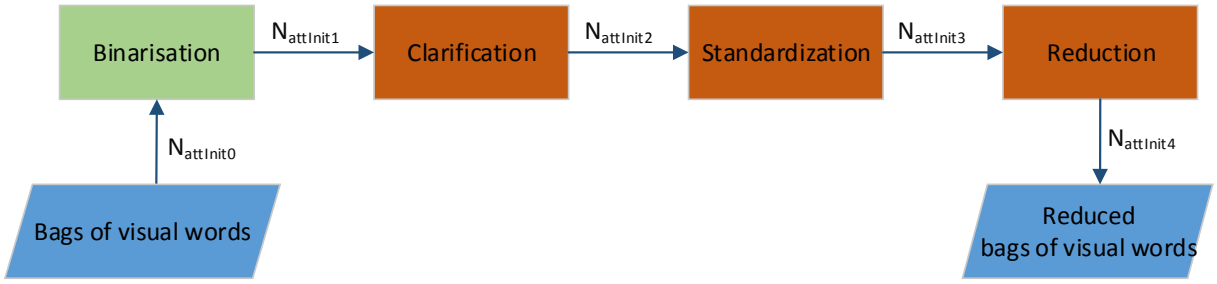


FIGURE 5.2: Chaîne de traitement finale réalisée par l'algorithme RedAttsSansPerte en partant du sac de mots visuels et en allant vers le sac réduit de mots visuels.

Par exemple :

– La Fraction d'Attributs réduits par l'étape **Clarification** de l'algorithme RedAttsSansPerte est $FA_{12} = \frac{N_{attInit_1} - N_{attInit_2}}{N_{attInit_1}}$ (cf. figure 5.2).

– La FA de l'algorithme RedAttsSansPerte est $FA_{14} = \frac{N_{attInit_1} - N_{attInit_4}}{N_{attInit_1}}$.

A la fin de l'étape de binarisation, certains attributs peuvent être nuls pour toutes les observations. Par conséquent, la valeur de FA d'un algorithme (RedAttsSansPerte ou RedAttsFloue) correspond à la proportion entre le nombre d'attributs réduits par l'algorithme et le nombre d'attributs non nuls après la binarisation.

Dans un deuxième temps, pour répondre à la deuxième question, nous nous proposons de comparer les performances de classification sur les bases de données avant et après avoir appliqué notre méthode de réduction. Il existe plusieurs mesures des performances de classification. Certains se calculent à partir de la matrice de confusion tels que les métriques de : F-mesure (F-score), précision, Balanced Error Rate (BER). Le détail de chaque métrique est présenté dans l'annexe A.1. Dans le cas où la classification est équilibrée [3], une des mesures présentées ci-dessus suffit pour représenter le résultat de classification car ces mesures s'appuient sur l'hypothèse que le nombre d'objets dans chaque classe est équivalent.

5.1.2 Description des données

Cette section décrit les différents ensembles de données utilisés pour les expérimentations.

5.1.2.1 Caltech-256

La base Caltech-256 [4] contient 30607 images réparties en 257 classes. 256 de ces classes sont associées à des objets naturels et artificiels présentés dans de bonnes conditions d'éclairage, de "pose", d'arrière-plan, de résolution [Griffin 2007]. La dernière classe de cette base est une classe spéciale, appelée "clutter", permettant de tester le rejet de l'arrière-plan. Le nombre d'images minimum par classe est de 80. Cette base est l'un des jeux de données les plus utilisés dans la communauté de reconnaissance et de détection d'objets.

5.1.2.2 Pascal (VOC)

La base PASCAL [5] est une base de données créée à l'origine pour la compétition "The PASCAL Visual Object Classes (VOC) challenge" [Everingham 2012]. Cette base est de-

3. Le nombre d'images dans chaque classe est équivalent.
4. http ://www.vision.caltech.edu/Image_Datasets/Caltech256/
5. http ://host.robots.ox.ac.uk/pascal/VOC/

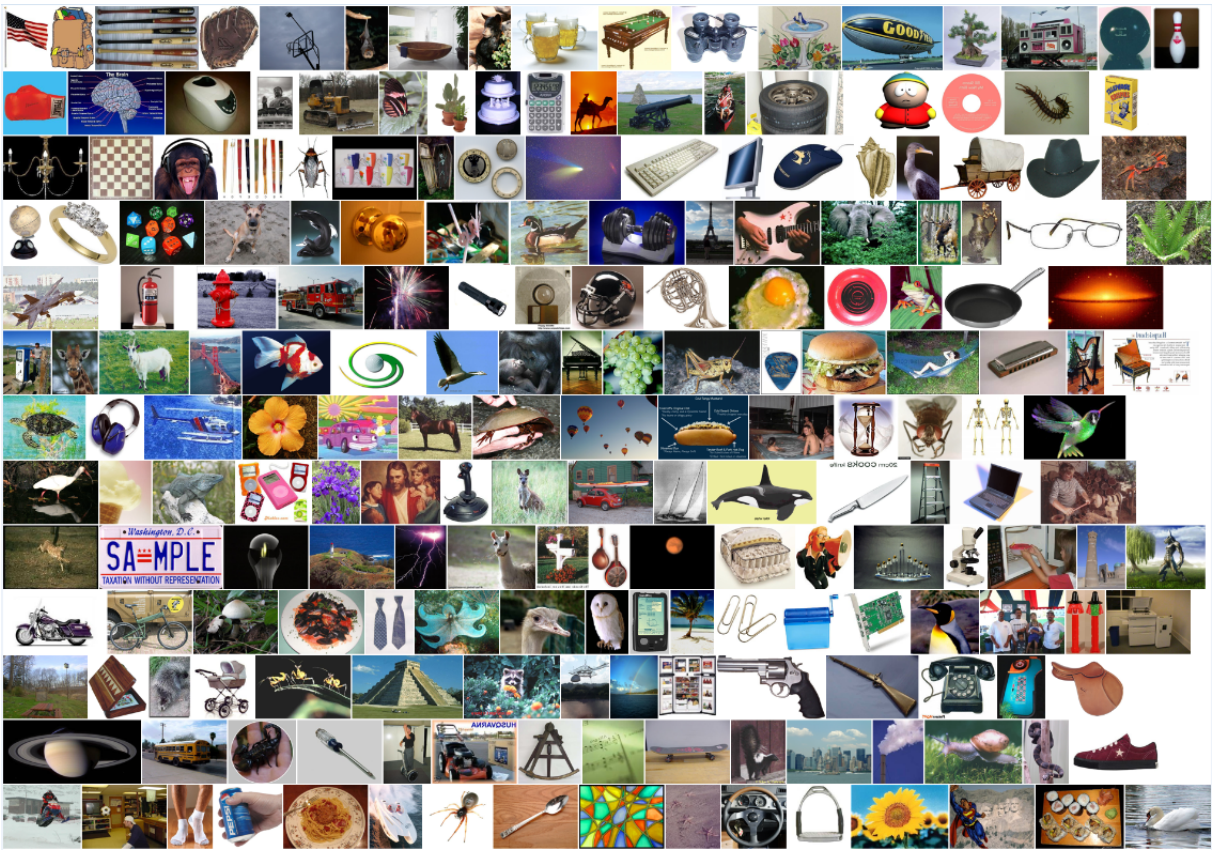


FIGURE 5.3: Base de données CALTECH256.

venue l'une des références dans la communauté de l'apprentissage automatique, plus précisément dans le domaine de la reconnaissance et de la détection d'objets visuels. Plusieurs compétitions VOC ont été organisées par le passé. Pour notre part, nous utilisons la version de 2012 dont la description détaillée est présentée dans [Everingham 2012]. La base VOC2012 contient 22500 images. Chaque image est annotée avec un ou plusieurs labels, correspondant à 20 classes d'objets telles que l'avion, le chat, le vélo, le bateau, la chaise, etc.

5.1.2.3 MIR flickr

Flickr est un site web de partage de photographies et de vidéos gratuites. Les photos mises en ligne sur ce site sont très variées en terme de qualité et de droit d'auteur. Les images de la base MIR Flickr sont sélectionnées en fonction de leurs droits d'auteur, leur

FIGURE 5.4: Base de données VOC2012.

capacité à représenter un domaine donné, et aussi de la qualité visuelle de la photographie. En particulier, cette base est dédiée à l'amélioration de la recherche d'images à partir d'annotations manuelles et de méta-données au format EXIF (Exchangable image file format). Les thèmes généraux associés à ces photos sont le ciel, la mer, le lac, les personnes, la vie des plantes, des arbres, les animaux, les transports, etc. La base de données MIR Flickr [Huiskes 2008] que nous avons utilisée est la version de 2008 avec 25000 images sous licence de droit d'auteur Creative Commons, téléchargée sur le site web Flickr [6].

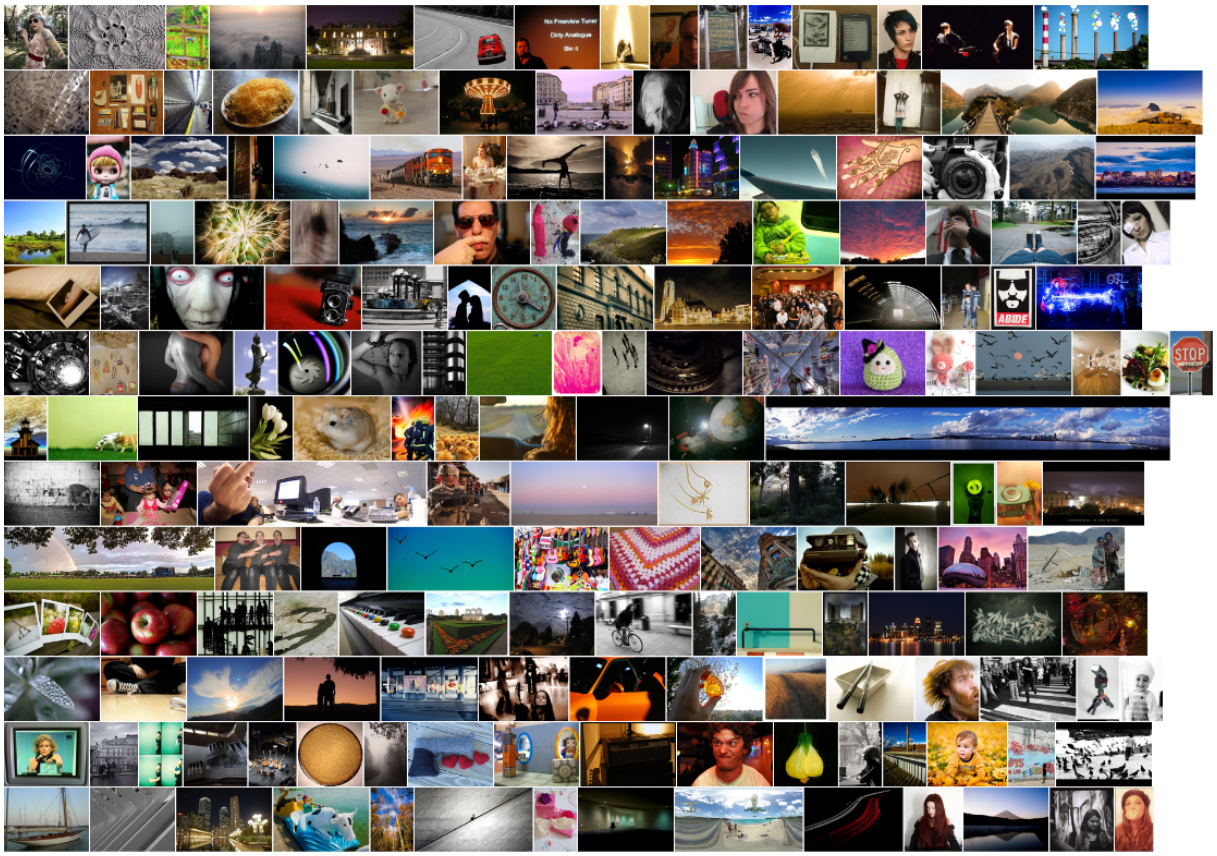


FIGURE 5.5: Base de données MIR fmickr.

5.1.2.4 COREL

La base COREL-5k est une base d'images standard qui contient 5000 images issues de 50 "Corel Stock Photo CDs" [Duygulu 2002]. Chaque CD inclut 100 images du même thème. Chaque image a été annotée avec des mots-clés (de 1 à 5) tels que montagne, piscine, tigre, fumée, etc. Globalement, il existe 371 mot-clés. Cette base est utilisée dans le domaine de l'annotation sémantique, de l'indexation et de la recherche d'images par le contenu.

6. https ://www.flickr.com/

FIGURE 5.6: Base de données COREL.

5.1.2.5 NIPS2003

NIPS2003 est un workshop organisé en 2003 pour mettre en compétition les méthodes de sélection de caractéristiques. Ce workshop a permis de constituer cinq jeux de données : Arcene, Gisette, Dexter, Dorothea et Madelon, qui viennent de différents domaines avec différentes propriétés. Le tableau 5.1 donne, sous forme synthétique, les informations essentielles sur ces jeux de données. Les "probes" sont des attributs supplémentaires bruités ajoutés aux données pour vérifier la performance de l'algorithme de réduction. Les ensembles de données qui ne sont pas binaires, sont préalablement quantifiés entre 0 et 999. Chaque ensemble de données ne contient que deux classes. Dans notre contexte, nous traitons les trois ensembles de données : Arcene, Gisette, Dexter car ils sont représentés sous forme d'histogramme, comme les sacs de mots visuels.

Arcene est une base dont les données sont constituées des spectres de masses obtenus par la technique SELDI (Surface-Enhanced Laser Desorption / Ionization - une méthode

utilisée en médecine). Les attributs ne sont pas des images, mais constituent des histogrammes obtenus à partir de données réelles utilisées en médecine.

Gisette est un sous-ensemble de la base MNIST [LeCun 1998]. Gisette contient des images en noir et blanc des chiffres manuscrits 4 et 9.

Dexter est un sous-ensemble du benchmark de catégorisation de textes de Reuters [Lewis 1997]. Reuters utilise un modèle à base de sacs de mots [Salton 1975] pour représenter les données.

Dorothea est un ensemble de données concernant la recherche de médicaments. L'ensemble de données originales vient du laboratoire DuPont Pharmaceuticals Research et du KDD (Knowledge Discovery in Data Mining) Cup 2001. Cette base est une base binaire. Le détail des attributs n'a pas été fourni pour éviter l'influence d'une connaissance de la nature des caractéristiques *a priori* sur le processus de sélection.

Madelon est une base synthétique contenant 5 attributs informatifs et 15 attributs redondants. Les attributs redondants n'ayant aucune puissance prédictive. L'ordre des attributs a été randomisé. Les exemples sont séparés en deux classes aléatoirement correspondant aux étiquettes +1 et -1.

La description plus détaillée de cette base est disponible dans le rapport de Guyon [Guyon 2003].

Ensemble de données	**Type de données**	**Nombre d'objets**	**Nombre d'attributs**	**Nombre de probes**
Arcene	dense integer	900	10000	3000
Gisette	dense integer	13500	5000	2500
Dexter	sparse integer	2600	20000	10053
Dorothea	sparse binary	19500	100000	50000
Madelon	dense integer	4400	500	480

TABLE 5.1: Tableau de synthèse représentant les jeux des données de la base NIPS 2003.

5.2 Evaluation de l'algorithme RedAttsSansPerte

Dans cette section, l'objectif est de vérifier si la réduction des mots visuels est effective en appliquant l'algorithme de réduction RedAttsSansPerte du chapitre 4. Nous avons calculé la réduction de dimension avec cet algorithme sur plusieurs jeux de données. Nous évaluons les résultats quantitativement dans la sous-section 5.2.1 et qualitativement dans la sous-section 5.2.2.

5.2.1 Point de vue quantitatif

A partir des images, les deux étapes suivantes (*cf.* figure 5.7) sont nécessaires avant appliquer l'algorithme de réduction des mots visuels :

(1) Création du sac de mots visuels.

(2) Normalisation et binarisation (Pre-processing).

Afin d'analyser l'influence de ces étapes sur la capacité de réduction de l'algorithme RedAttsSansPerte, nous divisons la discussion en trois parties :

- La sous-section 5.2.1.1 décrit la capacité de réduction de cet algorithme sur plusieurs bases de données (point de vue global de la réduction).
- La sous-section 5.2.1.2 décrit la capacité de réduction de cet algorithme sur une même base de données en fonction des méthodes de normalisation utilisées (influence de la normalisation).
- La sous-section 5.2.1.3 décrit la capacité de réduction de cet algorithme sur une même base de données en fonction des méthodes de construction des sacs de mots visuels (influence de la détection, de la description et du clustering).

5.2.1.1 Réduction par l'algorithme RedAttsSansPerte

Dans cette partie, nous avons testé l'algorithme RedAttsSansPerte sur les huit bases suivantes : VOC 2005, COREL, mirFlick, VOC 2012, CALTECH256, Arcène, Gisette, Dexter.

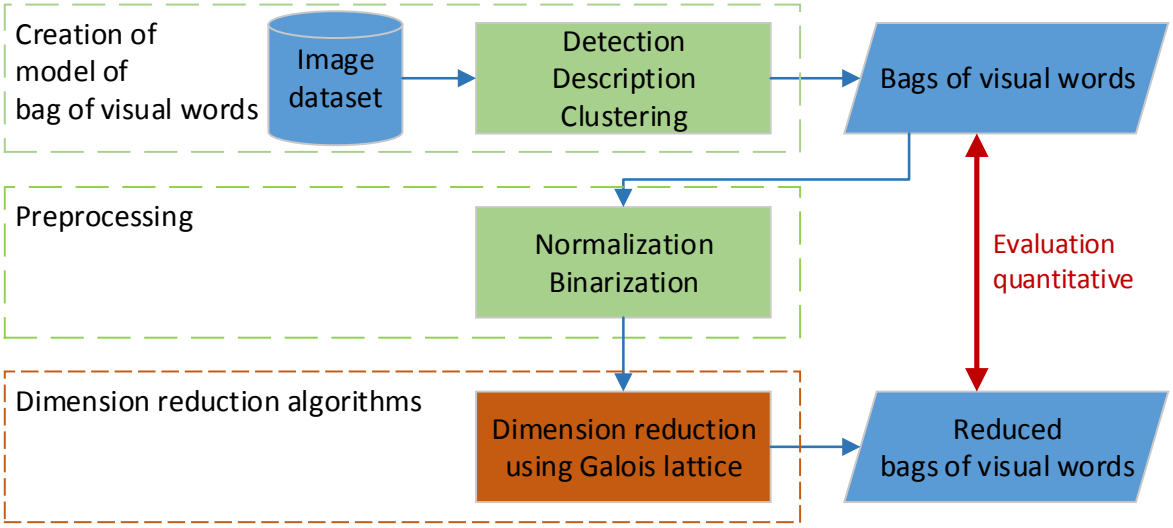


FIGURE 5.7: Chaîne de traitement réalisée par nos algorithmes en partant de la base d'images et en allant vers le sac réduit de mots visuels.

Le tableau 5.2 montre les cinq premières bases de données images et les méthodes utilisées afin d'obtenir les sacs de mots visuels [12]. Ces sacs de mots visuels ne sont pas normalisés. De ce fait, nous appliquons la normalisation NormLigneMax [13] sur ces sacs de mots visuels [14]. Les trois derniers ensembles de données (Arcene, Dexter, Gisette) sont des histogrammes déjà normalisés.

7. Le détecteur Harris-Laplace pour l'extraction des contours et le détecteur Laplacien de Gaussiennes pour la détection des blobs.
8. SIFT (Scale-Invariant Feature Transform) [Lowe 1999].
9. K-moyennes [Lloyd 1982] a été implémenté dans la librairie OpenCV.
10. Harris-Laplace [Mikolajczyk 2004].
11. CMI (Color Moment Invariants) [Mindru 2004].
12. Les sacs de mots visuels de la base VOC2005 et la base COREL sont les signatures précalculées issues de travaux de Dounia AWAD et Nhu Van NGUYEN au sein du laboratoire L3i. Les sacs de mots visuels de la base VOC2012 et la base MIR flickr sont les signatures précalculées issues de travaux de Syntyche GBEHOUNOU au sein du laboratoire XLIM-SIC. Les sacs de mots visuels de la base CALTECH256 sont issus de nos travaux pendant la thèse.
13. Cette méthode est présentée dans la sous-section 4.2.1 du chapitre 4.
14. La raison de ce choix est présentée dans la section 4.2.1 du chapitre 4.

Database	Images nb	Features nb	Detector	Descriptor	Dictionary of visual words
Dataset 1 (PASCAL VOC 2005)	1354	262	Harris-Laplace and Laplacian [7]	SIFT [8]	K-means [9]
PASCAL (VOC 2012)	17124	4096	Harris-Laplace [10]	CMI [11]	Random selection of all key points
MIR flickr	24991	4096	Harris-Laplace	CMI	Random selection of all key points
COREL	4998	500	SIFT [8]	SIFT [8]	K-means [9]
CALTECH 256	30607	500	SIFT [8]	SIFT [8]	K-means [9]

TABLE 5.2: Ensembles de données et méthodes utilisées pour obtenir les sacs de mots visuels. [12]

Étant donné que l'algorithme RedAttsSansPerte nécessite des données binaires en entrée, nous utilisons un seuil de binarisation que nous faisons varier entre [0, 1] avec un intervalle de 0.1 pour étudier l'influence de ce paramètre. Nous voulons aussi étudier s'il est possible d'obtenir un seuil de binarisation optimal automatiquement.

Les figures 5.8, 5.9, 5.10 montrent le comportement global de l'algorithme de réduction RedAttsSansPerte en fonction du seuil de binarisation sur les huit bases de données. Nous distinguons la capacité de réduction de notre algorithme et la capacité de réduction due au seuil de binarisation.

Analyse

Tout d'abord, nous pouvons observer que, quels que soient la base et le seuil, des attributs sont supprimés par l'algorithme de réduction. Pour autant, cette réduction n'est pas uniforme et dépend de la base. Cependant, la courbe de réduction de chaque histogramme a la même forme. En effet, un seuil de binarisation plus élevé entraîne mécaniquement une réduction d'attributs plus importante.

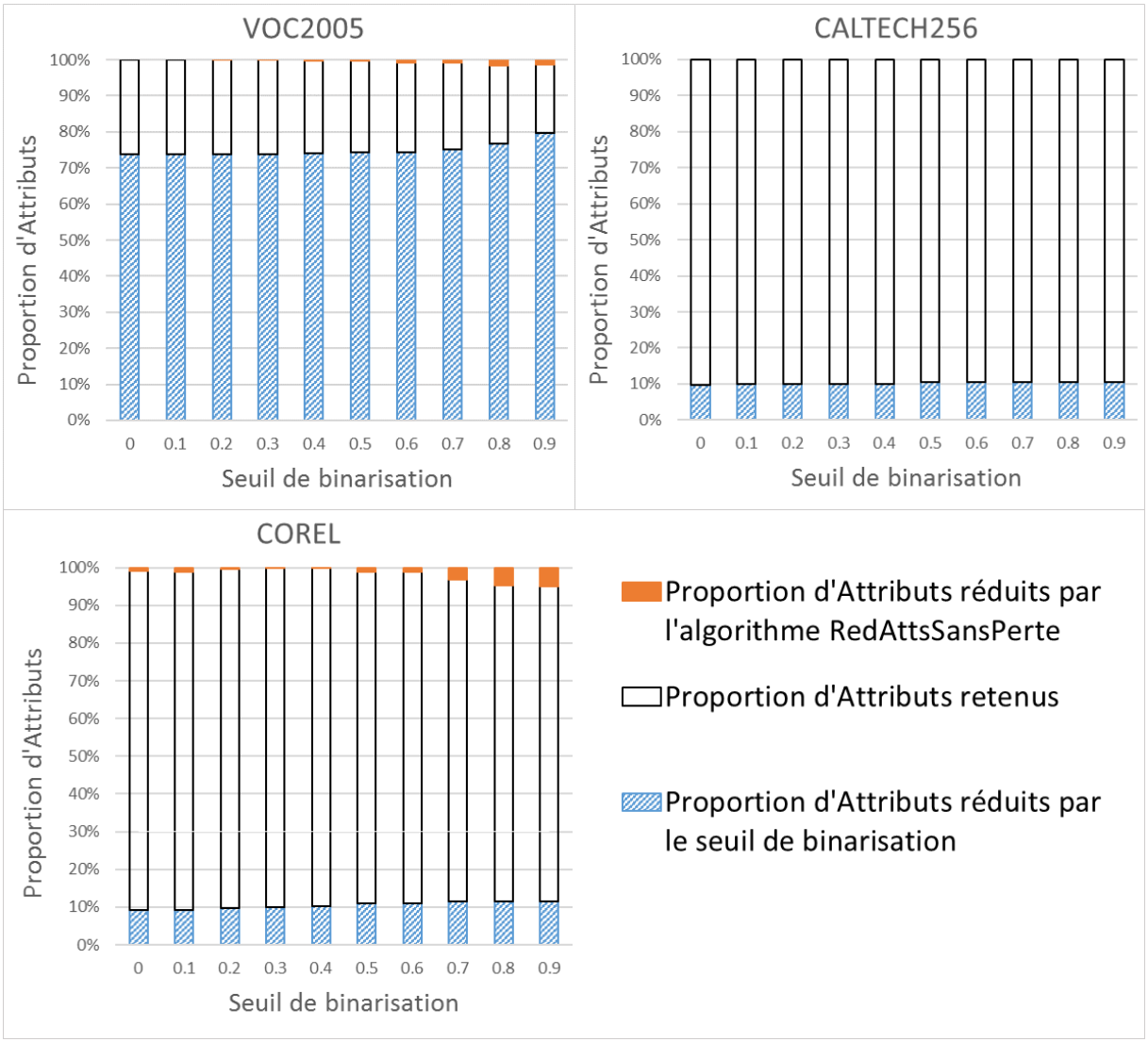


FIGURE 5.8: Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur trois bases de données : VOC2005, CALTECH256 et COREL5k.

Par ailleurs, nous observons que la réduction d'attributs se fait dans sept bases sur huit : Les réductions sur les bases VOC2005, COREL, VOC2012, MIRflickr, Arcene, Dexter, Gisette sont au maximum respectivement de 1.6%, 5%, 11%, 10.5%, 26%, 33%, 41% des attributs réduits par rapport au nombre d'attributs initiaux. Comme nous pouvons le remarquer, et comme nous l'envisagions, le taux de réduction varie en fonction du seuil de

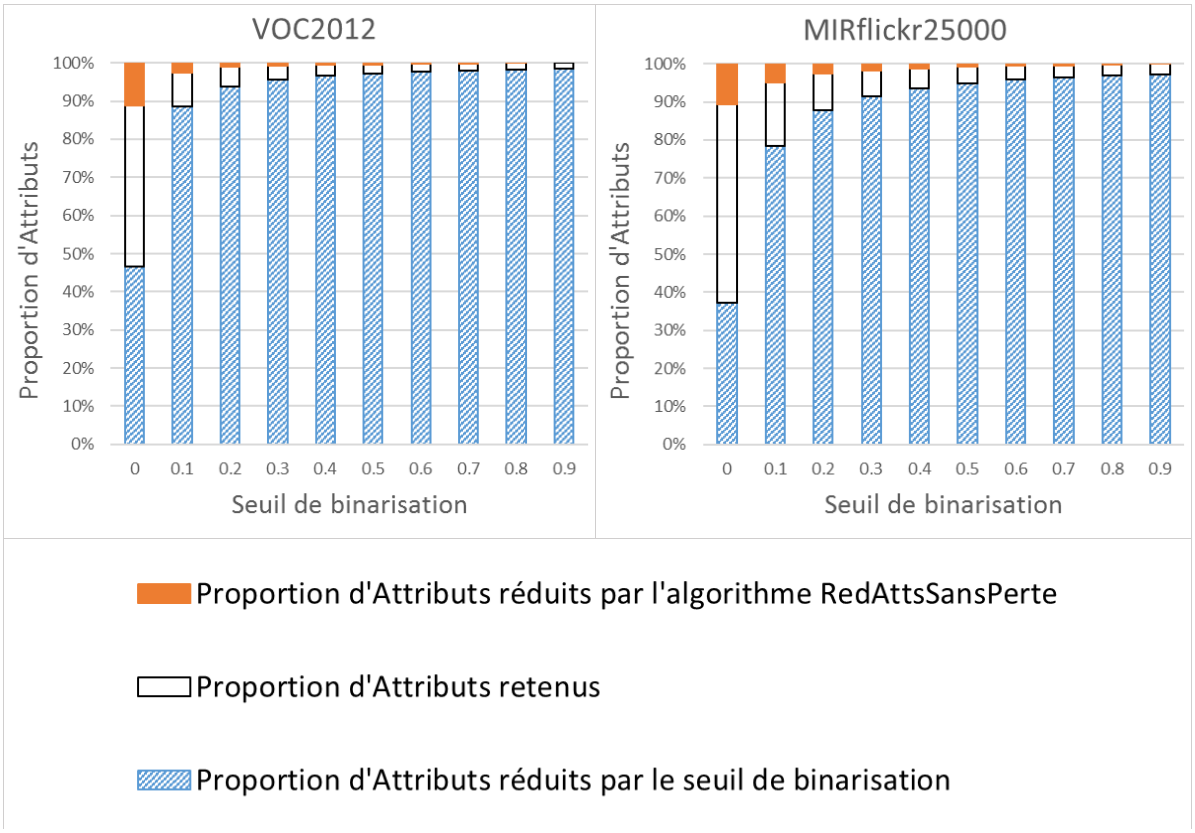


FIGURE 5.9: Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur deux bases de données : VOC2012 et MIRflickr25000.

binarisation. Cependant, pour les trois bases CALTECH256, COREL et VOC2005, le taux de réduction (FA) maximum ne dépasse pas 5% des attributs alors que pour MIRflickr et VOC2012 la réduction est d'environ 11% des attributs. Le taux de réduction dépend donc des données.

Ces comportements peuvent s'expliquer par le fait que différentes méthodes ont été utilisées pour obtenir ces sacs de mots visuels. En effet, les mots visuels des bases VOC2012 et MIRflickr sont obtenus en utilisant une sélection aléatoire des attributs alors que les mots visuels des bases CALTECH256, COREL ou VOC2005 sont obtenus en utilisant un

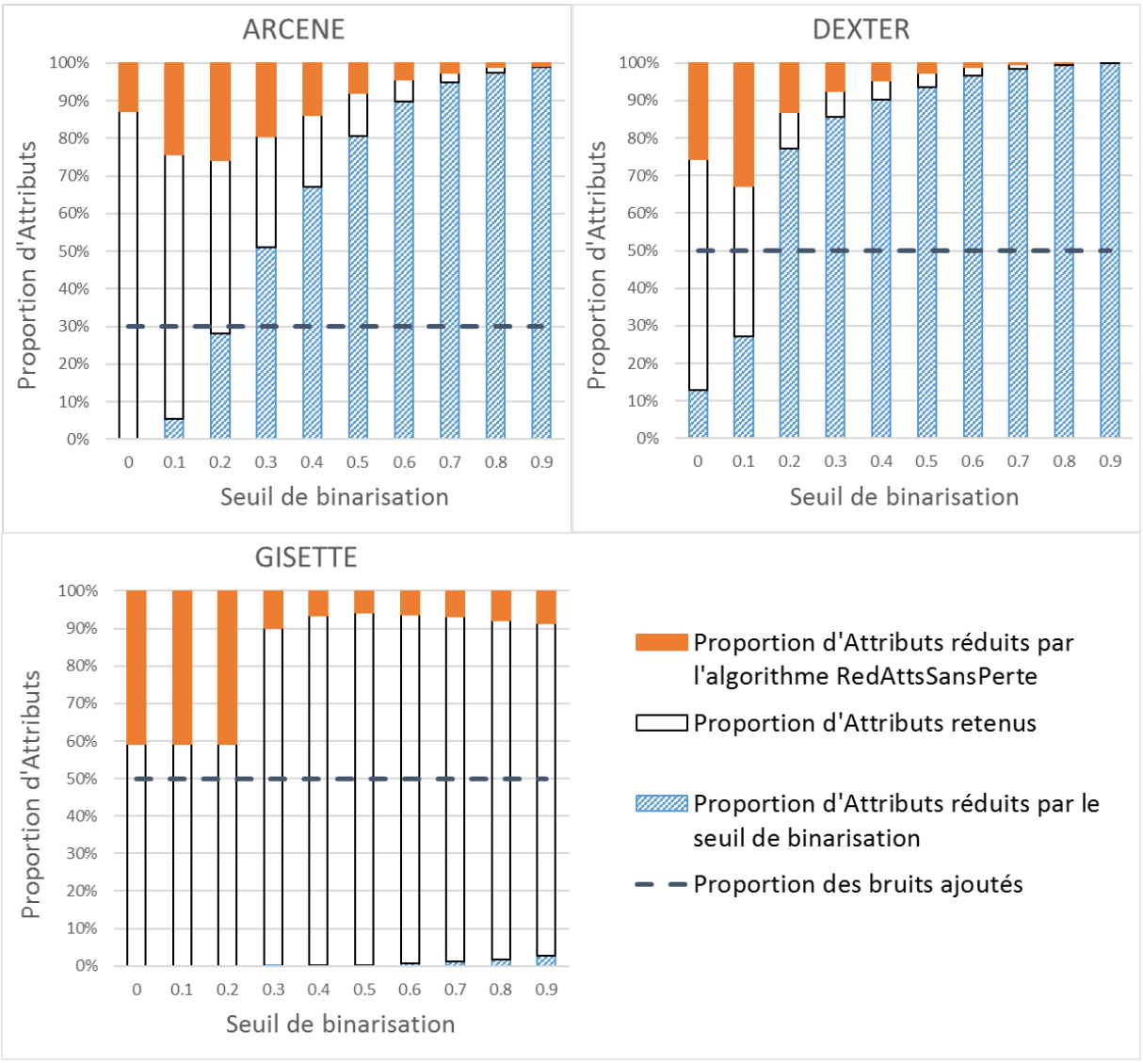


FIGURE 5.10: Rapport entre le nombre d'attributs réduits (obtenus par l'application d'un seuil binaire et par l'application de l'algorithme RedAttsSansPerte) et le nombre d'attributs initiaux sur trois bases de données : Arcene, Dexter, Gisette. La ligne pointillée est le ratio du nombre des bruits ajoutés sur le nombre des attributs initiaux.

algorithme de clustering comme l'algorithme K-moyennes. La corrélation entre les mots visuels des bases VOC2012 et MIRflickr est ainsi plus faible que les mots visuels des bases CALTECH256, COREL ou VOC2005. Par conséquent, le pourcentage de réduction d'attributs de MIRflickr et VOC2012 dans le meilleur des cas est de 10 - 11% au lieu de

5% sur la base COREL ou 1.6% sur la base VOC2005.

Les propriétés de l'ensemble de données sont aussi un facteur qui influence le taux de réduction d'attributs. En effet, concernant les ensembles CALTECH256 et COREL, la même procédure est utilisée pour obtenir les sacs de mots visuels. Pourtant, le taux de réduction d'attributs maximum de COREL est d'environ 5% alors qu'il est de 0% pour CALTECH256.

Le nombre d'attributs réduits par l'algorithme de réduction RedAttsSansPerte n'est pas uniforme vis à vis de l'augmentation du seuil de binarisation. Notre algorithme comprend trois étapes : clarification, standardisation et réduction. Afin d'analyser le comportement de cet algorithme et la contribution de chaque étape de l'algorithme, nous étudions l'évolution du rapport d'attributs supprimés (FA) pour chaque étape de l'algorithme de réduction RedAttsSansPerte (*cf.* sous-section 5.1.1). Nous rappelons ces calculs :

$$FA_{clarification} = FA_{12} = \frac{N_{attInit_1} - N_{attInit_2}}{N_{attInit_1}}; \tag{5.2.1}$$

$$FA_{standardisation} = FA_{23} = \frac{N_{attInit_2} - N_{attInit_3}}{N_{attInit_2}}; \tag{5.2.2}$$

$$FA_{reduction} = FA_{34} = \frac{N_{attInit_3} - N_{attInit_4}}{N_{attInit_3}}; \tag{5.2.3}$$

Les figures 5.11, 5.12, 5.13 illustrent la fraction d'attributs réduits à chaque étape en fonction du seuil de binarisation sur huit bases de données. Les valeurs de FA_1, FA_2, FA_3 sont calculées comme présenté dans les équations 5.2.1, 5.2.2 et 5.2.3 en amont.

La courbe FA_1 dans les trois figures 5.11, 5.12 et 5.13 illustre le ratio d'attributs réduits par l'étape *clarification*[15]. Dans cette étape, nous cherchons les attributs qui sont équivalents dans une même clique du graphe de précédence. Nous gardons un seul représentant et supprimons le reste des attributs dans cette clique. Dans les ensembles de données Arcene et Dexter, le nombre des attributs qui sont équivalents est plus important que celui dans l'ensemble de données Gisette. Avec la base Arcene et la base VOC2005, nous remarquons que plus le seuil de binarisation augmente, plus le nombre d'attributs réduits par l'étape de clarification augmente. Cela indique que, avec un seuil binaire approprié, ces attributs sont associés aux mêmes objets et donc équivalents et seront supprimés par l'étape de clarification. Nous pouvons expliquer de la même manière, la réduction par l'étape de

15. Étape 1.

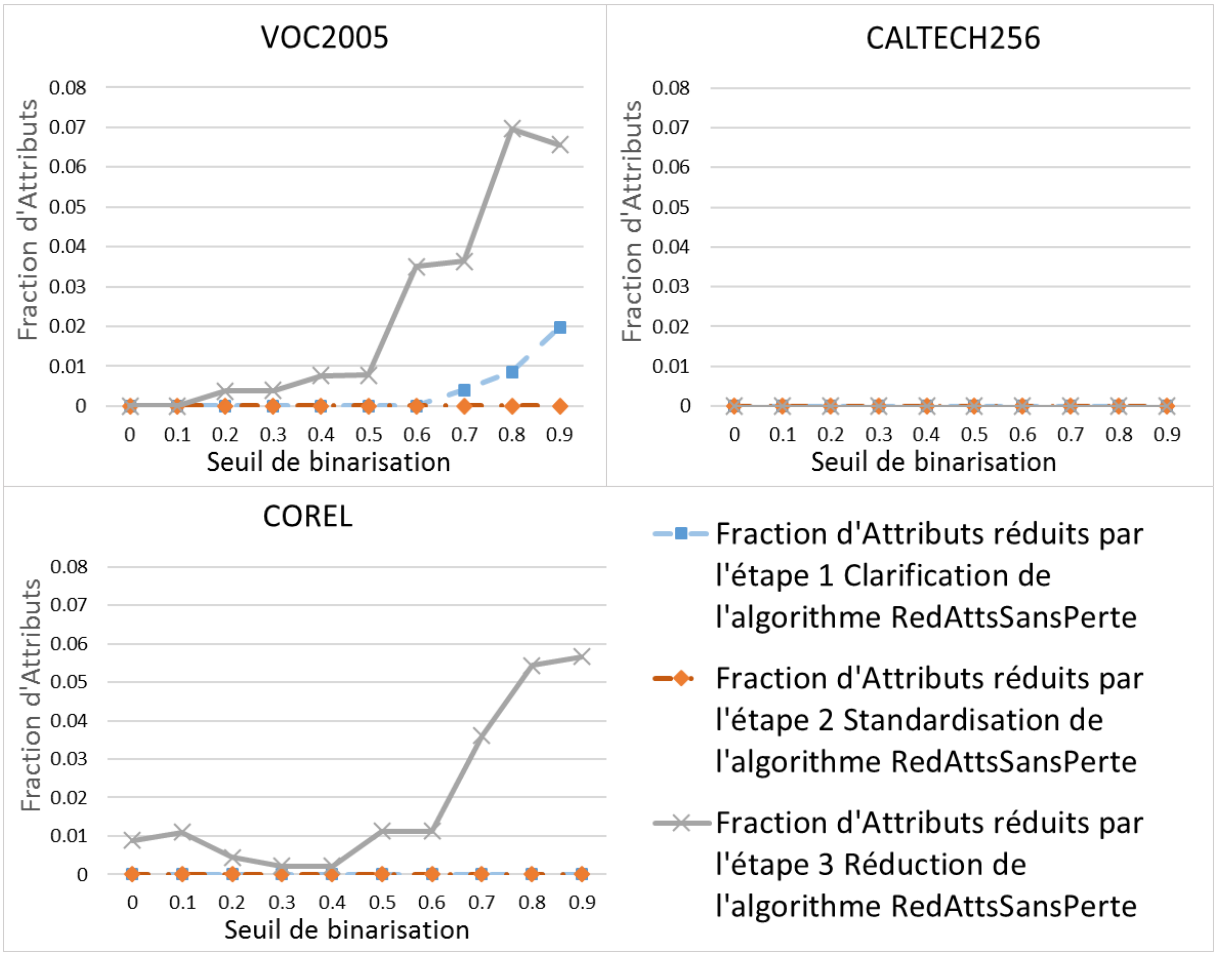


FIGURE 5.11: Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_1, la courbe rouge (trait plein), FA_2, la courbe bleue (trait tiret), et FA_3, la courbe vert (trait points et tirets alternés).

clarification dans les ensembles de données VOC2012, MIRflickr et Dexter. Dans la base Gisette, le nombre des valeurs qui sont inférieures à 0.9 est faible. La binarisation ne réduit presque rien. Par conséquent, l'influence du seuil de binarisation sur la quantité d'attributs équivalents est insignifiante.

L'étape de *standardisation*[16] de l'algorithme RedAttsSansPerte ne réduit rien pour les huit bases CALTECH256, VOC2005, COREL, VOC2012, MIRflickr, Arcene, Dexter et Gisette car il n'existe pas d'attribut commun à tous les objets dans ces trois bases.

16. Étape 2.

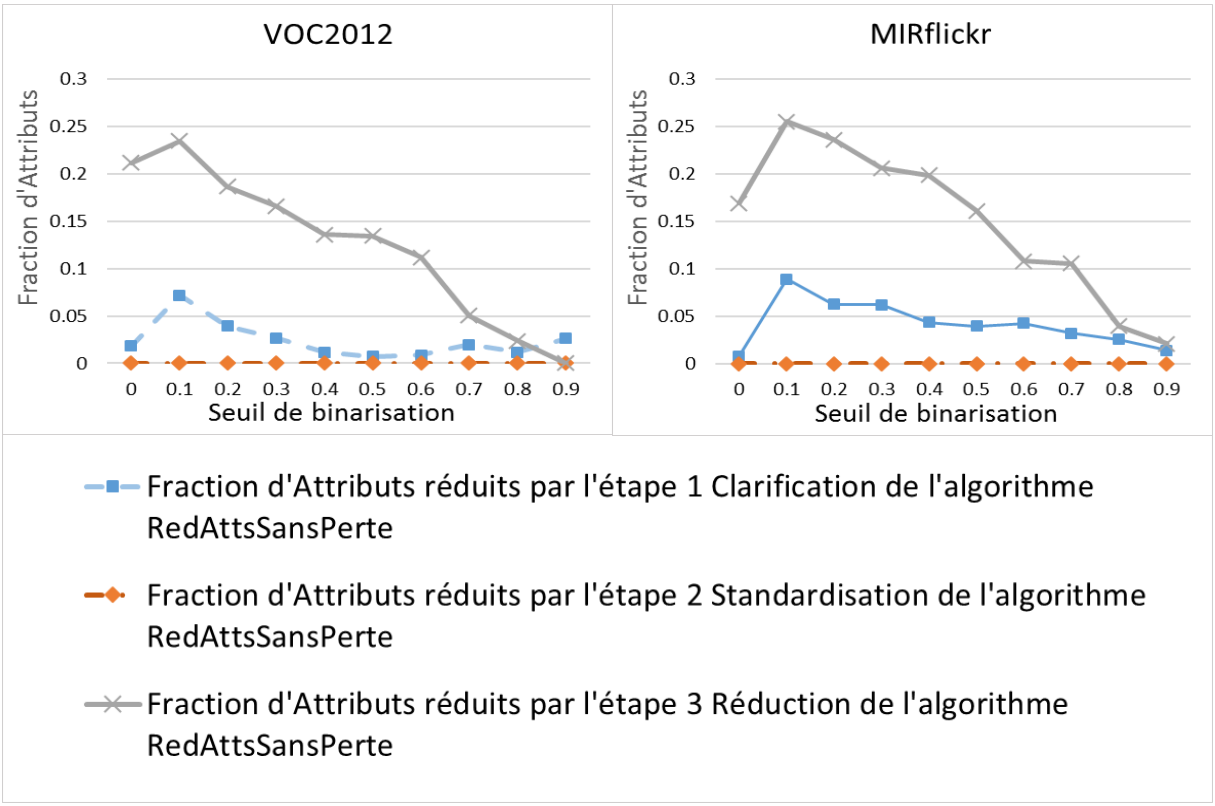


FIGURE 5.12: Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_1, la courbe rouge (trait plein), FA_2, la courbe bleu (trait tiret), et FA_3, la courbe vert (trait points et tirets alternés).

L'étape de *réduction*[17] de l'algorithme RedAttsSansPerte réduit des attributs qui sont associés aux mêmes objets qu'un ensemble d'attributs Nous remarquons une chute de la quantité d'attributs réduits par l'étape 3 dans l'ensemble de données Gisette à partir du seuil de binarisation 0.3. Cette chute vient du fait que des attributs permettant d'identifier une équivalence E_x ont été supprimés avec le seuil de binarisation supérieur ou égal à 0.3. Ce qui signifie que des attributs redondants dans ce troisième cas ne sont pas supprimés.

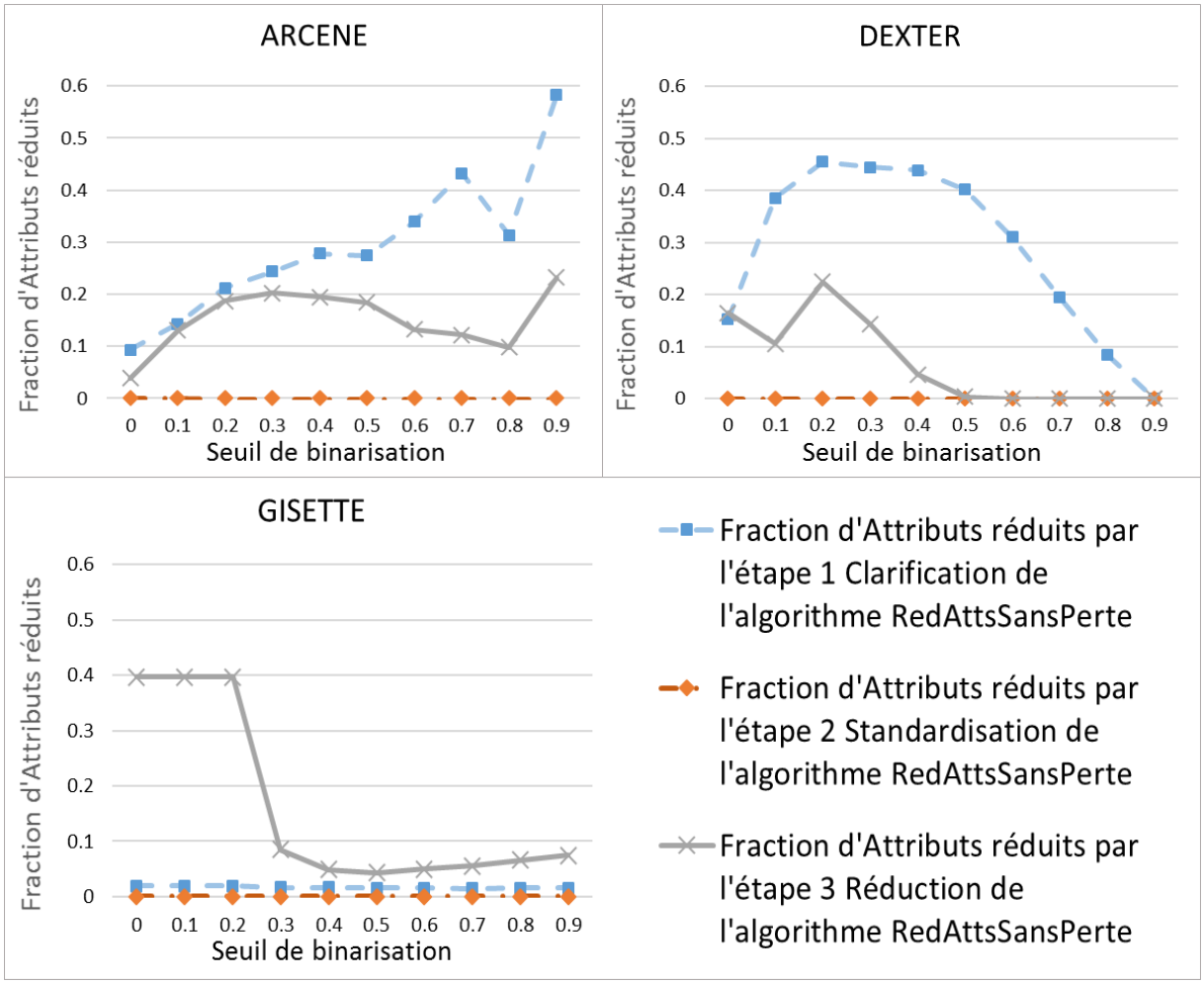
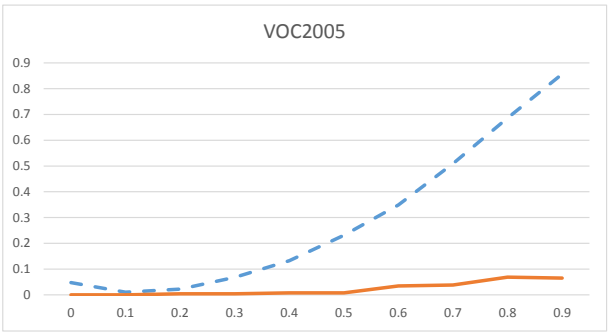


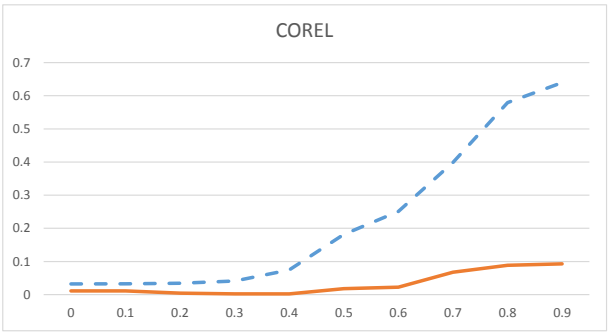
FIGURE 5.13: Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux avec FA_{12}, la courbe rouge (trait plein), FA_{23}, la courbe bleu (trait tiret), et FA_{34}, la courbe vert (trait points et tirets alternés).

Difficultés liés à la détermination d'un seuil binaire optimal

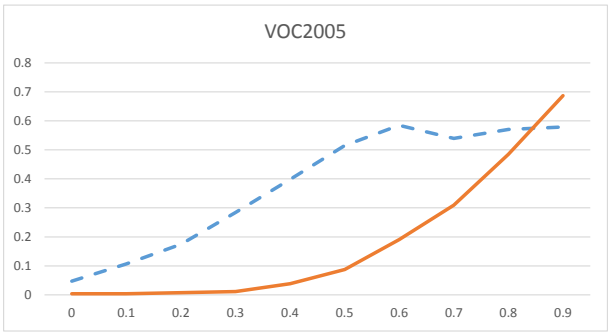
Étant donné que le seuil de binarisation influence la capacité de réduction des mots visuels, nous voulons déterminer automatiquement le seuil binaire qui soit le meilleur compromis entre le nombre d'attributs réduits et le nombre d'images non correctement représentées (i.e. mal classifiées). Par conséquent, ce compromis s'exprime comme le croisement entre la courbe représentant le nombre d'attributs réduits et celle représentant les

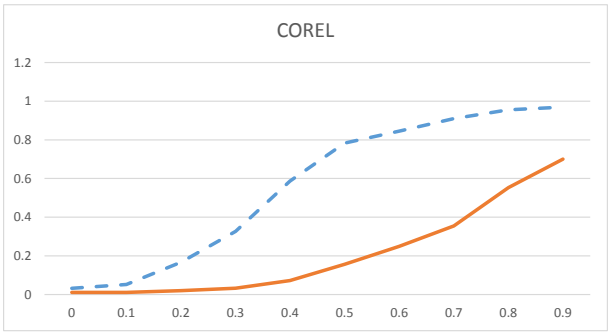
17. Étape 3.





(a) Normalisation par ligne (NormLigneMax)





(b) Normalisation par colonne (NormColonne-Max)

FIGURE 5.14: Croisements entre le nombre d'attributs réduits (la courbe bleu, trait tiret) et le nombre d'images réduites (la courbe rouge, trai plein) en fonction du seuil de binarisation dans le cas d'une normalisation par ligne ou d'une normalisation par colonne.

images non représentées. Ce point d'intersection nous permet d'obtenir le seuil binaire automatiquement.

Comme le montre la figure 5.14, le croisement attendu n'est pas toujours possible. Ce comportement est normal car plus le seuil binaire augmente, plus les matrices objets/attributs deviennent creuses [18]. Il n'existe donc pas de certitude qu'un tel croisement existe. Le résultat dépend en pratique de l'ensemble de données et n'est donc pas adapté au calcul d'un seuil optimal.

18. La plupart des valeurs sont nulles.

Conclusion

Bien que l'algorithme RedAttsSansPerte soit sensible aux données, nous avons obtenu une réduction intéressante sur les ensembles de données COREL5k, VOC2005, MIRflickr25000, VOC2012, Arcene, Dexter, Gisette. Cependant, sur la base CALTECH256, nous n'obtenons pas de réduction. Nous émettons l'hypothèse que la méthode de normalisation peut influencer la capacité de réduction de l'algorithme. Nous vérifions cette hypothèse dans la sous-section suivante.

5.2.1.2 Effets de l'étape de normalisation sur la réduction

La question que nous avons envisagée est la suivante : dans quelle mesure la méthode de normalisation influence la capacité de réduction de dimension de l'algorithme RedAttsSansPerte ?

Nous avons testé trois méthodes différentes de normalisation sur la base de données CALTECH256. Nous ne réduisons rien avec la normalisation NormLigneMax et NormColonneMax, mais une réduction a été obtenue avec la normalisation NormLigneSomme [19](*cf.* figure 5.15).

Avec la normalisation NormLigneSomme, une légère réduction d'attributs (maximum 19.7%) apparait dans l'intervalle [0.1,0.4] du seuil de binarisation (figure 5.15). Cela montre bien que l'algorithme de réduction est sensible à la méthode de normalisation des données.

5.2.1.3 Effet de l'étape de création des sacs de mots visuels sur la réduction

Nous avons montré l'efficacité de la réduction d'attributs de l'algorithme RedAttsSansPerte sur différentes bases de données dans la sous-section 5.2.1.1. Nous avons montré aussi la sensibilité de la construction des sacs de mots visuels à l'étape de normalisation, et notamment l'effet de cette étape sur la réduction.

19. La normalisation NormLigneSomme est présentée dans la sous-section 4.2.1.3 du chapitre 4.

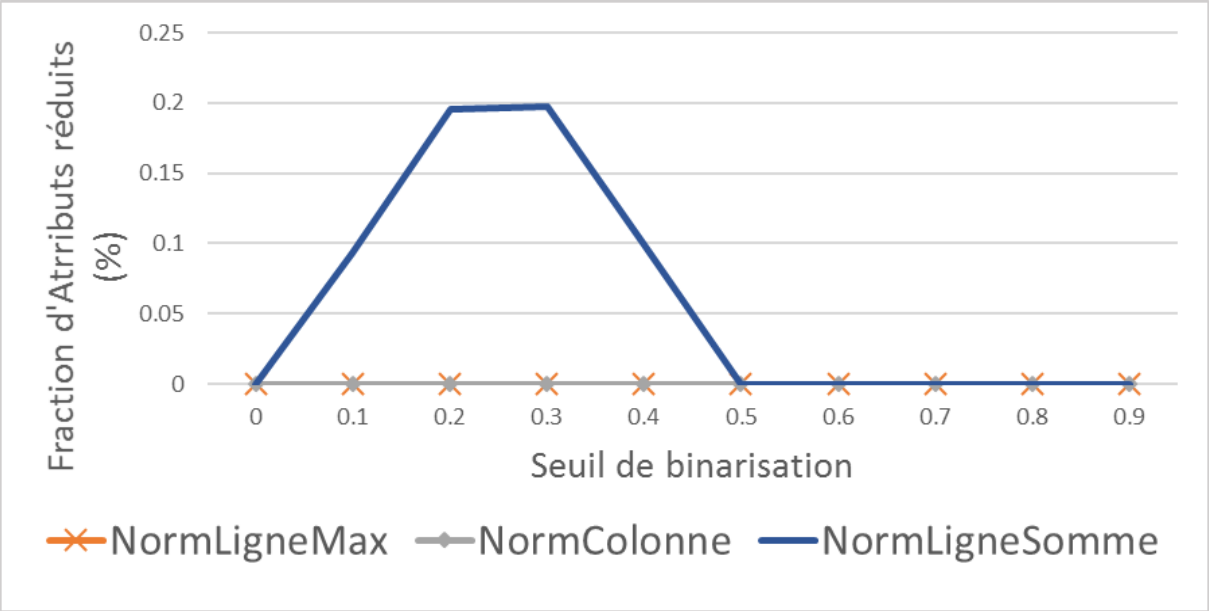


FIGURE 5.15: Fraction d'Attributs réduits par l'algorithme RedAttsSansPerte sur le sac de 500 mots visuels de la base de données CALTECH256 en fonction du type de normalisation.

Vu la sensibilité de la réduction aux méthodes de construction des sacs de mots visuels, nous avons émis les hypothèses suivantes :

- Différentes méthodes de détection, de description et de clustering produisent différents sacs de mots visuels et peuvent entrainer différentes capacités de réduction d'attributs.
- Le nombre de mots visuels initiaux influence la capacité de réduction d'attributs.

Nous allons mettre en évidence ces hypothèses dans cette sous-section.

Cette partie illustre la réduction des mots visuels issus d'une base d'images, en utilisant différents détecteurs, descripteurs, méthodes de regroupement et système d'encodage pour obtenir les sacs de mots visuels. Comme présenté dans la figure 5.16, ces étapes sont nécessaires pour construire un sac de mots visuels. En fonction de l'objectif du traitement d'images (*e.g.* la reconnaissance de visage, la reconnaissance de personne), les méthodes utilisées dans chaque étape ne sont pas les mêmes. Par exemple, des méthodes de détection qui prennent en compte des caractéristiques spécifiques du visage (*e.g.* la distance entre deux yeux et la bouche sur un visage) ont été développées pour la reconnaissance de visage (*e.g.* [Rowley 1998, Viola 2004]). Et les méthodes de détection développées pour la reconnaissance de personnes prennent en compte des caractéristiques spécifiques de la personne (*e.g.* le rapport largeur/hauteur d'une personne) comme [Salas 2011, Rodriguez 2011].

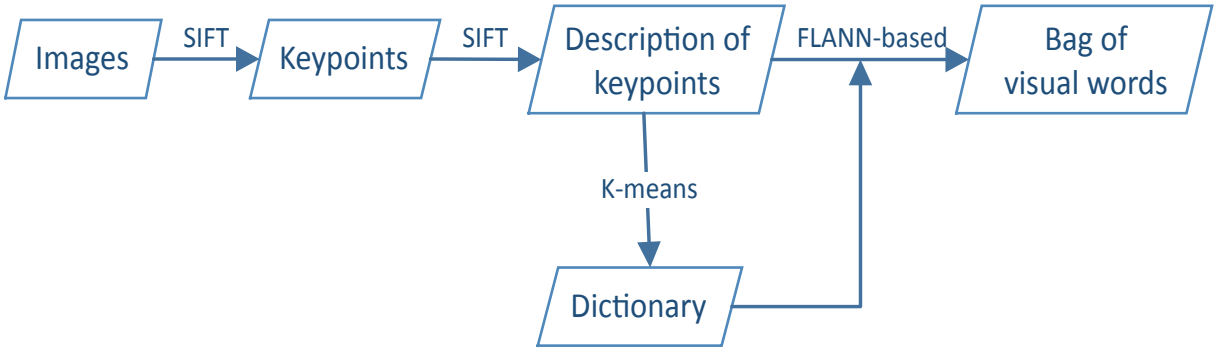


FIGURE 5.16: Chaîne de traitement classique permettant d'obtenir le modèle de sac de mots visuels. Les méthodes utilisées sont SIFT, SIFT, K-means et FLANN-based pour les étapes de détection, description, clustering et encodage respectivement.

L'objectif de cette sous-section est de vérifier l'hypothèse selon laquelle les différentes méthodes de détection, de description et de regroupement ainsi que leurs paramètres peuvent influencer la capacité de réduction.

Nous considérons les sacs de mots visuels obtenus sur la base VOC2012 par Syntyche GBEHOUNOU [20]. Le dictionnaire a été construit avec la méthode de détection Harris-Laplace [Mikolajczyk 2004], la méthode de description CMI (Color Moment Invariants) [Mindru 2004] et les mots visuels ont été choisis aléatoirement.

Nous construisons un deuxième dictionnaire de mots visuels avec différentes méthodes sur cette même base afin de comparer ces deux sacs de mots visuels. Nous choisissons la méthode SIFT [Lowe 2004] pour la détection et la description, la méthode K-moyennes [Lloyd 1982] pour le regroupement et la méthode FLANN-based [Itseez 2014, Muja 2013] pour l'encodage comme présenté dans la figure 5.16. La notation est SIFT+SIFT+K-moyennes. Les raisons sont discutées dans la conclusion du chapitre 3.

La figure 5.17 illustre les différentes capacités de réduction d'attributs sur deux sacs de mots visuels différents pour la même base de données. Nous pouvons observer que sur la même base, les méthodes de calcul du dictionnaire (détection, description et regroupement) influencent la réduction. En effet, avec les sacs de mots visuels obtenus par les méthodes "Harris-Laplace+CMI+aléatoire", nous obtenons une réduction d'attributs importante. Alors qu'elle n'est pas significative avec les sacs de mots visuels obtenus par

20. Membre du laboratoire XLIM-SIC en 2013.

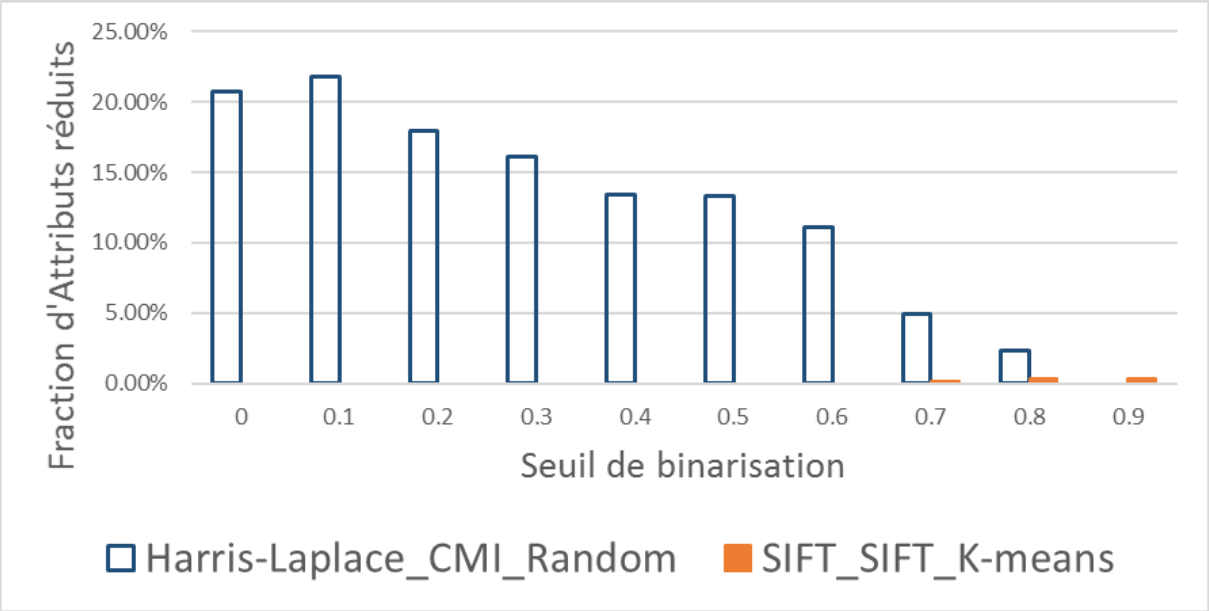


FIGURE 5.17: Différents capacités de réduction d'attributs de l'algorithme RedAttsSans-Perte sur la base de données VOC2012 avec différentes méthodes de construction du sac de mots visuels.

les méthodes "SIFT+SIFT+K-moyennes". L'hypothèse à l'origine de cette sous-section est donc validée. Par conséquent, les méthodes de construction d'un sac de mots visuels influencent la capacité de réduction de notre algorithme.

Lors de la création du sac de mots visuels, beaucoup de paramètres ont un impact sur les sacs de mots visuels obtenus. Ces sacs, qui représentent des images, sont appelés le contexte [21]. Ce contexte est l'entrée de la méthode de réduction d'attributs RedAttsSans-Perte. De ce fait, la capacité de réduction de l'algorithme proposé est changée en fonction du contexte d'entrée. Un des paramètres qui changent les caractéristiques du sac de mots visuels est la taille de ce sac. Nous testons sur un seul ensemble de données avec plusieurs tailles de sacs de mots visuels.

L'hypothèse est que le sac initial de mots visuels est trop petit pour représenter toutes les informations pertinentes dans les images. Par conséquent, ce sac ne contient pas d'information redondante qui pourrait être réduite.

21. La définition 2.2.1 du chapitre 2.

Nous avons expérimenté cette hypothèse en utilisant les deux premiers types de normalisation, dix seuils de binarisation et les paramètres des méthodes d'extraction d'attributs par défaut afin d'obtenir trois dictionnaires de mots visuels de tailles différentes : 500, 1000 et 5000 mots visuels. La base de données utilisée est CALTECH256. Cependant, nous n'avons pas obtenu de réduction du nombre d'attributs. Selon [Hou 2010], la taille optimale du dictionnaire des mots visuels servant à représenter une base d'images est entre 16157 et 64009. Par conséquent, les dictionnaires que nous avons testés sont trop petits pour représenter toutes les informations pertinentes qui sont contenues dans les images. Ainsi, ces dictionnaires ne contiennent pas d'information redondante qui pourrait être réduite. Pour des raisons matérielles, nous n'avons pas eu l'opportunité de tester avec des tailles plus grandes.

5.2.2 Point de vue qualitatif

Dans cette sous-section, nous évaluons qualitativement l'algorithme RedAttsSansPerte. Premièrement, nous vérifions l'hypothèse selon laquelle les mots visuels peuvent être réduits mais restent suffisamment représentatifs pour permettre d'effectuer une bonne classification des catégories présentes dans les différentes bases de données. Deuxièmement, nous évaluons la qualité du sac de mots visuels réduits par rapport au sac de mots visuels initial en utilisant la F-mesure.

5.2.2.1 Réduction par classe

Nous présentons ci-dessous le protocole expérimental, sa mise en œuvre et les résultats de la réduction de dimension par classe sur les bases VOC 2005 et CALTECH256.

Comme nous l'avons remarqué dans la figure 5.8, la réduction par l'algorithme RedAttsSansPerte n'est pas significative sur la base VOC2005 (1.6% d'attributs dans le meilleur des cas), et nous n'observons pas de réduction sur la base CALTECH256, lorsque nous considérons l'ensemble des catégories. Dans le cas où nous appliquons l'algorithme de réduction sur chaque classe, le nombre d'attributs est réduit et ce d'autant plus que le seuil de binarisation est élevé. Par exemple, avec un seuil binaire de 0.9, le nombre d'attributs retenus pour la classe Persons de la base VOC2005 est de 58 après réduction à comparer aux 262 attributs initiaux. Par contre, si nous considérons l'ensemble des classes, il n'y a

Seuil de binarisation	Attributs restants sur les 262 initiaux				Ensemble des 4 classes
	Persons	Moto	Cars	Bike	
0.0	262	262	262	261	262
0.1	261	262	262	261	262
0.2	253	261	261	260	262
0.3	227	260	259	253	262
0.4	193	250	255	230	262
0.5	157	235	245	193	262
0.6	116	202	232	158	262
0.7	86	170	212	127	262
0.8	68	134	173	94	262
0.9	58	103	148	78	262

TABLE 5.3: Nombre d'attributs restants (sur les 262 attributs initiaux) après réduction par catégorie pour la base VOC 2005 - dataset 1, en fonction du seuil de binarisation. La dernière colonne donne le nombre d'attributs subsistant si on considère l'ensemble des 4 classes. Dans ce cas, on ne peut pas supprimer un attribut s'il est utilisé pour catégoriser des classes. De ce fait, il faut prendre l'union de tous les attributs.

pas de réduction.

Cette expérimentation montre que l'algorithme RedAttsSansPerte peut difficilement s'appliquer dans le domaine de recherche d'images par le contenu comme une méthode de réduction de dimension. Cependant, il peut être appliqué au niveau de l'étape d'indexation des images pour choisir un sous-ensemble des attributs pertinents qui représentent une classe. Ce sous-ensemble servira à construire l'index pour les images qui appartiennent à cette classe.

Seuil de binarisation	Attributs restants sur les 1000 initiaux					Ensemble des 256 classes
	C1	C57	Moyenne	C200	C256	
0.0	944	904	929.96	994	996	1000
0.1	926	899	909.90	978	994	1000
0.2	817	819	833.28	909	983	1000
0.3	652	693	713.52	705	941	1000
0.4	452	501	491.14	408	830	1000
0.5	317	391	388.66	267	691	1000
0.6	143	203	197.23	177	354	1000
0.7	101	138	118.91	100	193	999
0.8	89	125	103.36	87	123	999
0.9	86	123	98.37	79	102	1000

TABLE 5.4: Nombre d'attributs restant (sur les 1000 attributs initiaux) après réduction par catégorie pour la base CALTECH256, en fonction du seuil de binarisation. La dernière colonne donne le nombre des attributs subsistant pour l'union des attributs restants des 256 catégories. La réduction se fait sur le sac normalisé de mots visuels par la normalisation NormLigneMax.

5.2.2.2 Évaluation par la F-mesure

En fonction du seuil de binarisation

Notre objectif est d'évaluer les capacités de l'algorithme RedAttsSansPerte à réduire les attributs en conservant des informations pertinentes. La figure 5.18 illustre les résultats de la F-mesure sur l'ensemble des attributs retenus et l'ensemble des attributs initiaux en fonction du seuil de binarisation et de la valeur de la Fraction d'Attributs réduits (FA) par cet algorithme par rapport au nombre d'attributs initiaux.

Avec l'ensemble de données Arcène, les valeurs de la F-mesure après l'application de l'algorithme RedAttsSansPerte sont presque égales à celles avant réduction. La Fraction d'Attributs réduits (FA) avec la base Arcène se situe entre 13% et 68%. Avec le seuil de binarisation à 0.6, l'algorithme réduit à 43% le nombre d'attributs en ayant une valeur de la F-mesure supérieure à celle avant la réduction. De la même manière, l'algorithme RedAttsSansPerte améliore légèrement les performances de classification avec des seuils

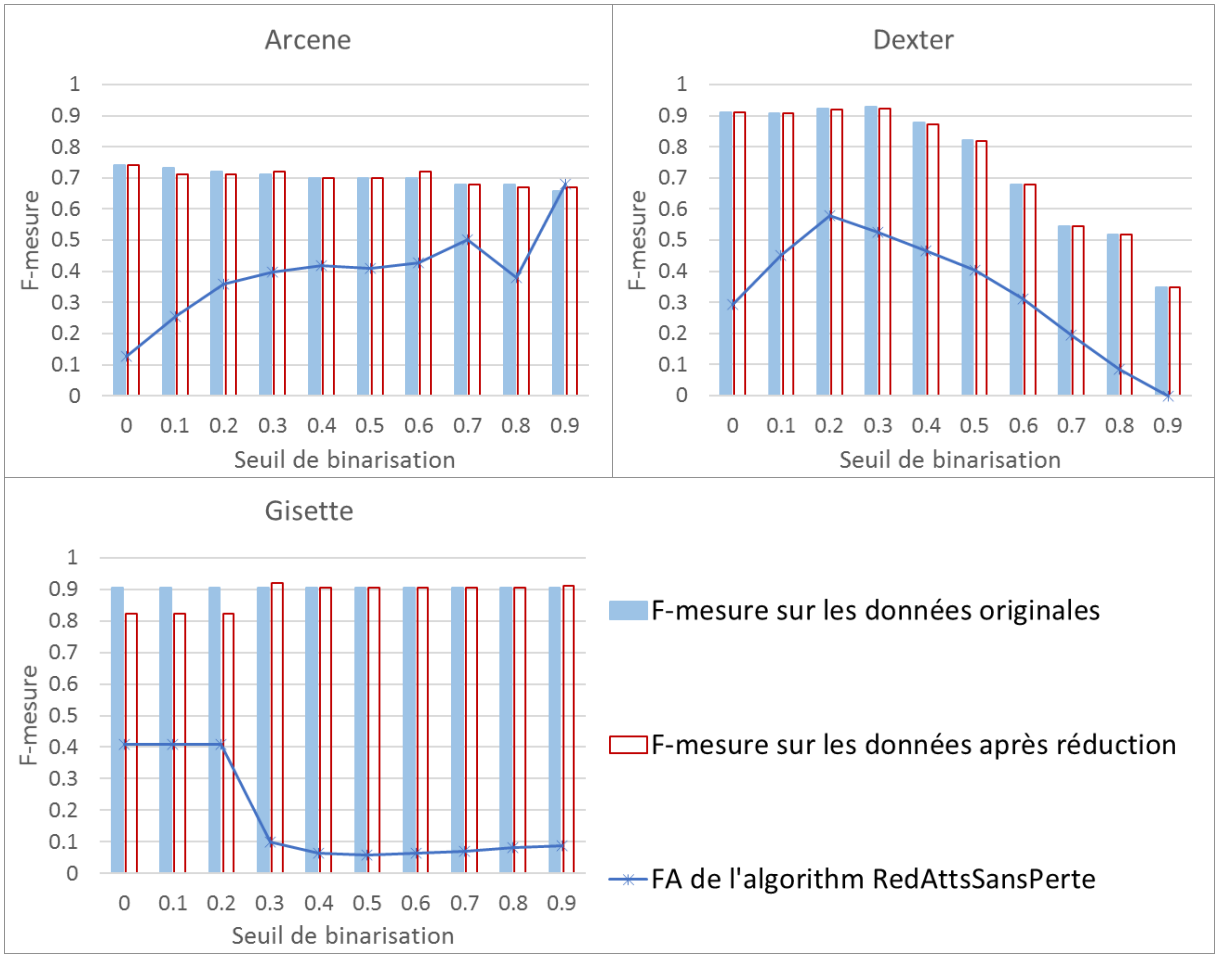


FIGURE 5.18: Comparaison du résultat de F-mesure sur les données avant et après l'application de l'algorithme de réduction RedAttsSansPerte. La courbe est la Fraction d'Attributs réduits (FA) de l'algorithme RedAttsSansPerte.

de binarisation égaux à 0.3 et 0.9.

Dans l'ensemble de données Gisette, les valeurs de la F-mesure après l'application de l'algorithme avec des seuils de binarisation de 0.0, 0.1, 0.2 sont plus faibles (0.08) que celles avant réduction quand l'algorithme réduit d'environ 40% le nombre d'attributs. A partir du seuil 0.3, les valeurs de la FA de l'algorithme sont autour de 10% mais les valeurs de la F-mesure après réduction sont supérieures ou égales à celles avant réduction. La diminution de la FA de l'algorithme sur cette base en fonction du seuil de binarisation est liée surtout à l'étape 3 de celui-ci. Cette analyse est présentée dans la section 5.2.1.1. Rappelons que

des attributs permettant d'identifier une équivalence E_x ont été supprimés lorsque le seuil de binarisation est supérieur ou égal à 0.3. Des attributs x qui sont équivalents à E_x ne sont donc pas supprimés par l'algorithme et donc le nombre d'attributs réduits diminue.

Sur l'ensemble de données Dexter, les données sont peu nombreuses, le contexte ne contient que 0.5% de valeurs non nulles. Dexter est donc une base de données contenant peu d'information, et la réduction est plus difficile pour des telles données. Quand le seuil de binarisation augmente, le nombre d'attributs avant la réduction diminue et les valeurs de la F-mesure chutent aussi. Cependant, ces valeurs après réduction sont approximativement égales à celles avant réduction.

5.3 Evaluation de l'algorithme RedAttsFloue

Dans cette section, nous évaluons l'algorithme RedAttsFloue (section 4.4 du chapitre 4) de la même manière (quantitativement et qualitativement), que dans la partie 5.2. Nous expérimentons cet algorithme sur trois bases de données : Arcene, Dexter et Gisette.

5.3.1 Point de vue quantitatif

Comme nous l'avons présenté dans la section 4.4 du chapitre 4, un des paramètres les plus importants de l'algorithme RedAttsFloue est le seuil de flexibilité.

5.3.1.1 Seuil de flexibilité fixe

La proportion d'attributs réduits par l'algorithme RedAttsFloue avec un seuil de flexibilité $\delta = 0.1$ en fonction du seuil de binarisation sur les trois ensembles de données Arcene, Gisette et Dexter est présenté dans la figure 5.19. La courbe de la proportion d'attributs réduits de l'algorithme RedAttsFloue a une forme similaire à celle de l'algorithme RedAttsSansPerte. De plus, pour la plupart des seuils de binarisation le nombre d'attributs réduits par l'algorithme RedAttsFloue, comme attendu, est supérieur à celui

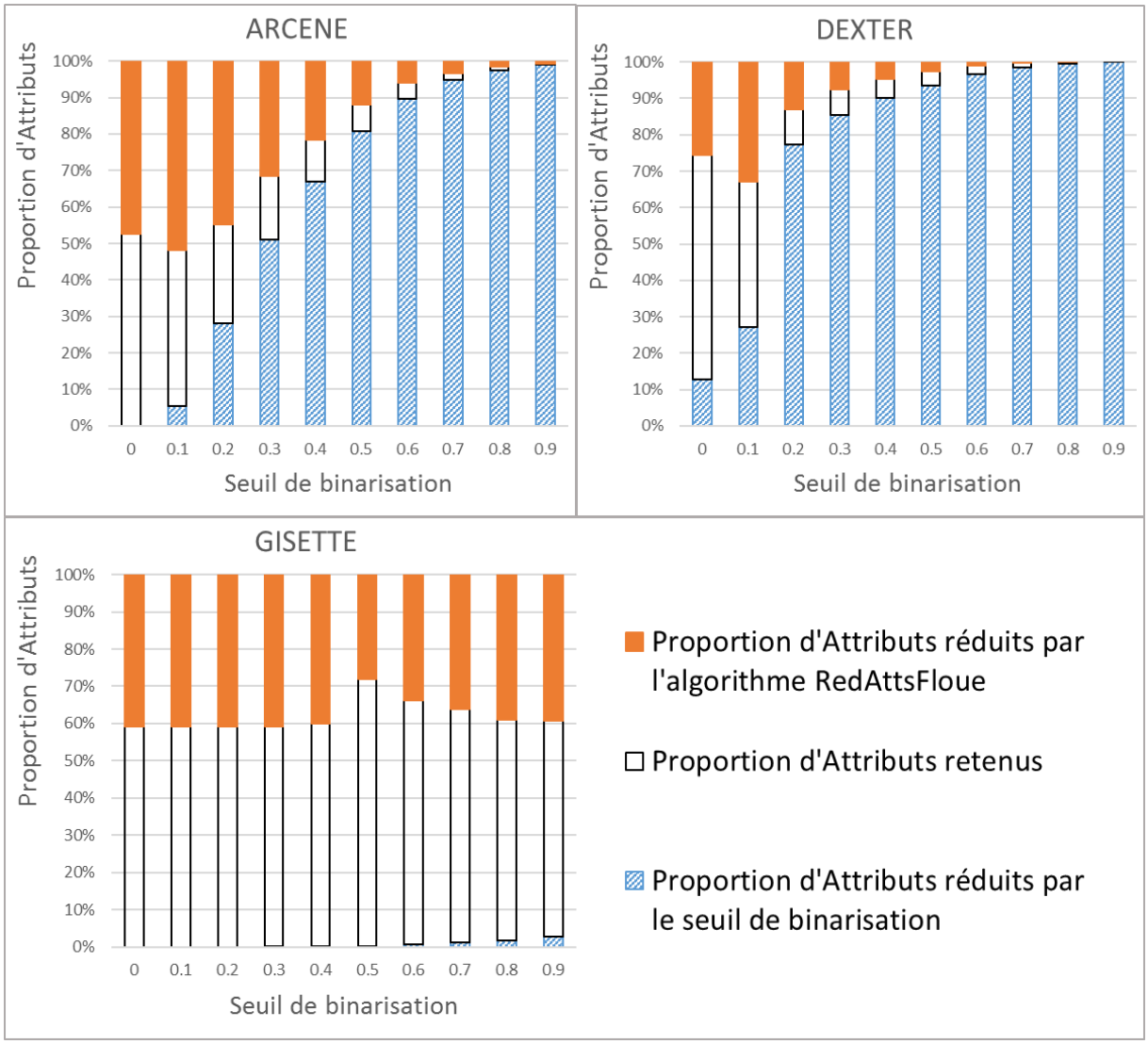


FIGURE 5.19: Proportion d'attributs réduits de l'algorithme de réduction RedAttsFloue et proportion d'attributs réduits par seuil de binarisation sur l'ensemble initial d'attributs des trois ensembles de données Arcene, Dexter et Gisette.

obtenu avec l'algorithme RedAttsSansPerte. Nous comparons les résultats de classification sur les données réduites par chacun des algorithmes dans la sous-section 5.3.2.

L'algorithme RedAttsFloue contient cinq étapes de réduction. La capacité de réduction des trois premières étapes ne dépend pas du seuil de flexibilité. Autrement dit, le rapport

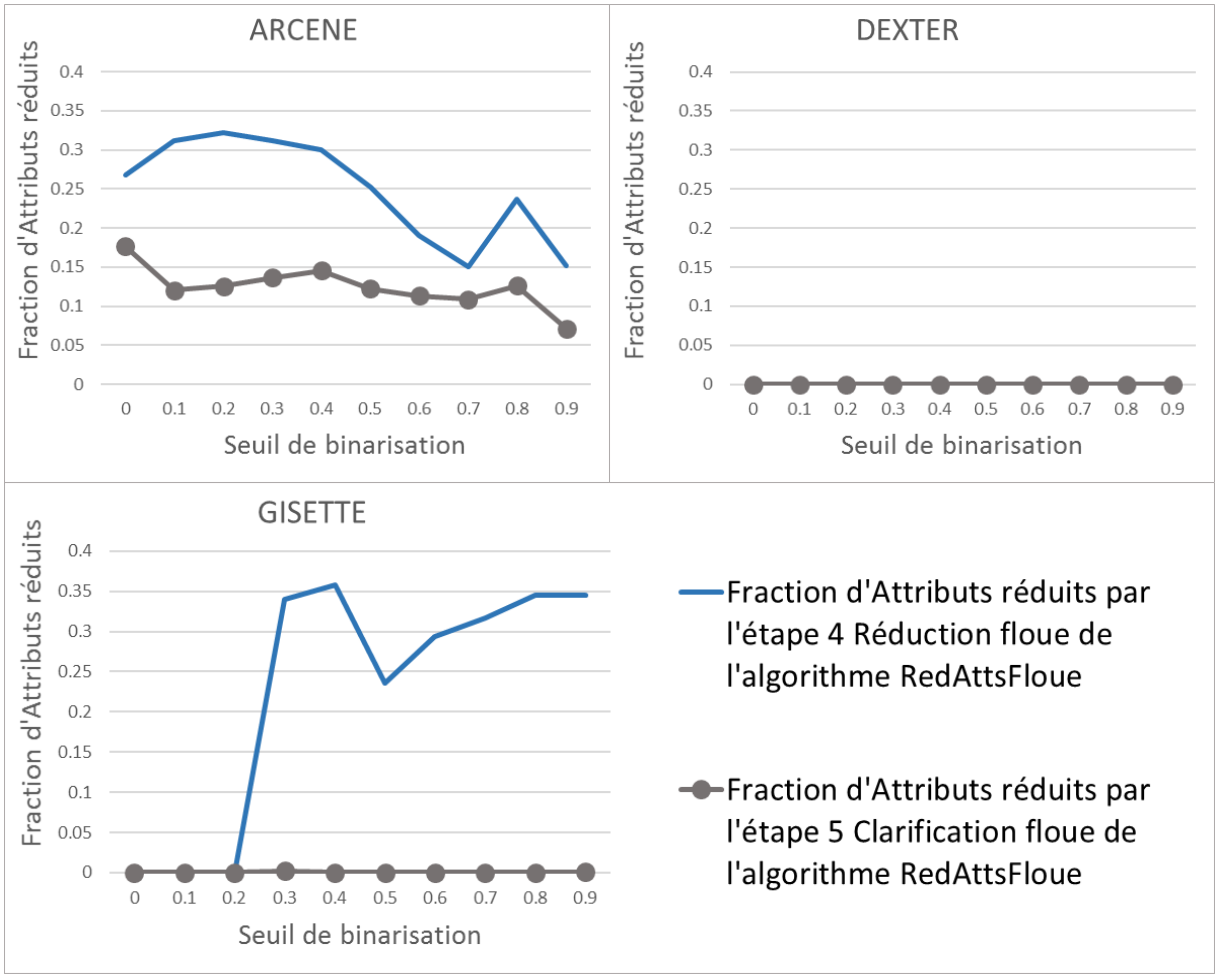


FIGURE 5.20: Fraction d'Attributs réduits (FA) de deux dernières étapes de l'algorithme RedAttsFloue avec le seuil de flexibilité est égal à 0.1 : FA_4, l'étape de la réduction floue, FA_5, l'étape de la clarification floue.

d'attributs supprimés pour les trois première étapes (FA_1, FA_2, FA_3) est identique pour les deux algorithmes.

Afin d'analyser le comportement de l'algorithme RedAttsFloue, nous utilisons la fraction d'attributs supprimés pour les deux étapes *réduction floue* et *clarification floue* (sous-section 5.1.1) :

$$FA_{reductionFloue} = FA_{45} = \frac{N_{attInit_4} - N_{attInit_5}}{N_{attInit_4}}; \tag{5.3.1}$$

$$FA_{clarificationFloue} = FA_{56} = \frac{N_{attInit_5} - N_{attInit_6}}{N_{attInit_5}}; \quad (5.3.2)$$

La figure 5.20 illustre, pour les trois ensembles de données, le comportement des deux étapes floues de l'algorithme en fonction du seuil de binarisation. Les ensembles de données Arcène et Gisette ont plus d'attributs similaires que Dexter. Ces comportements peuvent provenir de la densité de données : Arcène a environ 50% d'attributs non nuls, Gisette en a environ 13% et Dexter n'en a que 0.5%.

5.3.1.2 Variation du seuil de flexibilité

L'algorithme **RedAttsFlou** a été appliqué sur les trois bases Arcène, Dexter et Gisette. Nous faisons varier le seuil de flexibilité entre 0.1 et 0.9 avec un pas de 0.1. Les résultats de réduction sur la base Dexter à partir du seuil de flexibilité 0.3 ne sont pas disponibles (voir le tableau 5.5) car le graphe flou demande plus de mémoire que la configuration que nous avons utilisée (4GB de RAM).

FA_{45}		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de	**0.0**	0.0006	0.0036	nd [22]	nd	nd	nd	nd	nd	nd
binarisation	**0.1**	0	0.0021	0.0063	0.0333	nd	nd	nd	nd	nd

Table 5.5: Fraction d'Attributs réduits par les deux étapes 4 et 5 de l'algorithme RedAttsFloue sur la base Dexter avec des seuils de binarisation égaux à 0.0 et 0.1.

Les courbes de la figure 5.21 illustrent la fraction d'attributs réduits par les deux étapes réduction floue et clarification floue sur la base de données Arcene en fonction du seuil de flexibilité. Comme nous pouvons le remarquer dans cette figure, un seuil de flexibilité plus élevé entraine la plupart du temps un gain en taux d'attributs réduits. Cependant, quand la perte d'information est trop importante, la fraction d'attributs réduits n'augmente plus. Nous observons le même comportement sur les trois bases de données. Nous montrons en annexe B les six tableaux B.1, B.2, B.3, B.4, B.5, B.6 qui présentent les valeurs de FA_4, FA_5 de l'algorithme RedAttsFloue en fonction du seuil de flexibilité et du seuil de binarisation sur les trois ensembles de données Arcene, Dexter et Gisette.

22. non disponible.

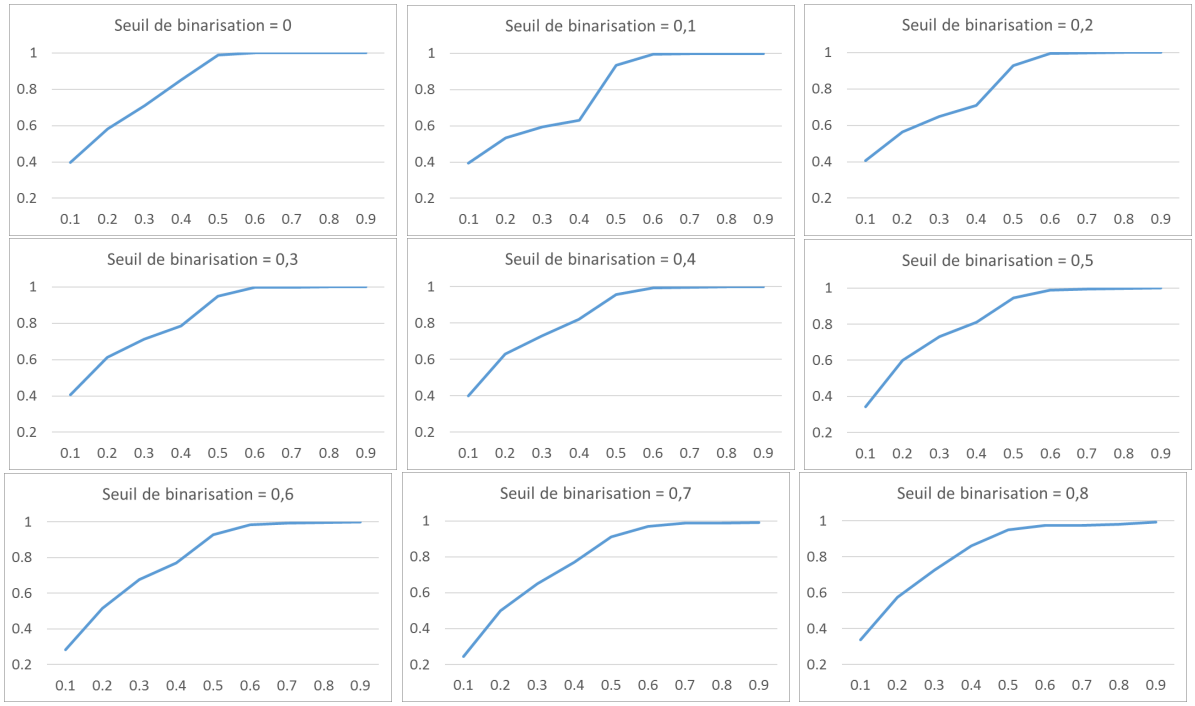


FIGURE 5.21: Fraction d'Attributs réduits des deux étapes floues de l'algorithme de réduction RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de binarisation [0, 0.8] et du seuil de flexibilité [0.1, 0.9].

La figure 5.22 illustre la variation des valeurs de FA_4, FA_5 en fonction du seuil de flexibilité avec un seuil de binarisation choisi. Dans les trois bases de données, un seuil de flexibilité plus élevé entraine une réduction plus importante par chaque étape, ce qui est conforme à l'utilisation attendue de la flexibilité (section 4.4 du chapitre 4).

5.3.2 Point de vue qualitatif

Le résultat de la F-mesure appliquée après l'algorithme RedAttsFloue varie en fonction du seuil de flexibilité, ce qui n'est pas le cas de l'algorithme RedAttsSansPerte. Un seuil de flexibilité plus élevé entraine mécaniquement une perte d'information et par la même occasion une baisse des performances en classification. Nous remarquons un accroissement de la valeur de la F-mesure de 0.44 à 0.56 avec un seuil de flexibilité compris entre 0.7 et 0.8 sur la base Arcene. Cela ne signifie pas que le classifieur a effectué une meilleure classification des données avec un seuil de flexibilité fixé à 0.8 qu'avec un seuil de flexibilité

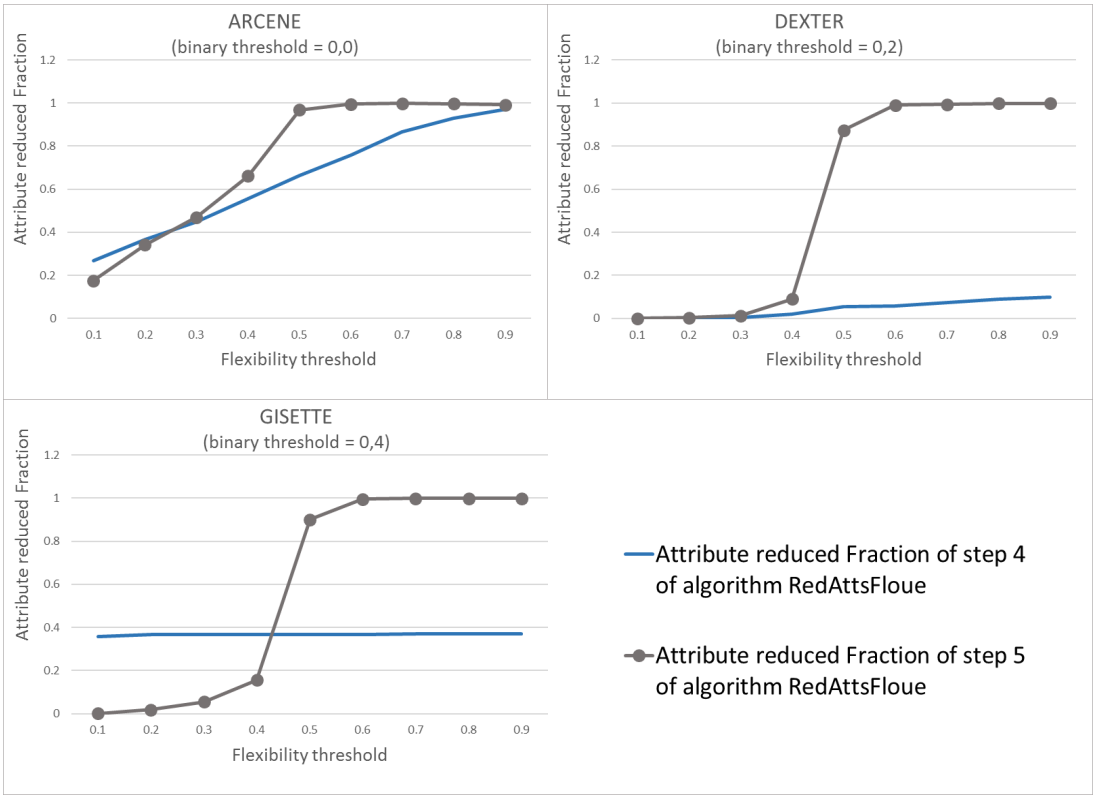


FIGURE 5.22: Rapport entre le nombre d'attributs réduits et le nombre d'attributs initiaux par l'algorithme RedAttsFloue (où le seuil de binarisation est égal à 0) en fonction du seuil de flexibilité : FA_{45}, l'étape de la *réduction floue*, FA_{56}, l'étape de la *clarification flou*.

fixé à 0.7. Ce phénomène se produit car le nombre d'objets de la classe 1 et celui de la classe 2 sont respectivement de 44 et de 56 dans le sous-ensemble de validation de la base Arcene. Lorsque le classifieur catégorise l'intégralité des objets dans une seule classe, la valeur de la F-mesure est égale à 0.44 ou 0.56. Le changement de cette valeur s'effectue en fonction de la classe qui a été choisie, et non pas de la capacité de classification du classifieur.

Avec la base de données Arcene, les valeurs de la F-mesure avant et après l'application de l'algorithme RedAttsSansPerte sont similaires. La Fraction d'Attributs réduits (FA) se situe entre 13% et 68% en fonction du seuil de binarisation. Celles de l'algorithme RedAttsFloue sont similaires à celles de l'algorithme RedAttsSansPerte avec FA entre 47% et 74%. Avec le seuil de binarisation à 0.6, les deux algorithmes réduisent de 43% à 59% le nombre d'attributs en ayant une valeur de la F-mesure supérieure à celle avant la

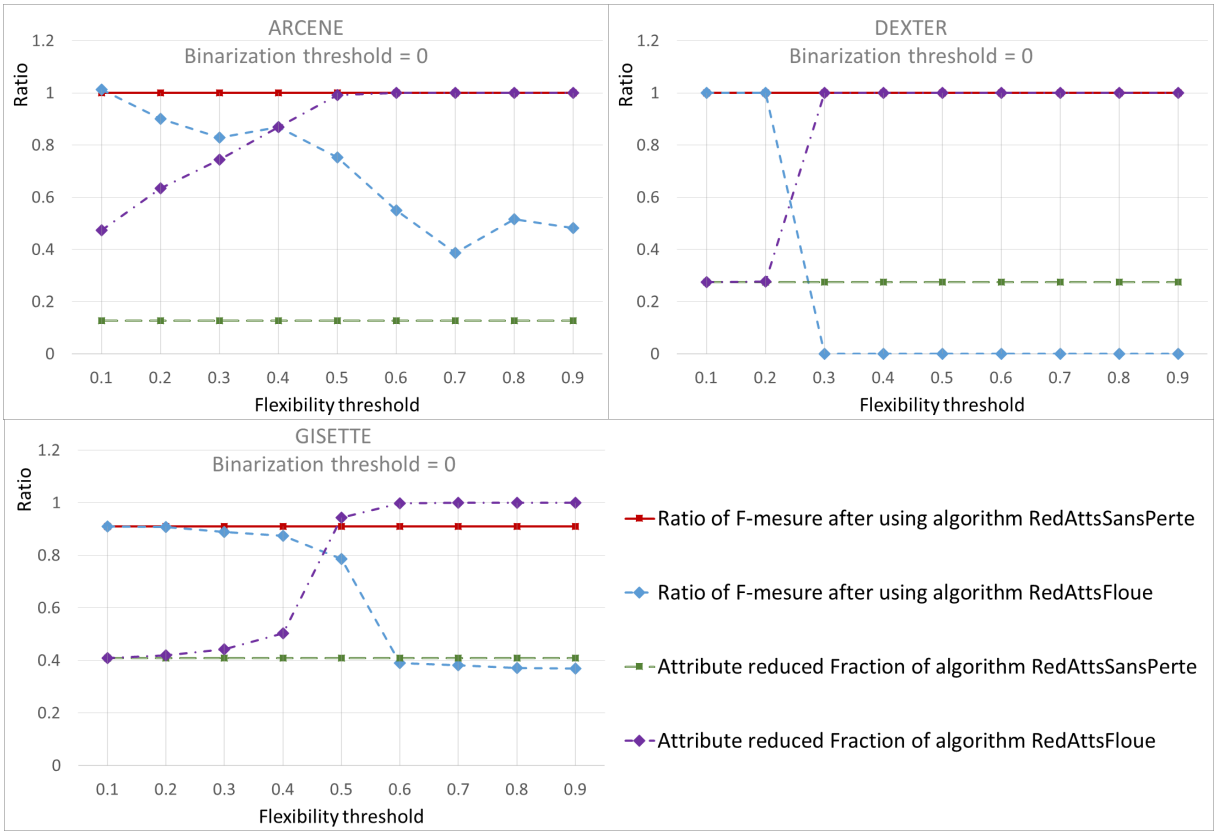


FIGURE 5.23: Comparaison du résultat de FA et de F-mesure sur les données au seuil de binarisation 0.0 de deux algorithmes de réduction : RedAttsSansPerte et RedAttsFloue en fonction du seuil de flexibilité.

réduction. De même, l'algorithme RedAttsSansPerte améliore légèrement les performances de classification avec des seuils de binarisation égaux à 0.3 et 0.9 quand l'algorithme RedAttsFloue l'améliore avec des seuils de binarisation égaux à 0.0, 0.4 et 0.7.

Pour la base Gisette, l'algorithme RedAttsFloue produit un résultat identique à celui de l'algorithme RedAttsSansPerte avec les seuils binaires inférieurs ou égaux à 0.2. Malgré la suppression des attributs permettant d'identifier une équivalence E_x lorsque le seuil de binarisation est supérieur ou égal à 0.3, l'algorithme RedAttsFloue réussit à supprimer des attributs x équivalents aux E_x. La Fraction d'Attributs réduits de l'algorithme RedAttsFloue maintient donc sa performance quand celle de l'algorithme RedAttsSansPerte a considérablement diminué. Ce cas montre la robustesse de l'algorithme RedAttsFloue. Comme nous l'avons analysé, des attributs qui ont été supprimés par l'étape de bina-

risation sont inclus dans l'ensemble équivalent E_x pour déterminer plusieurs attributs redondants x dans le contexte initial. L'algorithme RedAttsFloue a été capable de réduire la plupart de ces attributs redondants sans l'aide de l'intégralité de E_x. Au contraire, sans l'intégralité de E_x, l'algorithme RedAttsSansPerte n'arrive pas à supprimer les attributs redondants. C'est pourquoi avec les seuils de binarisation au-dessus de 0.2, l'algorithme de réduction RedAttsFloue réduit de trois à quatre fois plus le nombre d'attributs que l'algorithme RedAttsSansPerte. Cependant, les valeurs de la F-mesure de l'algorithme RedAttsFloue diminue. Ces résultats sont cohérents avec ceux que nous attendions puisque le seuil de flexibilité (la perte d'information correspondant) de l'algorithme RedAttsFloue est aussi réduit.

Sur l'ensemble de données Dexter, l'algorithme RedAttsFloue a le même comportement que l'algorithme RedAttsSansPerte. Étant donné que les données sont peu nombreuses, le contexte ne contient que 0.5% de valeurs non nulles. Par conséquent, deux attributs quelconques sont susceptibles de ne pas être égaux et ne seront pas supprimés sans un seuil élevé de flexibilité. Au contraire avec le cas de la réduction sur l'ensemble de données Gisette, la chute de la FA n'est pas en corrélation avec une perte des attributs en fonction du seuil de binarisation. Les attributs obtenus après la binarisation ne sont simplement pas réductibles sans une perte d'information élevée. Dexter est une base de données creuse (sparse data). La réduction est plus difficile pour des telles données.

En général, l'algorithme RedAttsFloue est plus efficace que l'algorithme RedAttsSansPerte car il réduit plus d'attributs tout en maintenant de bonnes performances de classification dans la plupart des cas. Il montre la stabilité des performance lorsque les données varient.

5.4 Conclusion

Dans ce chapitre, nous avons mené des expérimentations permettant de mettre en avant les capacités de réduction d'attributs redondants de l'algorithme RedAttsSansPerte appliqué aux signatures d'images en considérant l'ensemble des signatures comme un contexte formel. Nous avons testé notre algorithme de réduction sur plusieurs bases d'images. Nous avons aussi expérimenté la capacité de réduction des attributs de l'algorithme RedAttsFloue (l'extension de l'algorithme RedAttsSansPerte) sur trois bases de données.

Nous avons analysé le comportement de ces algorithmes d'un point de vue quantitatif et d'un point de vue qualitatif. Les résultats montrent que l'algorithme RedAttsFloue surpasse l'algorithme RedAttsSansPerte en terme de performances de réduction. L'algorithme RedAttsFloue entraine toutefois une perte d'information due à l'utilisation d'un seuil de flexibilité qui étend la similarité entre attributs. En fonction de la valeur de ce seuil de flexibilité, la quantité d'attributs réduits varie.

Points clés

Positionnement

- ❏ A partir de la chaîne de traitement présentée en figure 5.1, nous avons exploré différentes possibilités pour effectuer la réduction du contexte formel.
- ❏ Nous avons évalué l'algorithme RedAttsSansPerte et l'algorithme RedAttsFloue avec le point de vue quantitatif et qualitatif.

Contributions

- ❏ Nous avons mené des expérimentations permettant de mettre en avant que l'algorithme de réduction s'appuyant sur la théorie des treillis peut réduire les attributs redondants dans un contexte formel.
- ❏ Nous avons mené des expérimentations permettant d'analyser le comportement de l'algorithme de réduction RedAttsFloue sur les contextes formels.
- ❏ Nous montrons que l'algorithme de réduction RedAttsFloue est plus efficace en termes de réduction d'attributs que l'algorithme de réduction RedAttsSansPerte car il accepte une perte d'information, tout en ayant d'une complexité moindre due à une structure moins complexe du graphe de précédence flou.

Conclusions et perspectives

Rappel des objectifs

Les travaux exposés dans ce manuscrit sont tournés vers la réduction de dimension appliquée au domaine de l'image, et plus particulièrement à la sélection d'un sous-ensemble d'attributs efficaces pour la catégorisation et le regroupement de données numériques. En effet, avec la croissance exponentielle des volumes de données de nos jours, il est plus que nécessaire de réduire leur taille et donc leur dimension.

La première contribution est un algorithme de réduction qui repose sur la structure de treillis. Après l'avoir introduit, nous avons observé sa capacité de réduction d'attributs appliquée au domaine de l'image et plus particulièrement à des images décrites par des mots visuels.

La seconde proposition est une extension floue de cet algorithme pour une plus grande capacité de réduction tout en acceptant une perte d'information. En général, cette perte d'information réduit les performances de classification de données, mais dans certains cas comme présentés dans chapitre 5, cet algorithme est plus efficace que l'algorithme exact en terme de réduction d'attributs et de performances de classification.

Bilan des travaux effectués

Nous avons commencé par développer un algorithme de réduction de dimension (RedAttsSansPerte) qui s'appuie sur la structure de treillis.

Le chapitre 2 présente les éléments de la théorie des treillis dont nous avons besoin : le contexte, le treillis des concepts, le système de fermeture et le treillis de fermés. Il présente aussi le théorème fondamental qui stipule que la structure du treillis des concepts ne change pas (*i.e.* il est isomorphe à celui construit lorsque seuls les irréductibles sont conservés dans le contexte), ce qui signifie que les corrélations entre les données sont conservées. Notre algorithme RedAttsSansPerte repose sur ce théorème, à savoir qu'il ne conserve que les attributs "irréductibles".

Concernant la mise en œuvre de l'algorithme, nous avons utilisé le graphe de précédence où les nœuds correspondent aux attributs. Nous avons montré le lien entre ce graphe de précédence et la sous-hiérarchie de Galois (AC-poset) dans le chapitre 4. Quelques expérimentations menées dans le chapitre 5 montrent que l'algorithme RedAttsSansPerte semble prometteur. Nous avons mené aussi des expérimentations permettant de mettre en avant l'instabilité de l'algorithme de réduction en fonction des méthodes que nous appliquons dans les étapes en amont de la chaîne de traitement afin d'obtenir les sacs de mots visuels.

La théorie sur laquelle nous nous sommes appuyés garantit que nous conservons un ensemble minimal d'attributs pour maintenir les descriptions des objets. En revanche, l'obtention d'un taux de réduction significatif n'est lui pas garanti. C'est pourquoi nous avons proposé une approche de réduction floue, algorithme RedAttsFloue, qui permet un taux de réduction plus important qu'avec l'algorithme RedAttsSansPerte au prix d'une perte d'information. Cette approche introduit le graphe de précédence flou où des attributs "similaires" sont reliés entre eux, la similarité dépendant d'un seuil de flexibilité. Nous avons donné une définition formelle du graphe de précédence flou et nous avons démontré ses propriétés. Cet algorithme est une extension de l'algorithme initial où sont supprimés les attributs réductibles et aussi les attributs "similaires", donc considérés comme apportant peu d'information aux traitements en aval de la chaîne, tels que la catégorisation ou le regroupement de données.

Apports, limites et perspectives

Contributions

Les méthodes de réduction présentées dans cette thèse possèdent les avantages suivants :
- Les algorithmes de réduction peuvent s'intégrer à un traitement supervisé ou non supervisé.
- L'algorithme RedAttsSansPerte propose une réduction exacte des attributs qui permet une réduction sans perte d'informations car la structure du treillis est maintenue et la taille de l'ensemble d'attributs obtenu est minimale pour cette propriété.
- La réduction est réalisée dans un temps polynomial en utilisant le graphe de précédence.
- Ce graphe est utilisé pour une extension au cas flou, avec un gain au niveau du taux de réduction d'attributs.

Néanmoins, nos algorithmes ont certaines limites.

Limites

Premièrement, la nécessité de binariser les données initiales fait perdre de l'information.

Deuxièmement, bien que nous puissions garantir que la réduction des attributs ne change pas la structure du treillis des concepts avec l'algorithme RedAttsSansPerte, nous ne garantissons pas l'obtention d'un taux de réduction significatif.

Troisièmement, l'algorithme RedAttsFloue, en utilisant la notion de graphe flou, apporte un taux de réduction plus élevé que l'algorithme RedAttsSansPerte mais a pour principale contrainte une perte d'information. Cette perte d'information entraîne une baisse de performances de classification (i.e. l'évaluation des méthodes par la classification). Toutefois, il existe des cas où la quantité d'attributs réduits est importante (*i.e.* avec un seuil de binarisation 0.5, la valeur de la Fraction d'Attributs réduits est 0.058% par l'algorithme RedAttsSansPerte contre 0.28% par l'algorithme RedAttsFloue) et la baisse de performances est faible (*i.e.* avec un seuil de binarisation 0.5, la valeur de la

F-mesure est 0.905 avec les données obtenues par l'algorithme RedAttsSansPerte contre 0.890 avec les données obtenues par l'algorithme RedAttsFloue). En somme, en fonction de l'application, le choix entre ces deux algorithmes peut varier.

Perspectives

Les algorithmes que nous avons proposés prennent des données binaires en entrée, cependant les données extraites d'images sont le plus souvent des données numériques. La transformation de données discrètes en données binaires s'appelle la binarisation (un cas particulier de la discrétisation). Le résultat de l'algorithme de réduction de dimension dépend de l'étape de discrétisation des attributs. En fonction de la table binaire obtenue après cette étape, nous n'obtenons pas le même treillis et ainsi les mêmes résultats de réduction d'attributs. Les différentes approches de discrétisation se catégorisent en deux types : discrétisation globale et discrétisation locale. Cependant, la discrétisation des attributs augmente le nombre des attributs, ce qui permet de bien représenter les données. Dans le cadre de nos expérimentations, où l'objectif est de réduire la taille de l'ensemble des attributs, nous ne voulons pas augmenter le nombre d'attributs avant de faire la réduction. C'est pour cela que nous avons utilisé une binarisation classique. Cette technique peut être catégorisée en discrétisation globale car elle utilise une seule condition (le seuil) pour binariser toutes les données.

Une des pistes d'amélioration possibles est la discrétisation locale [Girard 2009]. Un des points forts de notre algorithme de réduction est sa capacité de réduction dans les deux cas de l'apprentissage supervisé et non supervisé. Cependant, la discrétisation locale de Girard dépend de l'étiquetage [23] et de ce fait n'est applicable que dans le cas supervisé. Il existe des critères de discrétisation non supervisés [Kaytoue 2011] qui s'appuient sur les structures de patrons. Une autre alternative serait la génération des treillis directement à partir de données numériques (i.e. sans binarisation) en construisant des intervalles de valeurs des attributs, par exemple comme l'a réalisé [Polaillon 1998]. Cependant, cette dernière proposition n'est, à première vue, pas très attractive puisque notre objectif tend à la réduction du nombre des attributs, or, en construisant des intervalles, le nombre des attributs va augmenter. Une possibilité est que le nombre d'attributs réduits par l'algorithme de réduction soit supérieur au nombre d'attributs créés par la discrétisation. Cette hypothèse est intéressante mais reste à vérifier.

23. C'est la vérité terrain.

Une autre piste d'amélioration possible serait l'application de la logique floue directement à la théorie de treillis. Cependant, la théorie de la réduction dans le cadre flou possède des verrous scientifiques dont plusieurs pistes possibles sont en cours d'exploitation [Belohlavek 1999, Burusco 2000, Yahia 2001, Jaoua 2002, Krajci 2003, Georgescu 2004, Medina 2013]. Parmi elles, les auteurs de [Burusco 2000] définissent le concept flou en utilisant l'opérateur d'implication. Les auteurs de [Belohlavek 1999] définissent le treillis des concepts flou sur la structure générale. Le treillis flou que Belohlavek propose possède presque toutes les propriétés du treillis des concepts classiques. Cependant, dans le cadre flou, la réduction du treillis est un problème plus complexe, essentiellement parce qu'il y a deux opérations génératrices impliquées : l'intersection, qui est la seule opération impliquée dans le cas binaire, et le décalage, qui est dégénéré dans le cas binaire. En 2013, Belohlavek présente ses premiers résultats concernant ce problème [Belohlavek 2013].

En outre, l'algorithme RedAttsSansPerte peut être utilisé comme une méthode d'évaluation de la pertinence des méthodes d'extraction des mots visuels utilisés sur un ensemble de données. Autrement dit, il peut servir à évaluer la qualité des signatures des images.

Annexes

Annexe A

A.1 Mesures de la performance de classification

		Prediction outcome		Total
		p	n	
Actual value	p'	TP (True Positive)	FN (False Negative)	P'
	n'	FP (False Positive)	TN (True Negative)	N'
	Total	P	N	

TABLE A.1: Matrice de confusion.

A.1.1 F-mesure (F-score)

$$F = \frac{2 \times précision \times rappel}{précision + rappel} \tag{A.1}$$

où

$$\text{précision} = \frac{TP}{TP + FP} \tag{A.2}$$

$$\text{rappel} = \frac{TP}{TP + FN} \tag{A.3}$$

A.1.2 Balanced Error Rate (BER)

BER est la moyenne des erreurs de chaque classe.

$$\text{BER} = 0.5 * (\frac{FN}{TP + FN} + \frac{FP}{FP + TN}) \tag{A.4}$$

Annexe B

B.1 Résultat de réduction de l'algorithme RedAttsFloue

Arcene		Seuil de flexibilité								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de bina-risation	**0.0**	0.2680	0.3650	0.4487	0.5579	0.6627	0.7586	0.8651	0.9301	0.9696
	0.1	0.3115	0.3964	0.4513	0.4983	0.5332	0.5689	0.6145	0.6749	0.8201
	0.2	0.3224	0.4229	0.4786	0.5232	0.5715	0.6142	0.6482	0.6943	0.7606
	0.3	0.3115	0.4501	0.5239	0.5768	0.6072	0.6352	0.66049	0.6942	0.7350
	0.4	0.2981	0.4584	0.5255	0.5728	0.6056	0.6322	0.6566	0.6816	0.7092
	0.5	0.2526	0.4277	0.5218	0.5767	0.6098	0.6333	0.6524	0.6725	0.6960
	0.6	0.1902	0.3633	0.4771	0.5399	0.5891	0.63450	0.6604	0.6859	0.6961
	0.7	0.1494	0.3333	0.4559	0.5096	0.6054	0.6552	0.6782	0.6897	0.6973
	0.8	0.2420	0.4268	0.5478	0.6943	0.7452	0.7516	0.7643	0.7643	0.7771
	0.9	0.1515	0.4242	0.5152	0.6061	0.6364	0.6364	0.6667	0.6667	0.6667

TABLE B.1: Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de flexibilité et du seuil de binarisation.

Arcene		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de binarisation	**0**	0.1765	0.3413	0.4705	0.6611	0.9681	0.9957	0.9992	0.9967	0.9925
	0.1	0.1205	0.2265	0.2618	0.2634	0.8615	0.9914	0.9978	0.9969	0.9961
	0.2	0.1253	0.2429	0.3261	0.3944	0.8311	0.9888	0.9969	0.9986	0.9991
	0.3	0.1361	0.2943	0.3938	0.4916	0.8695	0.9926	0.993	0.9978	0.9987
	0.4	0.1446	0.317	0.4287	0.5834	0.8931	0.983	0.9924	0.9967	0.9982
	0.5	0.1224	0.2998	0.439	0.5535	0.8571	0.9691	0.985	0.992	0.9971
	0.6	0.1132	0.24	0.3799	0.5018	0.8223	0.9535	0.98	0.9838	0.9944
	0.7	0.1126	0.2471	0.3592	0.5313	0.7767	0.9111	0.9643	0.963	0.9747
	0.8	0.1261	0.2556	0.3944	0.5417	0.8	0.8974	0.8919	0.9189	0.9714
	0.9	0.0714	0.1579	0.25	0.5385	0.75	0.75	0.7273	0.8182	0.9091

TABLE B.2: Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Arcene en fonction du seuil de flexibilité et du seuil de binarisation.

Dexter		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de binarisation	**0**	0.0006	0.0031	nd	nd	nd	nd	nd	nd	nd
	0.1	0	0.0015	0.0043	0.0158	nd	nd	nd	nd	nd
	0.2	0	0.001	0.0038	0.0205	0.0534	0.0576	0.0729	0.0881	0.0996
	0.3	0	0	0	0.006	0.0836	0.0836	0.1131	0.1338	0.1496
	0.4	0	0	0	0.0025	0.0353	0.0359	0.0422	0.0472	0.051
	0.5	0	0	0	0	0.0043	0.0043	0.0051	0.0068	0.0068
	0.6	0	0	0	0	0	0	0	0	0
	0.7	0	0	0	0	0	0	0	0	0
	0.8	0	0	0	0	0	0	0	0	0
	0.9	0	0	0	0	0	0	0	0	0

TABLE B.3: Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Dexter en fonction du seuil de flexibilité et du seuil de binarisation.

Dexter		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de binarisation	**0**	0	0.0005	nd	nd	nd	nd	nd	nd	nd
	0.1	0	0.0006	0.002	0.0178	nd	nd	nd	nd	nd
	0.2	0	0.0033	0.0129	0.0895	0.8742	0.9909	0.9938	0.9979	0.9989
	0.3	0	0.0027	0.0153	0.0687	0.9118	0.9577	0.9858	0.9981	0.9987
	0.4	0	0.0013	0.0063	0.0354	0.908	0.9301	0.9671	0.9974	0.9987
	0.5	0	0	0.0009	0.0103	0.9408	0.9528	0.9811	0.9974	0.9983
	0.6	0	0	0	0.0015	0.9793	0.9896	0.9896	0.997	0.997
	0.7	0	0	0	0	0.9799	0.9833	0.9866	0.9933	0.9933
	0.8	0	0	0	0	0.9623	0.9717	0.9717	0.9811	0.9906
	0.9	0	0	0	0	0.7778	0.7778	0.7778	0.7778	0.8889

TABLE B.4: Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Dexter en fonction du seuil de flexibilité et du seuil de binarisation.

Gisette		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de binarisation	**0**	0	0	0	0	0	0	0	0.3449	0.3606
	0.1	0	0	0	0	0	0	0	0.3453	0.3561
	0.2	0	0	0	0	0	0	0.0003	0.0014	0.0024
	0.3	0.0158	0	0	0	0	0	0.0003	0.0014	0.0024
	0.4	0.0205	0.0534	0.0576	0.0729	0.0881	0.0996	0.0003	0.0014	0.0024
	0.5	0.006	0.0836	0.0836	0.1131	0.1338	0.1496	0.3398	0.3411	0.3418
	0.6	0.0025	0.0353	0.0359	0.0422	0.0472	0.051	0.3581	0.3658	0.3664
	0.7	0	0.0043	0.0043	0.0051	0.0068	0.0068	0.2364	0.3626	0.3709
	0.8	0	0	0	0	0	0	0.2936	0.3646	0.3689
	0.9	0	0	0	0	0	0	0.3174	0.365	0.3663

TABLE B.5: Fraction d'Attributs réduits par l'étape 4 (réduction floue) de l'algorithme RedAttsFloue sur l'ensemble de données Gisette en fonction du seuil de flexibilité et du seuil de binarisation.

Gisette		**Seuil de flexibilité**								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Seuil de binarisation	**0**	0	0	0.9623	0.9717	0.9717	0.9811	0.9906	0.0003	0.0083
	0.1	0	0	0.7778	0.7778	0.7778	0.7778	0.8889	0.0007	0.0056
	0.2	0	0	0	0	0	0	0	0.0169	0.0542
	0.3	0.0178	0	0	0	0	0	0	0.0169	0.0542
	0.4	0.0895	0.8742	0.9909	0.9938	0.9979	0.9989	0	0.0169	0.0542
	0.5	0.0687	0.9118	0.9577	0.9858	0.9981	0.9987	0.002	0.0165	0.055
	0.6	0.0354	0.908	0.9301	0.9671	0.9974	0.9987	0.0003	0.0179	0.0547
	0.7	0.0103	0.9408	0.9528	0.9811	0.9974	0.9983	0	0.0167	0.0555
	0.8	0.0015	0.9793	0.9896	0.9896	0.997	0.997	0	0.0152	0.049
	0.9	0	0.9799	0.9833	0.9866	0.9933	0.9933	0	0.0106	0.0439

TABLE B.6: Fraction d'Attributs réduits par l'étape 5 (clarification floue) de l'algorithme RedAttsFloue sur l'ensemble de données Gisette en fonction du seuil de flexibilité et du seuil de binarisation.

Annexe C

C.1 Isomorphisme de graphes

Dans le cadre de la théorie des graphes, un **isomorphisme de graphes f** entre les graphes G et H est une bijection entre les sommets de G et ceux de H, telle qu'une paire de sommets $\{u, v\}$ de G est une arête de G si et seulement si $\{f(u), f(v)\}$ est une arête de H.

C.2 Catégorisation des méthodes de réduction de dimension

C.2.1 Méthode statistique

Dans ce manuscrit, une méthode est catégorisée comme une méthode statistique lorsqu'elle utilise une fonction statistique pour décrire des données. Par exemple, la moyenne, le médiane, l'écart-type, etc.

C.2.2 Méthode logique

Dans ce manuscrit, une méthode logique fait référence à une méthode algébrique. Et une méthode algébrique fait référence à une méthode de résolution d'une équation impliquant deux ou plusieurs variables dans lesquelles l'une des variables est exprimée en fonction d'une des autres variables.

C.3 Matrice d'affinité (Affinity matrix)

Une matrice d'affinité A est une matrice qui contient des valeurs où une valeur A_{ij} est calculée par une mesure pour déterminer la distance ou la similarité entre les points i et j où $i \in I$, I est l'ensemble d'éléments en ligne I de la matrice A et $j \in J$, J est l'ensemble d'éléments en colonne J de la matrice A. La contrainte du choix d'une mesure de similarité que nous pouvons utiliser pour calculer la matrice d'affinité reste floue. Néanmoins, une mesure de similarité est choisie en fonction de la distribution de données dans l'espace de Hilbert sur laquelle la matrice d'affinité agit.

C.4 Indépendance statistique (Statistical independence)

Notons les variables aléatoires $y_1, y_2, ..., y_m$ avec la fonction de densité jointe de ces variables $f(y_1, y_2, ...y_m)$. Pour simplifier, supposons que la valeur moyenne de ces variables soit nulle. Les variables y_i sont (mutuellement) indépendantes si la fonction de densité peut être factorisée [Athanasios Papoulis 1991] :

$$f(y_1, y_2, ...y_m) = f_1(y_1)f_2(y_2)...f_m(y_m)$$

où $f_i(y_i)$ est la densité marginale de y_i. Pour différencier cette forme de l'indépendance avec les autres notions de l'indépendance, par exemple l'indépendance linéaire, cette propriété est appelé **l'indépendance statistique**.

L'indépendance statistique doit être distinguer avec la non-corrélation. **La non-corrélation**

entre deux variables y_i, y_j est définie comme suit :

$$E\{y_i y_j\} - E\{y_i\}E\{y_j\} = 0, \text{ pour } i \neq j.$$

L'indépendance est plus forte que la non-corrélation. En effet, si les variables y_i, y_j sont indépendantes, nous avons :

$$E\{g_1(y_i)g_2(y_j)\} - E\{g_1(y_i)\}E\{g_2(y_j)\} = 0, \text{ pour } i \neq j$$

pour toutes les fonctions g_1, g_2 [Athanasios Papoulis 1991]. Ces fonctions doivent être **mesurables** : $g_i : (X, \mathcal{X}) \leftrightarrow (Y, \mathcal{Y})$ où $(X, \mathcal{X})$ et $(Y, \mathcal{Y})$ sont les espaces mesurables, X, Y sont des ensembles d'éléments avec leurs tribus [1] $\mathcal{X}$, $\mathcal{Y}$ respectivement.

C.5 Distribution gaussienne

En théorie des probabilités et en statistique, la loi gaussienne (aussi appelée la loi normale) est l'une des lois de probabilité pour modéliser des phénomènes naturels issus de plusieurs événements aléatoires. La densité de probabilité de la loi gaussienne est la **distribution gaussienne** (également appelée fonction de masse de la loi normale).

1. $\sigma - algèbre$

Bibliographie

[Aggarwal 2012] Charu C Aggarwal et Chengxiang Zhai. *A survey of text classification algorithms*. In Mining text data, chapitre 6, page 524. Springer Science & Business Media, 2012.

[Aggarwal 2014] Charu C. Aggarwal. *A Survey of Stream Classification Algorithms*. In Data classification : algorithms and applications, chapitre 9, page 707. CRC Press, 2014.

[Aha 1996] David W. Aha et Richard L. Bankert. *A comparative evaluation of sequential feature selection algorithms*. In Doug Fisher et Hans-J. Lenz, editeurs, Lecture notes in statistics, volume 112 of *Lecture Notes in Statistics*, chapitre Learning f, pages 199–206. Springer New York, New York, NY, 1996.

[Aine 2007] Sandip Aine, P. P. Chakrabarti et Rajeev Kumar. *AWA*-a window constrained anytime heuristic search algorithm*. In International Joint Conferences on Artificial Intelligence Organization, pages 2250–2255. Morgan Kaufmann Publishers Inc., jan 2007.

[Andrews 2009] S Andrews et Simon Andrews. *In-Close, a fast algorithm for computing formal concepts*. In International Conference on Conceptual Structures (ICCS), Moscow, 2009.

[Anily 1987] S. Anily et A. Federgruen. *Simulated Annealing Methods with General Acceptance Probabilities on JSTOR*. Journal of applied probability, vol. 24, no. 3, pages 657–667, 1987.

[Archibald 2007] Rick Archibald et George Fann. *Feature Selection and Classification of Hyperspectral Images With Support Vector Machines*. IEEE Geoscience and Remote Sensing Letters, vol. 4, no. 4, pages 674–677, oct 2007.

[Arthur 2009] David Arthur et Sergei Vassilvitskii. *Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method*. SIAM Journal on Computing, vol. 39, no. 2, pages 766–782, 2009.

[Athanasios Papoulis 1991] Athanasios Papoulis. Probability, random variables, and stochastic processes. McGraw-Hill, Inc., 3 édition, 1991.

[Awad 2015] Dounia Awad. *Toward a perceptual object recognition system.* ELCVIA Electronic Letters on Computer Vision and Image Analysis, vol. 14, no. 3, dec 2015.

[Baklouti 2005] Fatma Baklouti, Gérard Lévy et Richard Emilion. *A fast and general algorithm for Galois lattices building.* Journal of Symbolic Data Analysis, vol. 3, pages 19–31, 2005.

[Barbut 1970] Marc Barbut et Bernard Monjardet. Ordre et classification : algèbre et combinatoire. Hachette, Paris, 1970.

[Battiti 1994] R Battiti. *Using mutual information for selecting features in supervised neural net learning.* IEEE transactions on neural networks, vol. 5, no. 4, pages 537–550, jan 1994.

[Baudat 2000] G. Baudat et F. Anouar. *Generalized Discriminant Analysis Using a Kernel Approach.* Neural Computation, vol. 12, no. 10, pages 2385–2404, oct 2000.

[Bay 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars et Luc Van Gool. *SURF : Speeded Up Robust Features.* Computer Vision and Image Understanding (CVIU), vol. 110, no. 3, pages 346–359, 2008.

[Bekkerman 2003] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby et Yoad Winter. *Distributional word clusters vs. words for text categorization.* The Journal of Machine Learning Research, vol. 3, pages 1183–1208, mar 2003.

[Belkin 2001] Mikhail Belkin et Partha Niyogi. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering.* In The 14th International Conference on Neural Information Processing Systems : Natural and Synthetic, pages 585–591, 2001.

[Bell 1997] Anthony J. Bell et Terrence J. Sejnowski. *The "Independent Components" of Natural Scenes are Edge Filters.* Vision Research, vol. 37, no. 23, pages 3327–3338, 1997.

[Belohlavek 1999] Radim Belohlavek. *Fuzzy Galois Connections.* Mathematical Logic Quarterly, vol. 45, no. 4, pages 497–504, 1999.

[Belohlavek 2010] Radim Belohlavek, Rudolf Kruse et Vilem Vychodil. *Discovery of optimal factors in binary data via a novel method of matrix decomposition.* Journal of Computer and System Sciences, vol. 76, no. 1, pages 3–20, 2010.

[Belohlavek 2013] Radim Belohlavek et Jan Konecny. *Toward reduction of formal fuzzy context.* In Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pages 221–225. IEEE, jun 2013.

[Bernard 1987] Paul-Marie Bernard et Claude Lapointe. Mesures statistiques en épidémiologie. Presses de l'Université du Québec, 1987.

[Berry 2005] Anne Berry, Marianne Huchard, Ross M. McConnell, Alain Sigayret et Jeremy Spinrad. *Efficiently Computing a Linear Extension of the Sub-hierarchy of a Concept Lattice*. In International Conference on Formal Concept Analysis (ICFCA), pages 208–222. Springer, 2005.

[Berry 2014] Anne Berry, Alain Gutierrez, Marianne Huchard, Amedeo Napoli et Alain Sigayret. *Hermes : a simple and efficient algorithm for building the AOC-poset of a binary relation*. Annals of Mathematics and Artificial Intelligence, vol. 72, no. 1-2, pages 45–71, oct 2014.

[Bertet 2012] Karell Bertet. *The dependence graph of a lattice*. In International Conference on Concept Lattices and their Applications (CLA), pages 223–231, Malaga, 2012.

[Birkhoff 1940] Garrett Birkhoff. Lattice Theory. American Mathematical Society, 1st édition, 1940.

[Bisiani 1987] R. Bisiani. *Beam Search*. In S.C. Shapiro, editeur, Encyclopedia of Artificial Intelligence, chapitre Beam searc, pages 56–58. John Wiley & Sons, encycloped édition, 1987.

[Blum 1997] Avrim L. Blum et Pat Langley. *Selection of relevant features and examples in machine learning*. Artificial Intelligence, vol. 97, no. 1-2, pages 245–271, dec 1997.

[Bolón-Canedo 2014] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño et A. Alonso-Betanzos. *A framework for cost-based feature selection*. Pattern Recognition, vol. 47, no. 7, pages 2481–2489, jul 2014.

[Bolovinou 2012] A. Bolovinou, I. Pratikakis et S. Perantonis. *Bag of spatio-visual words for context inference in scene classification*. Pattern Recognition, vol. 46, no. 3, pages 1053–1039, sep 2012.

[Bordat 1986] J. P. Bordat. *Calcul pratique du treillis de Galois d'une correspondance*. Mathématiques et Sciences Humaines, vol. 96, pages 31–47, 1986.

[Bosch 2006] Anna Bosch, Andrew Zisserman et Xavier Munoz. *Scene Classification Via pLSA*. In Aleš Leonardis, Horst Bischof et Axel Pinz, editeurs, 9th European Conference on Computer Vision, volume 3954 of *Lecture Notes in Computer Science*, pages 517–530, Graz, Austria, 2006.

[Bouman 2005] Charles A. Bouman, Michael Shapiro, Gregory W. Cook, C. Brian Atkins, Hui Cheng, Jennifer G. Dy et Sean Borman. *CLUSTER : An Unsupervised Algorithm for Modeling Gaussian Mixtures*, 2005.

[Boyle 1988] R. D. Boyle et R. C. Thomas. Computer vision : a first course. Blackwell Scientific Publications, Ltd., may 1988.

[Breiman 1984] Leo Breiman, Jerome Friedman, Charles J. Stone et R. A. Olshen. Classification and regression trees. 1984.

[Breiman 1996] Leo Breiman. *Bagging Predictors*. Machine Learning, vol. 24, no. 2, pages 123–140, 1996.

[Breiman 2001] Leo Breiman. *Random Forests*. Machine Learning, vol. 45, no. 1, pages 5–32, 2001.

[Burusco 2000] A. Burusco et R. Fuentes-González. *Concept lattices defined from implication operators*. Fuzzy Sets and Systems, vol. 114, no. 3, pages 431–436, sep 2000.

[Canny 1986] John Canny. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pages 679–698, nov 1986.

[Cao 2010] Yang Cao, Changhu Wang, Zhiwei Li, Liqing Zhang et Lei Zhang. *Spatial-bag-of-features*. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3352–3359. IEEE, jun 2010.

[Caspard 2003] Nathalie Caspard et Bernard Monjardet. *The lattices of closure systems, closure operators, and implicational systems on a finite set : a survey*. Discrete Applied Mathematics, vol. 127, no. 2, pages 241–269, 2003.

[Cattell 1966] Raymond B. Cattell. *The Scree Test For The Number Of Factors*. Multivariate Behavioral Research, vol. 1, no. 2, pages 245–276, apr 1966.

[Chandrashekar 2014] Girish Chandrashekar et Ferat Sahin. *A survey on feature selection methods*. Computers & Electrical Engineering, vol. 40, no. 1, pages 16–28, jan 2014.

[Chatfield 2011] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi et Andrew Zisserman. *The devil is in the details : an evaluation of recent feature encoding methods*. In Jesse Hoey, Stephen McKenna et Emanuele Trucco, editeurs, Proceedings of the British Machine Vision Conference, pages 76.1–76.12. BMVA Press, 2011.

[Chung 1997] Fan R. K. Chung. Spectral Graph Theory, Numéro 92. AMS, second édition, 1997.

[Colomb 2011] Pierre Colomb, Alexis Irlande, Olivier Raynaud et Yoan Renaud. *A closure algorithm using a recursive decomposition of the set of Moore co-families*. In International Conference on Concept Lattices and their Applications (CLA), pages 131–141, 2011.

[Comon 1994] Pierre Comon. *Independent component analysis, A new concept ?* Signal Processing, vol. 36, pages 287–314, 1994.

[Cook 1993] Dianne Cook, Andreas Buja et Javier Cabrera. *Projection Pursuit Indexes Based on Orthonormal Function Expansions*. Journal of Computational and Graphical Statistics, vol. 2, no. 3, page 225, sep 1993.

[Coustaty 2011] Mickaël Coustaty. *Contribution à l'analyse complexe de documents anciens, application aux lettrines*. PhD thesis, University of La Rochelle, 2011.

[Cruz 1998] J. R. Cruz et C. C. Y. Dorea. *Simple conditions for the convergence of simulated annealing type algorithms.* Journal of Applied Probability, vol. 35, no. 4, pages 885–892, dec 1998.

[Dao 2014] Ngoc Bich Dao, Karell Bertet et Arnaud Revel. *Reduction dimension of bags of visual words with FCA.* In Concept Lattices and their Applications, volume 1252, pages 219–230, Kosice, Slovakia, sep 2014.

[Dao 2016] Ngoc Bich Dao, Sebastien Eskenazi, Karell Bertet et Arnaud Revel. *A fuzzy precedence graph definition for algebra-based dimension reduction.* In IEEE World Congress on Computational Intelligence - International Conference of Fuzz-IEEE, pages 1826–1833, Vancouver, Canada, jul 2016.

[Dash 1997] M. Dash et H. Liu. *Feature selection for classification.* Intelligent data analysis, vol. 1, no. 3, pages 131–156, 1997.

[Debuse 1997] Justin C.W. Debuse et Victor J. Rayward-Smith. *Feature Subset Selection within a Simulated Annealing Data Mining Algorithm.* Journal of Intelligent Information Systems, vol. 9, no. 1, pages 57–81, 1997.

[DeSarbo 1984] Wayne S. DeSarbo, J. Douglas Carroll, Linda A. Clark et Paul E. Green. *Synthesized clustering : A method for amalgamating alternative clustering bases with differential weighting of variables.* Psychometrika, vol. 49, no. 1, pages 57–78, mar 1984.

[Devaney 1997] Mark Devaney et Ashwin Ram. *Efficient feature selection in conceptual clustering.* In The 4th International Conference on Machine Learning, pages 92–97, Nashville, TN, 1997. Morgan Kaufmann.

[Dicky 1995] Hervé Dicky, Christophe Dony, Marianne Huchard et Thérèse Libourel. *ARES, Adding a class and REStructuring Inheritance Hierarchy.* In Bases de Données Avancées (BDA), pages 25–42, 1995.

[Ding 2005] Chris Ding et Hanchuan Peng. *Minimum redundancy feature selection from microarray gene expression data.* Journal of Bioinformatics and Computational Biology, vol. 03, no. 02, pages 185–205, apr 2005.

[Doak 1992] Justin Doak. *CSE-92-18 - An Evaluation of Feature Selection Methodsand Their Application to Computer Security.* Rapport technique, jan 1992.

[Duygulu 2002] Pinar Duygulu, Kobus Barnard, J. F. G. de Freitas et David A. Forsyth. *Object Recognition as Machine Translation : Learning a Lexicon for a Fixed Image Vocabulary.* In European Conference on Computer Vision (ECCV), pages 97–112, Copenhagen, 2002.

[Dy 2000] Jennifer G. Dy et Carla E. Brodley. *Feature Subset Selection and Order Identification for Unsupervised Learning.* In Pat Langley, editeur, The17th International Conference on Machine Learning, pages 247–254, San Francisco, 2000. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©2000.

[Dy 2004] Jennifer G. Dy et Carla E. Brodley. *Feature Selection for Unsupervised Learning*. Journal of Machine Learning Research, vol. 5, pages 845–889, 2004.

[Elghazel 2010] Haytham Elghazel et Alex Aussem. *Feature Selection for Unsupervised Learning Using Random Cluster Ensembles*. In IEEE International Conference on Data Mining, pages 168–175. IEEE, dec 2010.

[Elghazel 2013] Haytham Elghazel et Alex Aussem. *Unsupervised feature selection with ensemble learning*. Machine Learning, vol. 98, no. 1-2, pages 157–180, apr 2013.

[Eskenazi 2017] Sébastien Eskenazi, Petra Gomez-Krämer et Jean-Marc Ogier. *A comprehensive survey of mostly textual document segmentation algorithms since 2008*. Pattern Recognition, vol. 64, pages 1–14, apr 2017.

[Everingham 2012] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn et A. Zisserman. *The PASCAL Visual Object Classes (VOC) Challenge*. International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, 2012.

[Farquhar 2005] J. D. R. Farquhar, Sandor Szedmak, Hongying Meng et John Shawe-taylor. *Improving "bag-of-keypoints" image categorisation : Generative Models and PDF-Kernels*. Rapport technique, University of Southampton, 2005.

[Fisher 1936] R. A. Fisher. *The use of multiple measurements in taxonomic problems*. The Annals of Eugenics, vol. 7, pages 179–188, 1936.

[Fisher 1987] Douglas H. Fisher. *Knowledge Acquisition Via Incremental Conceptual Clustering*. Machine Learning, vol. 2, no. 2, pages 139–172, sep 1987.

[Fodor 2002] Imola K. Fodor. *A survey of dimension reduction techniques*. Rapport technique, Lawrence Livermore National Lab., CA (US), 2002.

[Fong 1995] Philip W. L. Fong. *A Quantitative Study of Hypothesis Selection*. In The 12th International Conference on Machine Learning (ICML), pages 226–234, 1995.

[Forman 2003] George Forman. *An extensive empirical study of feature selection metrics for text classification*. The Journal of Machine Learning Research, vol. 3, pages 1289–1305, mar 2003.

[Foroutan 1987] Iman Foroutan et Jack Sklansky. *Feature Selection for Automatic Classification of Non-Gaussian Data*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 17, no. 2, pages 187–198, 1987.

[Fred 2005] Ana L N Fred et Anil K Jain. *Combining multiple clusterings using evidence accumulation*. IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 6, pages 835–50, jun 2005.

[Friedman 1974] Jerome H. Friedman et John W. Tukey. *A projection pursuit algorithm for exploratory data analysis*. IEEE transactions on computers, vol. C-23, no. 9, pages 881–890, 1974.

[Friedman 1987] Jerome H. Friedman. *Exploratory Projection Pursuit*. Journal of the American Statistical Association, vol. 82, no. 397, pages 249–266, 1987.

[Frigui 1997] Hichem Frigui et Raghu Krishnapuram. *Clustering by competitive agglomeration*. Pattern Recognition, vol. 30, no. 7, pages 1109–1119, jul 1997.

[Frigui 2004] Hichem Frigui et Olfa Nasraoui. *Unsupervised learning of prototypes and attribute weights*. Pattern Recognition, vol. 37, no. 3, pages 567–581, mar 2004.

[Frohlich 2003] H. Frohlich, O. Chapelle et B. Scholkopf. *Feature selection for support vector machines by means of genetic algorithm*. In The 15th IEEE International Conference on Tools with Artificial Intelligence, pages 142–148. IEEE Comput. Soc, 2003.

[Fu 2004] Huaiguo Fu et Engelbert Mephu Nguifo. *A Parallel Algorithm to Generate Formal Concepts for Large Data*. In International Conference on Formal Concept Analysis (ICFCA), pages 394–401. Springer Berlin Heidelberg, 2004.

[Fukunaga 1990] Keinosuke Fukunaga. Introduction to Statistical Pattern Recognition. 1990.

[Furcy 2005] David Furcy et Sven Koenig. *Scaling up WA* with commitment and diversity*. In International Joint Conference on Artificial Intelligence, volume 19, pages 1521–1522. Morgan Kaufmann Publishers Inc., jul 2005.

[Ganter 1984] B Ganter. Two basic algorithms in concept analysis. Numéro 831 de Preprint. Techn. Hochsch., Fachbereich Mathematik, 1984.

[Ganter 1999] Bernhard Ganter et Rudolf Wille. Formal Concept Analysis : Mathematical Foundations. Springer-Verlag, Berlin, 1st editio édition, dec 1999.

[Gély 2005] Alain Gély. *A Generic Algorithm for Generating Closed Sets of a Binary Relation*. In International Conference on Formal Concept Analysis (ICFCA), pages 223–234. Springer Berlin Heidelberg, 2005.

[Gennari 1989] John H. Gennari, Pat Langley et Doug Fisher. *Models of incremental concept formation*. Artificial Intelligence, vol. 40, no. 1-3, pages 11–61, sep 1989.

[Georgescu 2004] George Georgescu et Andrei Popescu. *Non-dual fuzzy connections*. Archive for Mathematical Logic, vol. 43, no. 8, pages 1009–1039, nov 2004.

[Gevers 1996] Th. Gevers et A.W.M. Smeulders. *A comparative study of several color models for color image invariant retrieval*. In International Workshop on Image Database and Multimedia Search, pages 17–23. University of Amsterdam, 1996.

[Ghahramani 1997] Zoubin Ghahramani et Geoffrey E. Hinton. *The EM algorithm for Mixtures of Factor Analyzers*. Rapport technique, University of Toronto, Toronto, Canada, 1997.

[Girard 2009] Nathalie Girard, Karell Bertet et Muriel Visani. *Vers une discrétisation locale pour les treillis dichotomiques*. In XVIèmes Rencontres de la Société Francophone de Classification, pages 113–116, sep 2009.

[Gluck 1985] Mark A. Gluck et James E. Corter. *Information Uncertainty and the Utility of Categories*. In The 7th Annual Conference of Cognitive Science Society, pages 283–287, 1985.

[Godin 1993] Robert Godin et Hafedh Mili. *Building and maintaining analysis-level class hierarchies using Galois Lattices*. ACM SIGPLAN Notices, vol. 28, no. 10, pages 394–410, oct 1993.

[Godin 1995] Robert Godin, Rokia Missaoui et Hassan Alaoui. *Incremental concept formation algorithms based on Galois (concept) lattices*. Appeared in Computational Intelligence, vol. 11, no. 2, pages 246–267, 1995.

[Griffin 2007] Greg Griffin, Alex D. Holub et Pietro Perona. *Caltech-256 Object Category Dataset*. Rapport technique, 2007.

[Grissa 2016] Dhouha Grissa, Mélanie Pétéra, Marion Brandolini, Amedeo Napoli, Blandine Comte et Estelle Pujos-Guillot. *Feature Selection Methods for Early Predictive Biomarker Discovery Using Untargeted Metabolomic Data.* Frontiers in molecular biosciences, vol. 3, page 30, 2016.

[Gu 2011] Quanquan Gu, Zhenhui Li et Jiawei Han. Linear Discriminant Dimensionality Reduction, volume 6911 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Guerin 2013] Clement Guerin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karell Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier et Arnaud Revel. *eBDtheque : A Representative Database of Comics*. In 2013 12th International Conference on Document Analysis and Recognition, pages 1145–1149. IEEE, aug 2013.

[Gutierrez-Osuna 2002] R. Gutierrez-Osuna. *Pattern analysis for machine olfaction : a review*. IEEE Sensors Journal, vol. 2, no. 3, pages 189–202, jun 2002.

[Guyon 2002] Isabelle Guyon, Jason Weston, Stephen Barnhill et Vladimir Vapnik. *Gene Selection for Cancer Classification using Support Vector Machines*. Machine Learning, vol. 46, no. 1-3, pages 389–422, 2002.

[Guyon 2003] Isabelle Guyon. *Design of experiments for the NIPS 2003 variable selection benchmark*. Rapport technique, 2003.

[Hall 1997] Mark A. Hall et Lloyd A. Smith. *Feature Subset Selection : A Correlation Based Filter Approach*. In International Conference on Neural Information Processing and Intelligent Information Systems, pages 855–858. Berlin : Springer, 1997.

[Hall 1999] Mark A. Hall. *Correlation-based feature subset selection for machine learning*. Doctor of philosophy, University of Waikato, Hamilton, NewZealand, 1999.

[Haralick 1973] Robert M. Haralick, K. Shanmugam et Its'hak Dinstein. *Textural features for image classification*. IEEE Transactions on Systems, Man and Cybernetics, vol. 3, no. 6, pages 610–621, 1973.

[Harris 1954] Zellig S. Harris. *Distributional structure.* Word, vol. 10, no. 2-3, pages 146–162, 1954.

[Harris 1988] Chris Harris et Mike Stephens. *A combined corner and edge detector.* In The 4th Alvey Vision Conference, pages 147–151, 1988.

[He 2004] Xiaofei He et Partha Niyogi. *Locality Preserving Projections.* In S. Thrun, L. K. Saul et B. Scholkopf, editeurs, Advances in Neural Information Processing Systems 16 (NIPS 2003), volume 16, pages 153–160. MIT Press, 2004.

[He 2005] Xiaofei He, Deng Cai et Partha Niyogi. *Laplacian score for feature selection.* In Advances in Neural Information Processing Systems 18, pages 507–514. MIT Press, 2005.

[Ho 1998] Tin Kam Ho. *The random subspace method for constructing decision forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pages 832–844, 1998.

[Hong 2008] Yi Hong, Sam Kwong, Yuchou Chang et Qingsheng Ren. *Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm.* Pattern Recognition, vol. 41, no. 9, pages 2742–2756, sep 2008.

[Hotelling 1933] H. Hotelling. *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology, vol. 24, no. 6, pages 417–441, 1933.

[Hou 2010] Jian Hou, Jianxin Kang et Naiming Qi. *On Vocabulary Size in Bag-of-Visual-Words Representation.* In Guoping Qiu, Kin Man Lam, Hitoshi Kiya, Xiang-Yang Xue, C.-C. Jay Kuo et Michael S. Lew, editeurs, The 11th Pacific Rim Conference on Multimedia, pages 414–424, Shanghai, China, September 21-24, 2010, 2010. Springer Berlin Heidelberg.

[Hu 1962] M. K. Hu. *Visual pattern recognition by moment invariants.* IRE Transaction Information Theory, vol. IT, no. 8, pages 179–187, 1962.

[Huang 2005] Joshua Zhexue Huang, Michael K Ng, Hongqiang Rong et Zichen Li. *Automated variable weighting in k-means type clustering.* IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 5, pages 657–68, may 2005.

[Huber 1985] Peter J. Huber. *Projection Pursuit.* The Annals of Statistics, vol. 13, no. 2, pages 435–475, jun 1985.

[Huiskes 2008] Mark J. Huiskes et Michael S. Lew. *The MIR flickr retrieval evaluation.* In The 1st ACM international conference on Multimedia information retrieval (MIR '08), pages 39–43, New York, USA, oct 2008. ACM Press.

[Hunt 1966] Earl B. Hunt, Janet Marin et Philip J. Stone. Experiments in induction. 1966.

[Hyvärinen 1997] Aapo Hyvärinen. *New approximations of differential entropy for independent component analysis and projection pursuit.* In The conference on Advances

in neural information processing systems 10 (NIPS1997), pages 273–279, Denver, Colorado, 1997. MIT Press.

[Hyvarinen 1999] Aapo Hyvarinen. *Survey on independent component analysis.* Neural Computing Surveys, vol. 2, pages 94–128, 1999.

[Ikeda 2014] Madori Ikeda et Akihiro Yamamoto. *Local Feature Selection by Formal Concept Analysis for Multi-class Classification.* In Peng WC. et al., editeur, Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 470–482. Springer, Cham, 2014.

[Itseez 2014] Itseez. *The OpenCV Reference Manual*, 2014.

[Itseez 2015] Itseez. *Open Source Computer Vision Library*, 2015.

[J. Sivic 2003] J. Sivic et A. Zisserman. *Video Google : a text retrieval approach to object matching in videos.* In The 9th IEEE International Conference on Computer Vision, volume 2, pages 1470–1477. IEEE, 2003.

[Jaoua 2002] Ali Jaoua et Samir Elloumi. *Galois connection, formal concepts and Galois lattice in real relations : application in a real classifier.* Journal of Systems and Software, vol. 60, no. 2, pages 149–163, 2002.

[Jiang 2007] Yu-Gang Jiang, Chong-Wah Ngo et Jun Yang. *Towards optimal bag-of-features for object categorization and semantic video retrieval.* In The 6th ACM international conference on Image and video retrieval - CIVR '07, pages 494–501, New York, New York, USA, jul 2007. ACM Press.

[Johnstone 2009] Iain M Johnstone et D Michael Titterington. *Statistical challenges of high-dimensional data.* Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, vol. 367, no. 1906, pages 4237–53, nov 2009.

[Jones 1987] M. C. Jones et Robin Sibson. *What is Projection Pursuit ?* Journal of the Royal Statistical Society. Series A (General), vol. 150, no. 1, pages 1–37, 1987.

[Kambhatla 1997] Nandakishore Kambhatla et Todd K. Leen. *Dimension Reduction by Local Principal Component Analysis.* Neural Computation, vol. 9, pages 1493–1516, 1997.

[Kass 1980] G. V. Kass. *An Exploratory Technique for Investigating Large Quantities of Categorical Data.* Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 29, no. 2, pages 119–127, 1980.

[Kaytoue 2011] Mehdi Kaytoue. *Mining numerical data with formal concept analysis and pattern structures.* PhD thesis, Universiy Henri Poincaré - Nancy 1, 2011.

[Khotanzad 1990] Alireza Khotanzad et Yaw Hua Hong. *Invariant Image Recognition by Zernike Moments.* IEEE transactions on pattern analysis and machine intelligence, vol. 12, no. 5, pages 489–497, 1990.

[Kirkpatrick 1983] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi. *Optimization by simulated annealing.* Science, vol. 220, no. 4598, pages 671–680, 1983.

[Kohavi 1997] Ron Kohavi et George H. John. *Wrappers for feature subset selection.* Artificial Intelligence, vol. 97, no. 1-2, pages 273–324, dec 1997.

[Kononenko 1994] Igor Kononenko. *Estimating attributes : Analysis and extensions of RELIEF.* In Francesco Bergadano et Luc Raedt, editeurs, ECML, volume 784 of *Lecture Notes in Computer Science*, pages 171–182, 1994.

[Koyutürk 2003] Mehmet Koyutürk et Ananth Grama. *PROXIMUS : A framework for analyzing very high dimensional discrete-attributed datasets.* In The 9th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03), pages 147–156, New York, USA, 2003. ACM Press.

[Krajca 2008] Petr Krajca, Jan Outrata et Vilem Vychodil. *Parallel Recursive Algorithm for FCA.* In International Conference on Concept Lattice and their Applications (CLA), pages 71–82, 2008.

[Krajci 2003] S. Krajci. *Cluster based efficient generation of fuzzy concepts.* Neural network world, vol. 13, no. 5, pages 521–530, 2003.

[Kwak 2002] N Kwak et Chong-Ho Choi. *Input feature selection for classification problems.* IEEE Transactions on Neural Networks, vol. 13, no. 1, pages 143–59, jan 2002.

[Land 1960] A. H. Land et A. G. Doig. *An automatic method of solving discrete programming problems.* Econometrica, vol. 28, no. 3, pages 497–520, 1960.

[Langley 1994] Pat Langley et Stephanie Sage. *Induction of selective Bayesian classifiers.* In The 10th international conference on Uncertainty in Artificial intelligence (UAI'94), pages 399–406. Morgan Kaufmann Publishers Inc., jul 1994.

[Lazebnik 2006] Svetlana Lazebnik, Cordelia Schmid et Jean Ponce. *Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories.* In Computer Vision and Pattern Recognition (CVPR), volume 2, pages 169–178, 2006.

[Leardi 1992] R. Leardi, R. Boggia et M. Terrile. *Genetic algorithms as a strategy for feature selection.* Journal of Chemometrics, vol. 6, no. 5, pages 267–281, sep 1992.

[Leardi 2000] Riccardo Leardi. *Application of genetic algorithm-PLS for feature selection in spectral data sets.* Journal of chemometrics, vol. 14, pages 643–655, 2000.

[Leblanc 2000] Hervé Leblanc. *Sous-hiérarchie de Galois : un modèle pour la construction et l'évolution des hiérarchies d'objets.* PhD thesis, University Montpellier II, 2000.

[LeCun 1998] Y. LeCun, L. Bottou, Y. Bengio et P. Haffner. *Gradient-based learning applied to document recognition.* In Proceedings of the IEEE, volume 86, pages 2278–2324, 1998.

[Lei Xu 1988] Lei Xu, Pingfan Yan et Tong Chang. *Best first strategy for feature selection.* In The 9th International Conference on Pattern Recognition, pages 706–708. IEEE Comput. Soc. Press, 1988.

[Lewis 1992] David D. Lewis. *An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task*. In ACM, editeur, The 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 37–50, Denmark, 1992. ACM.

[Lewis 1997] David D. Lewis. *Reuters-21578, Distribution 1.0*, 1997.

[Li 2008] Yuanhong Li, Ming Dong et Jing Hua. *Localized feature selection for clustering*. Pattern Recognition Letters, vol. 29, no. 1, pages 10–18, 2008.

[Lin 2008] Shih-Wei Lin, Zne-Jung Lee, Shih-Chieh Chen et Tsung-Yuan Tseng. *Parameter determination of support vector machine and feature selection using simulated annealing approach*. Applied Soft Computing, vol. 8, no. 4, pages 1505–1512, sep 2008.

[Lindeberg 1993] Tony Lindeberg. *Detecting salient blob-like image structures and their scales with a scale-space primal sketch : A method for focus-of-attention*. International Journal of Computer Vision, vol. 11, no. 3, pages 283–318, dec 1993.

[Lindeberg 1998] Tony Lindeberg. *Feature Detection with Automatic Scale Selection*. International Journal of Computer Vision, vol. 30, no. 2, pages 79–116, nov 1998.

[Lindig 2000] Christian Lindig. *Fast concept analysis*. In G. Stumme, editeur, International Conference on Conceptual Structures (ICCS)2, pages 235–248, Aachen, Germany, 2000. Shaker Verlag.

[Littlestone 1988] Nick Littlestone. *Learning Quickly When Irrelevant Attributes Abound : A New Linear-Threshold Algorithm*. Machine Learning, vol. 2, no. 4, pages 285–318, 1988.

[Liu 1996] Huan Liu et Rudy Setiono. *Feature Selection And Classification - A Probabilistic Wrapper Approach*. In The 9th International Conference on Industrial and Engineering Applications of AI and ES, pages 284–292, 1996.

[Lloyd 1982] S. Lloyd. *Least squares quantization in PCM*. IEEE Transactions on Information Theory, vol. 28, no. 2, pages 129–137, mar 1982.

[Lopez de Mantaras 1989] R. Lopez de Mantaras. *ID3 revisited : A distance based criterion for attribute selection*. Methodologies for Intelligent Systems, vol. 4, pages 342–350, 1989.

[Lotte 2007] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche et Bruno Arnaldi. *A review of classification algorithms for EEG-based brain–computer interfaces*. Journal of Neural Engineering, vol. 4, pages 24–48, 2007.

[Loughrey 2004] John Loughrey et Padraig Cunningham. *Overfitting in wrapper-based feature subset selection : the harder you try the worse it gets*. In Max Bramer, Frans Coenen et Tony Allen, editeurs, The 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004), pages 33–43, London, 2004. Springer London.

[Lowe 1999] David G. Lowe. *Object recognition from local scale-invariant features.* In The International Conference on Computer Vision, pages 1150–1157, Kerkyra, 1999.

[Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, nov 2004.

[Lu 2008] Haibing Lu, Jaideep Vaidya et Vijayalakshmi Atluri. *Optimal Boolean Matrix Decomposition : Application to Role Engineering.* In 2008 IEEE 24th International Conference on Data Engineering, pages 297–306. IEEE, apr 2008.

[Lu 2011] Haibing Lu. *Boolean matrix decomposition and extension with applications.* PhD thesis, Rutgers University Community, 2011.

[Maillot 2006] Nicolas Maillot, Jean-Pierre Chevallet, Vlad Valea et Joo Hwee Lim. *IPAL Inter-Media Pseudo-Relevance Feedback Approach to ImageCLEF 2006 Photo Retrieval.* Rapport technique, 2006.

[Mardia 1979] K. V. Mardia, John T. Kent et John M. Bibby. Multivariate analysis. Academic Press, 1979.

[Marill 1963] T. Marill et D. Green. *On the effectiveness of receptors in recognition systems.* IEEE Transactions on Information Theory, vol. 9, no. 1, pages 11–17, jan 1963.

[Maron 1994] Oded Maron et Andrew W. Moore. *Hoeffding Races : Accelerating Model Selection Search for Classification and Function Approximation.* In Advances in Neural Information Processing Systems, pages 59–66, 1994.

[Martinez 2001] Aleix M. Martinez et Avinash C. Kak. *PCA versus LDA.* IEEE Transactions on Pattern Analysis and Machine Inteligence, vol. 23, no. 2, pages 228–233, 2001.

[Medina 2013] J. Medina et M. Ojeda-Aciego. *Dual multi-adjoint concept lattices.* Information Sciences, vol. 225, pages 47–54, mar 2013.

[Meiri 2006] Ronen Meiri et Jacob Zahavi. *Using simulated annealing to optimize the feature selection problem in marketing applications.* European Journal of Operational Research, vol. 171, no. 3, pages 842–858, jun 2006.

[Metropolis 1953] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller et Edward Teller. *Equation of State Calculations by Fast Computing Machines.* The Journal of Chemical Physics, vol. 21, no. 6, page 1087, dec 1953.

[Miettinen 2010] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das et Heikki Mannila. *The Discrete Basis Problem.* In IEEE Transactions on Knowledge and Data Engineering, pages 1348–1362, 2010.

[Mikolajczyk 2001] K. Mikolajczyk et C. Schmid. *Indexing based on scale invariant interest points.* In The 8th IEEE International Conference on Computer Vision (ICCV), volume 1, pages 525–531. IEEE Comput. Soc, 2001.

[Mikolajczyk 2004] Krystian Mikolajczyk. *Scale & affine invariant interest point detectors.* International Journal of Computer Vision, vol. 60, no. 1, pages 63–86, oct 2004.

[Mindru 2004] Florica Mindru, Tinne Tuytelaars, Luc Van Gool et Theo Moons. *Moment invariants for recognition under changing viewpoint and illumination.* Computer Vision and Image Understanding, vol. 94, no. 1-3, pages 3–27, 2004.

[Minsky 1969] Marvin Minsky et Seymour A. Papert. Perceptrons : An introduction to computational geometry. The MIT Press, expanded e édition, 1969.

[Mishra 2011] Debahuti Mishra, Rajashree Dash, Amiya Kumar Rath et Milu Acharya. *Feature Selection in Gene Expression Data Using Principal Component Analysis and Rough Set Theory.* In Advances in experimental medicine and biology, volume 696, chapitre Software T, pages 91–100. 2011.

[Mokhtarian 1998] F. Mokhtarian et R. Suomela. *Robust image corner detection through curvature scale space.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pages 1376–1381, 1998.

[Moninder Singh 1995] Gregory M. Provan Moninder Singh. *A Comparison of Induction Algorithms for Selective and non-Selective Bayesian Classifiers.* In The 12th International Conference on Machine Learning, pages 497–505, 1995.

[Moore 1910] E. H. Moore. Introduction to a form of general analysis. 1910.

[Moore 1994] Andrew W. Moore et Mary S. Lee. *Efficient algorithms for minimizing cross validation error.* In W.W. Cohen et H. Hirsh, editeurs, The 11th International Conference on Machine Learning, pages 190–198, New Brunswick, NJ, 1994. Morgan Kaufmann, Los Altos, CA.

[Muja 2009] Marius Muja et David G. Lowe. *Fast approximate nearest neighbors with automatic algorithm configuration.* In International Conference on Computer Vision Theory and Applications (VISAPP), pages 331–340, 2009.

[Muja 2013] Marius Muja et David Lowe. *FLANN -Fast Library for Approximate Nearest Neighbors User Manual.* Rapport technique, Computer Science Department, University of British Columbia, Vancouver, BC, Canada, 2013.

[Mundra 2010] Piyushkumar A Mundra et Jagath C Rajapakse. *SVM-RFE with MRMR filter for gene selection.* IEEE transactions on nanobioscience, vol. 9, no. 1, pages 31–37, mar 2010.

[Narendra 1977] P. M. Narendra et K. Fukunaga. *A Branch and Bound Algorithm for Feature Subset Selection.* IEEE Transactions on Computers, vol. C-26, no. 9, pages 917–922, sep 1977.

[Nguifo 1998] Engelbert Mephu Nguifo et Patrick Njiwoua. *Using Lattice-Based Framework as a Tool for Feature Extraction.* In Huan Liu et Hiroshi Motoda, editeurs,

Feature Extraction, Construction and Selection, chapitre Using Latt, pages 205–218. Springer US, Boston, MA, 1998.

[Nourine 1999] Lhouari Nourine et Olivier Raynaud. *A fast algorithm for building lattices.* Information Processing Letters, vol. 71, no. 5-6, pages 199–204, sep 1999.

[Novovičová 2004] Jana Novovičová, Antonín Malík et Pavel Pudil. *Feature Selection using Improved Mutual Information for Text Classification.* In Lecture Notes in Computer Science, pages 1010–1017, 2004.

[Oh 2004] Il-Seok Oh, Jin-Seon Lee et Byung-Ro Moon. *Hybrid genetic algorithms for feature selection.* IEEE Transactions on Pattern Analysis and Machine Inteligence, vol. 26, no. 11, pages 1424–37, nov 2004.

[Oja 2014] E. Oja, A. Hyvärinen et P. Hoyer. *Image Feature Extraction and Denoising by Sparse Coding.* Pattern Analysis & Applications, vol. 2, no. 2, pages 104–110, mar 2014.

[Olshausen 1996] Bruno A. Olshausen et David J. Field. *Emergence of simple-cell receptive field properties by learning a sparse code for natural images.* Nature, vol. 381, no. 13, pages 607–609, 1996.

[Ore 1944] Oystein Ore. *Galois connexions.* Transactions of the American Mathematic Society1, vol. 55, pages 493–513, 1944.

[Pearl 1982] Judea Pearl et Jin H. Kim. *Studies in Semi-Admissible Heuristics.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-4, no. 4, pages 392–399, jul 1982.

[Pearson 1901] Karl Pearson. *On lines and planes of closest fit to systems of points in space.* Philosophical Magazine, vol. 2, pages 559–572, 1901.

[Peng 2005] Hanchuan Peng, Fuhui Long et Chris Ding. *Feature selection based on mutual information : criteria of max-dependency, max-relevance, and min-redundancy.* IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 8, pages 1226–1238, aug 2005.

[Perronnin 2010] Florent Perronnin, Jorge Sánchez et Thomas Mensink. *Improving the Fisher Kernel for Large-Scale Image Classification.* In European Conference on Computer vision (ECCV), pages 143–156. Springer Berlin Heidelberg, 2010.

[Philbin 2008] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic et Andrew Zisserman. *Lost in quantization : Improving particular object retrieval in large scale image databases.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, jun 2008.

[Pohl 1971] Ira Pohl. *Bi-directional search.* Machine Intelligence, vol. 6, pages 127–140, 1971.

[Polaillon 1998] Géraldine Polaillon. *Organisation et interprétation par les treillis de Galois de données de type multivalué, intervalle ou histogramme.* PhD thesis, Paris 9, 1998.

[Prewitt 1970] J. M. S. Prewitt. *Object enhancement and extraction.* Picture Processing and Psychopictorics, pages 75–149, 1970.

[Prum 2013] Sophea Prum. *On the use of a discriminant approach for handwritten word recognition based on bi-character models.* PhD thesis, University of La Rochelle, 2013.

[Pudil 1994] P. Pudil, J. Novovicova et J. Kittler. *Floating search methods in feature selection.* Pattern Recognition Letters, vol. 15, no. 11, pages 1119–1125, 1994.

[Punch 1993] W. F. Punch, E. D. Goodman, Min Pei, Lai Chia-Shun, P. Howland et R. Enbody. *Further research on feature selection and classification using genetic algorithms.* In The 5th International Conference Genetic Algorithms (ICGA), pages 557–564, Champaign Ill, 1993.

[Quelhas 2007] Pedro Quelhas et Jean-Marc Odobez. *Multi-level local descriptor quantization for Bag-of-Visterms image representation.* In The 6th ACM international conference on Image and video retrieval (CIVR), pages 242–249, Amsterdam, The Netherlands, 2007. ACM Press.

[Quinlan 1986] J. R. Quinlan. *Induction of Decision Trees.* Machine Learning, vol. 1, no. 1, pages 81–106, 1986.

[Quinlan 1993] John Ross Quinlan. C4.5 : Programs for Machine Learning. 1993.

[Rakotomalala 2002] Ricco Rakotomalala et Stéphane Lallich. *Construction d'arbres de decision par optimisation.* Revue Extraction des Connaissances et Apprentissage, vol. 16, no. 6, pages 685–703, 2002.

[Rasiwasia 2008] Nikhil Rasiwasia et Nuno Vasconcelos. *Scene classification with low-dimensional semantic spaces and weak supervision.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–6, Anchorage, AK, 2008. IEEE.

[Reddy 1977] D. Raj Reddy. *Speech understanding systems : summary of results of the five-year research effort at Carnegie-Mellon University.* Rapport technique, 1977.

[Rich Caruana 1994] Dayne Freitag Rich Caruana. *Greedy Attribute Selection.* In The Eleventh International Conference on Machine Learning, pages 28–36, 1994.

[Rich 2014] E. Rich et K. Knight. Readings in Artificial Intelligence and Software Engineering. 2014.

[Rigaud 2015] Christophe Rigaud, Clément Guérin, Dimosthenis Karatzas, Jean-Christophe Burie et Jean-Marc Ogier. *Knowledge-driven understanding of images in comic books.* International Journal on Document Analysis and Recognition (IJDAR), vol. 18, no. 3, pages 199–221, sep 2015.

[Rodriguez 2011] Mikel Rodriguez, Ivan Laptev, Josef Sivic, Jean-Yves Audibert et Ecole Normale Supérieure. *Density-aware person detection and tracking in crowds.* In IEEE International Conference on Computer Vision, Barcelone, Espagne, 2011. IEEE.

[Romero 2008] Enrique Romero et Josep María Sopena. *Performing feature selection with multilayer perceptrons.* IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, vol. 19, no. 3, pages 431–41, mar 2008.

[Rosten 2006] Edward Rosten et Tom Drummond. *Machine Learning for High-Speed Corner Detection.* In Aleš Leonardis, Horst Bischof et Axel Pinz, editeurs, European Conference on Computer Vision, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Rowley 1998] Henry A Rowley et Tomaso Poggio. *Neural network-based face detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pages 23–38, 1998.

[Ruck 1990] Dennis W. Ruck, Steven K. Rogers et Matthew Kabrisky. *Feature selection using a multilayer perceptron.* Journal of Neural Network Computing, vol. 2, no. 2, pages 40–48, 1990.

[Rumelhart 1986] D. E. Rumelhart, G. E. Hinton et R. J. Williams. *Learning internal representations by error propagation.* In Parallel distributed processing : explorations in the microstructure of cognition, chapitre Learning i, pages 318–362. MIT Press Cambridge, MA, USA, jan 1986.

[Salas 2011] Joaquín Salas et Carlo Tomasi. *People detection using cdolor and depth images.* In Martinez-Trinidad J.-F., editeur, Mexican Conference on Pattern Recognition, pages 127–135. Springer-Verlag Berlin Heidelberg 2011, 2011.

[Salton 1975] G. Salton, A. Wong et C. S. Yang. *A vector space model for automatic indexing.* Communications of the ACM, vol. 18, no. 11, pages 613–620, nov 1975.

[Schölkopf 1997] Bernhard Schölkopf, Smola Alexander et Müller Klaus-Robert. Kernel principal component analysis, volume 1327 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin/Heidelberg, 1997.

[Schölkopf 2002] Bernhard Schölkopf et Alexander J. Smola. Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2002.

[Schwarz 1978] Gideon Schwarz. *Estimating the Dimension of a Model.* The Annals of Statistics, vol. 6, no. 2, pages 461–464, mar 1978.

[Setiono 1997] R Setiono et H Liu. *Neural-network feature selector.* IEEE Transactions on Neural Networks, vol. 8, no. 3, pages 654–62, jan 1997.

[Siedlecki 1988] Wojciech Siedlecki et Jack Sklansky. *On automatic feature selection.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 2, no. 2, pages 197–220, jun 1988.

[Siedlecki 1989] W. Siedlecki et J. Sklansky. *A note on genetic algorithms for large-scale feature selection.* Pattern Recognition Letters, vol. 10, no. 5, pages 335–347, nov 1989.

[Skalak 1994] David B. Skalak. *Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms.* In Machine Learning : Proceedings of the Eleventh International Conference, pages 293–300, 1994.

[Stearns 1976] SD Stearns. *On selecting features for pattern classifiers.* In International Conference on Pattern Recognition, pages 71 – 75, 1976.

[Stigler 2007] Stephen M. Stigler. *The Epic Story of Maximum Likelihood.* Statistical Science, vol. 22, no. 4, pages 598–620, nov 2007.

[Stricker 1995] Markus A. Stricker et Markus Orengo. *Similarity of color images.* In Wayne Niblack et Ramesh C. Jain, editeurs, SPIE 2420, Storage and Retrieval for Image and Video Databases III, pages 381–392, San Jose, mar 1995. International Society for Optics and Photonics.

[Sugiyama 2006] Masashi Sugiyama. *Local Fisher discriminant analysis for supervised dimensionality reduction.* In The 23rd international conference on Machine learning (ICML'06), pages 905–912, New York, USA, jun 2006. ACM Press.

[Sutter 1995] Jon M. Sutter, Steve L. Dixon et Peter C. Jurs. *Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing.* Journal of Chemical Information and Modeling, vol. 35, no. 1, pages 77–84, jan 1995.

[Swain 1991] Michael J. Swain et Dana H. Ballard. *Color indexing.* International Journal of Computer Vision, vol. 7, no. 1, pages 11–32, nov 1991.

[Tan 2014] Ching Siang Tan, Wai Soon Ting, Mohd Saberi Mohamad, Weng Howe Chan, Safaai Deris et Zuraini Ali Shah. *A Review of Feature Extraction Software for Microarray Gene Expression Data.* BioMed Research International, vol. 2014, pages 1–15, 2014.

[Vafaie 1993] Haleh Vafaie et Kenneth De Jong. *Robust Feature Selection Algorithms.* In The 5th International Conference on Tools with Artificial Intelligence, pages 356–363, Rockville, 1993. IEEE Computer Society Press.

[van de Sande 2010] Koen E. A. van de Sande, Theo Gevers et Cees G. M. Snoek. *Evaluating color descriptors for object and scene recognition.* IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pages 1582–96, sep 2010.

[Van Gemert 2008] Jan C. Van Gemert, Jan-Mark Geusebroek, Cor J. Veenman et Arnold W. M. Smeulders. *Kernel Codebooks for Scene Categorization.* In The 10th European Conference on Computer Vision, pages 696–709, Marseille, France, 2008. Springer Berlin Heidelberg.

[Van Gemert 2010] Jan C Van Gemert, Cor J Veenman, Arnold W M Smeulders et Jan-Mark Geusebroek. *Visual word ambiguity*. IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 7, pages 1271–83, jul 2010.

[van Rijsbergen 1981] C.J. van Rijsbergen, D.J. Harper et M.F. Porter. *The selection of good search terms*. Information Processing & Management, vol. 17, no. 2, pages 77–91, 1981.

[Vergara 2014] Jorge R Vergara et Pablo A Estévez. *A Review of Feature Selection Methods Based on Mutual Information*. Neural Computing & Applications, vol. 24, no. 1, pages 175–186, 2014.

[Viola 2004] Paul Viola et Michael J. Jones. *Robust real-time face detection*. International Journal of Computer Vision, vol. 57, no. 2, pages 137–154, may 2004.

[Vychodil 2008] Vilem Vychodil. *A New Algorithm for Computing Formal Concepts*. In Trappl R., editeur, The 19th Cybernetics and Systems (EMCSR), pages 15–21, 2008.

[Wang 2010] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang et Yihong Gong. *Locality-constrained Linear Coding for image classification*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3360–3367. IEEE, jun 2010.

[Weldon 1996] Thomas P Weldon, William E Higgins et Dennis F Dunn. *Efficient Gabor filter design for texture segmentation*. Pattern Recognition, vol. 29, no. 12, pages 2005–2015, 1996.

[West 2003] Mike West. *Bayesian Factor Regression Models in the " Large p , Small n " Paradigm*. Bayesian Statistics, vol. 7, pages 723–732, 2003.

[Whitney 1971] A.W. Whitney. *A Direct Method of Nonparametric Measurement Selection*. IEEE Transactions on Computers, vol. C-20, no. 9, pages 1100–1103, sep 1971.

[Widrow 1960] B. Widrow et M. E. Hoff. *Adaptive switching circuits*. Rapport technique, 1960.

[Wille 1982] Rudolf Wille. *Restructuring Lattice Theory : An Approach Based on Hierarchies of Concepts*. In Ivan Rival, editeur, Ordered Sets, pages 445–470. Springer Netherlands, Dordrecht, 1982.

[Wolf 2005] Lior Wolf et Amnon Shashua. *Feature Selection for Unsupervised and Supervised Inference : The Emergence of Sparsity in a Weight-Based Approach*. The Journal of Machine Learning Research, vol. 6, pages 1855–1887, dec 2005.

[Yahia 2001] Sadok Ben Yahia et Ali Jaoua. *Discovering Knowledge from Fuzzy Concept Lattice*. In Abraham Kandel, Mark Last et Horst Bunke, editeurs, Data Mining and Computational Intelligence, volume 68, pages 167–190. Physica-Verlag HD, 2001.

[Yang 1998] J. Yang et V. Honavar. *Feature subset selection using a genetic algorithm.* IEEE Intelligent Systems, vol. 13, no. 2, pages 44–49, mar 1998.

[Yu 2003] Lei Yu et Huan Liu. *Feature selection for high-dimensional data : A fast correlation-based filter solution.* In The 12th International Conference on Machine Learning (ICML-2003), pages 856–863, Washington DC, 2003.

[Zahn 1972] Charles Zahn et Ralph Roskies. *Fourier descriptors for plane closed curves.* IEEE Transactions on computers, vol. 21, no. 3, pages 269–281, 1972.

[Zhang 2011] Yimeng Zhang, Zhaoyin Jia et Tsuhan Chen. *Image retrieval with geometry-preserving visual phrases.* In Conference on Computer Vision and Pattern Recognition, pages 809–816. IEEE, jun 2011.

[Zhou 2001] Feng Zhou, Ju Fu Feng et Qing Yun Shi. *Texture feature based on local Fourier transform.* In International Conference on Image Processing, volume 2, pages 610–613. IEEE, 2001.

[Zhou 2010] Xi Zhou, Kai Yu, Tong Zhang et Thomas S Huang. *Image Classification using Super-Vector Coding of Local Image Descriptors.* In Kostas Daniilidis, Petros Maragos et Nikos Paragios, editeurs, European Conference on Computer Vision (ECCV), pages 141–154. Springer Berlin Heidelberg, 2010.

Réduction de dimension sur le modèle de sac de mots visuels à l'aide de Analyse Formelle de Concept.

Résumé : La réduction des informations redondantes et/ou non-pertinentes dans la description de données est une étape importante dans plusieurs domaines scientifiques comme les statistiques, la vision par ordinateur, la fouille de données ou l'apprentissage automatique. Dans ce manuscrit, nous abordons la réduction de la taille des signatures des images par une méthode issue de l'Analyse Formelle de Concepts (AFC), qui repose sur la structure du treillis des concepts et la théorie des treillis. Les modèles de sac de mots visuels consistent à décrire une image sous forme d'un ensemble de mots visuels obtenus par clustering. La réduction de la taille des signatures des images consiste donc à sélectionner certains de ces mots visuels. Dans cette thèse, nous proposons deux algorithmes de sélection d'attributs (mots visuels) qui sont utilisables pour l'apprentissage supervisé ou non. Le premier algorithme, RedAttSansPerte, ne retient que les attributs qui correspondent aux irréductibles du treillis. En effet, le théorème fondamental de la théorie des treillis garantit que la structure du treillis des concepts est maintenue en ne conservant que les irréductibles. Notre algorithme utilise un graphe d'attributs, le graphe de précédence, où deux attributs sont en relation lorsque les ensembles d'objets à qui ils appartiennent sont inclus l'un dans l'autre. Nous montrons par des expérimentations que la réduction par l'algorithme RedAttsSansPerte permet de diminuer le nombre d'attributs tout en conservant de bonnes performances de classification. Le deuxième algorithme, RedAttsFloue, est une extension de l'algorithme RedAttsSansPerte. Il repose sur une version approximative du graphe de précédence. Il s'agit de supprimer les attributs selon le même principe que l'algorithme précédent, mais en utilisant ce graphe flou. Un seuil de flexibilité élevé du graphe flou entraîne mécaniquement une perte d'information et de ce fait une baisse de performance de la classification. Nous montrons par des expérimentations que la réduction par l'algorithme RedAttsFloue permet de diminuer davantage l'ensemble des attributs sans diminuer de manière significative les performances de classification.

Mots clés : réduction de dimension, sélection d'attributs, treillis, irréductible, analyse formelle de concepts, modèle de sac de mots visuels, graphe de précédence, graphe de précédence flou, méthode algébrique, logique floue.

Dimension reduction on bag of visual words with Formal Concept Analysis.

Abstract:
In several scientific fields such as statistics, computer vision and machine learning, redundant and/or irrelevant information reduction in the data description (dimension reduction) is an important step. This process contains two different categories : feature extraction and feature selection, of which feature selection in unsupervised learning is hitherto an open question. In this manuscript, we discussed about feature selection on image datasets using the Formal Concept Analysis (FCA), with focus on lattice structure and lattice theory. The images in a dataset were described as a set of visual words by the bag of visual words model. Two algorithms were proposed in this thesis to select relevant features and they can be used in both unsupervised learning and supervised learning. The first algorithm was the RedAttSansPerte, which based on lattice structure and lattice theory, to ensure its ability to remove redundant features using the precedence graph. The formal definition of precedence graph was given in this thesis. We also demonstrated their properties and the relationship between this graph and the AC-poset. Results from experiments indicated that the RedAttsSansPerte algorithm reduced the size of feature set while maintaining their performance against the evaluation by classification.
Secondly, the RedAttsFloue algorithm, an extension of the RedAttsSansPerte algorithm, was also proposed. This extension used the fuzzy precedence graph. The formal definition and the properties of this graph were demonstrated in this manuscript. The RedAttsFloue algorithm removed redundant and irrelevant features while retaining relevant information according to the flexibility threshold of the fuzzy precedence graph. The quality of relevant information was evaluated by the classification. The RedAttsFloue algorithm is suggested to be more robust than the RedAttsSansPerte algorithm in terms of reduction.

Keywords: dimension reduction, feature selection, lattice, irreducible, formal concept analysis, bag of visual words model, precedence graph, fuzzy precedence graph, algebraic method, fuzzy logic.

Laboratoire Informatique, Image, Interaction
Faculté des Sciences & Technologies
Avenue Michel Crépeau

17042 LA ROCHELLE CEDEX 1