HAL
open science

Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions

Mounzer Boubou

▶ **To cite this version:**

Mounzer Boubou. Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions. Mathématiques [math]. Université Claude Bernard - Lyon I, 2007. Français. NNT : . tel-00195779v2

HAL Id: tel-00195779

https://theses.hal.science/tel-00195779v2

Submitted on 11 Dec 2007

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.

N° d'ordre : 287 Année 2007

THÈSE

Présentée devant

Université Claude Bernard - Lyon I

pour obtenir

du DIPLOME DE DOCTEUR

(arrêté du 7 août 2006)

Spécialité :

Statistiques - Informatique

par

Mounzer BOUBOU

Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions

Directeur de thèse :

Michel LAMURE, professeur à l'université Lyon I

Thèse soutenue le 29 Novembre 2007 devant la Commission d'examen

Jury

Jean Paul AURAY	Président
Ivan LAVALLEE	Rapporteur
Djamel ZIGHED	Rapporteur
Christine VERDIER	Examinateur
Michel LAMURE	Directeur de thèse
Ahmed BOUNEKKAR	Co-Directeur de thèse

REMERCIEMENTS

Je tiens à exprimer ma gratitude à Monsieur Michel LAMURE, pour avoir accepté de diriger cette thèse tout en ayant l'obligeance de consacrer de son temps précieux pour donner des conseils très utiles qui m'ont permis d'enrichir et de mener à bien ce travail.

Mes remerciements vont également à mon co-directeur de thèse, Monsieur Ahmed BOUNEKKAR, Maître de Conférences à l'Université Claude Bernard, Lyon I, pour son aide et sa présence constante à chaque fois que je faisais appel à lui. Ses nombreuses remarques et suggestions ont contribué à améliorer la qualité de ce mémoire. Qu'il trouve ici l'expression de ma reconnaissance.

Je remercie mes deux rapporteurs : Monsieur Ivan LAVALLEE, Professeur à l'Université Paris 8 et Monsieur Abdelkader Djamel Zighed, Professeur à l'Université Lumière Lyon 2, pour avoir accepté de juger mon travail, de même que pour leur participation au jury.

Mes remerciements vont également à Madame Christine VERDIER, Professeur au **LIG** (**L**aboratoire d'**I**nformatique de **G**renoble) pour l'honneur qu'elle m'a fait en acceptant de participer à mon jury de thèse.

Une reconnaissance particulière pour Monsieur Jean-Paul AURAY, Directeur de recherche au CNRS, pour la générosité qu'il m'a témoigné et l'intérêt constant qu'il a porté à mon travail. Ses recommandations et ses conseils ont été significatifs et son aide incontestable. Je lui témoigne mon profond respect.

Je remercie chaleureusement tous les membres de l'équipe **MA^{2}D**.

Une pensée particulière pour Samo qui m'a supporté tout au long de ce travail ainsi que pour mes enfants Yazan et Layal.

Je remercie également mes parents, mes frères et soeurs qui n'ont pas cessé de nourrir ma motivation pour ce travail et qui m'ont assisté de leurs encouragements.

Finalement, que toutes celles et tous ceux qui m'ont aidé d'une manière ou une autre à la réalisation de ce travail soient ici remerciés.

Table des matières

I Analyse et Classification des Données **3**

1 Introduction **5**

2 Analyse des Données **9**

2.1 Introduction 10

2.2 Analyse des données classiques 10

2.2.1 Analyse en composantes principales **(ACP)** : 10

2.2.2 Analyse factorielle des correspondances **(AFC)** 11

2.2.3 Analyse Factorielle Multiple **(AFM)** 12

2.3 Analyse des données symboliques 13

2.3.1 Introduction 13

2.3.2 Présentation formelle de l'analyse de données symboliques 14

2.3.3 Les apports de l'analyse de données symboliques 16

2.3.4 Les principaux avantages des objets symboliques : 16

3 La classification **19**

3.1 Introduction 20

3.2 Définitions 22

3.3 Notion de ressemblance entre objets 25

3.3.1 Les mesures de proximité sur un ensemble 27

3.3.2 Différents types d'indices de dissimilarité 27

3.4 Présentation des méthodes de classification de données 31

3.5 Méthodes hiérarchiques 34

3.5.1 Classification Ascendante Hiérarchique (CAH) 35

3.5.2 Classification Hiérarchique Descendante(CHD) 40

3.6 Classification par partition **(CPP)** 44

3.6.1 Méthodes k-means 46

3.6.2 Méthodes k-medoids 49

3.6.3 TAXY 53
3.7 Classification par la densité 57
3.7.1 Approche basée sur la connexité de densité 58
3.7.2 Approche basée sur la fonction de densité 60
3.8 Classification basée sur la quantification par grille 61
3.9 Autres Méthodes 64
3.10 Comparaison de classification 66
3.11 Conclusion 69

II Quelques Approches de Classification des Données 73

4 Approche basée sur l'agrégation d'opinions 75
4.1 Introduction 76
4.2 Présentation du problème 77
4.2.1 Etude du critère d'optimisation 78
4.2.2 Calcule du nombre de désaccords 80
4.3 Formulation du problème 81
4.4 Résolution du problème 82
4.4.1 Application 84
4.5 Extension de la méthode 87
4.6 Cohérence et ressemblance entre groupes 89
4.7 Structuration de l'ensemble des variables 92
4.7.1 Formulation classique du problème 92
4.7.2 Une méthode heuristique pour la résolution du problème 94

5 Approche prétopologique 97
5.1 Introduction 98
5.2 Eléments de prétopologie 99
5.2.1 Définitions et propriétés de base : 99
5.2.2 Concepts de préfiltre, filtre, filtre stable 101
5.2.3 Concepts de prévoisinages et caractérisations 102
5.2.4 Comparaison de prétopologies 104
5.2.5 Encadrement d'une structure s par deux structures de type V . . . 105
5.2.6 Structure de type VDS et relation binaire 107
5.2.7 Relation réflexive et relations d'équivalences associées 107
5.2.8 Encadrement de s par deux structures de type VDS (cas E fini) . . 110

5.2.9 Concept de continuité 112
5.2.10 Prétopologie induite (second type de problème) 114
5.3 Application à un problème de classification 115
5.3.1 (Pré)topologie et partition 116
5.3.2 Familles de topologies-classes associées à s et éléments remarquables 117
5.3.3 Processus de construction de $\mathcal{T}_C^+(s)$ et de $\mathcal{T}_C^-(s)$ 117
5.3.4 Produit d'espaces prétopologique 119
5.3.5 Application à la classification 120
5.4 Algorithme de la méthode 127
5.4.1 Conclusion 129

6 Approche de la classification par l'agrégation des préférences 131
6.1 Introduction : 132
6.2 Position du problème 132
6.3 Etude de la structure des familles de variables décisives 136
6.4 Etude spécifique de la famille $\mathcal{D}_{(2,Ag)}$ 142
6.5 Hypothèse de finesse 146
6.6 Conclusion 151

7 Application 153

8 Conclusion et perspectives 171

Liste des tableaux

2.1 Tableau des données Symboliques 15

3.1 Bilan des définitions des indices de dissimilarité 28
3.2 Tableau de caractères pour les données dichtomiques 30
3.3 Tableau des indices en cas des données binaires 30
3.4 Sommaire des algorithmes de classification 70

4.1 Tableau des données cétacés 86

7.1 Tableau des données cétacés 154

Table des figures

3.1 Exemple d'une hiérarchie de parties de $\mathcal{I}$ 24
3.2 Techniques de classification 32

5.1 Adhérence et Intérieur 99
5.2 Présentation synthétique de l'ensemble des résultats 112
5.3 Exemple 1 de bases des voisinages 122
5.4 Exemple 2 de bases des voisinages 123
5.5 Graphe des relations entre les individus - cas q=3 126
5.6 Partition donnée dans le cas q=3 127
5.7 Graphe des relations entre les individus - cas q=2 127
5.8 Les Cliques Maximales trouvés dans ce graphe 128

7.1 Les Cliques Maximales trouvés dans le graphe avec q=14 163
7.2 Les Cliques Maximales trouvés dans le graphe avec q=13 165
7.3 Les Cliques Maximales trouvés dans le graphe avec q=12 166
7.4 Les Cliques Maximales trouvés dans le graphe avec q=11 167
7.5 Les Cliques Maximales trouvés dans le graphe avec q=10 168

Glossaire

$D_{(\alpha,(xy),P,Ag)}$ **:** Ensemble des variables décisives en α, sur le couple (x, y), pour la fonction de classification et pour l'agrégation Ag, 144

$\mathcal{E}$ **:** Relation d'équivalence, 116

$\mathcal{P}_V^+(s, E)$ **:** Ensemble des structures prétopologiques sur E, de type V plus fines que s, 113

$\mathcal{P}_V^-(s, E)$ **:** Ensemble des structures prétopologiques sur E, de type V moins fines que s, 113

$\mathcal{R}$ **:** Relation binaire, 115

$\mathcal{V}(x)$ **:** Famille des prévoisinages de x, 110

$\succcurlyeq_{pf}$ **:** Relation de finesse, 112

$\mathbf{t}_d$ **:** Topologie Discrète , 113

$\mathbf{t}_g$ **:** Topologie Grossière , 113

$a(.)$: Adhérence, 108

$i(.)$: Intérieur, 108

$s = (i, a)$ **:** Structure Prétopologique (Prétopologie), 108

ACP : Analyse en composantes principales, 12

AD : Analyse des Données, 12

ADS : Analyse des données symboliques, 14

AFC : Analyse factorielle des correspondances, 13

Première partie

Analyse et Classification des Données

Chapitre 1

Introduction

Contexte général du ce Document

"Des chercheurs qui cherchent, on en trouve. Des chercheurs qui trouvent, on en cherche."

Charles De Gaulle

Le cadre général de ce travail est celui de l'*Analyse et la Classification des données*. Nous sommes dans un âge souvent désigné sous le nom de l'âge de l'information. En cet âge de l'information, parce que nous croyons que l'information mène à la puissance et au succès, et grâce aux technologies sophistiquées telles que des ordinateurs, des satellites, etc . . . , nous avions rassemblé des quantités de l'information énormes. Au commencement, par l'arrivée des ordinateurs et des moyens pour la mémoire numérique de masse, ces collectes des données massives enregistrées sur les structures disparates sont très rapidement devenues accablantes. Cette première chose a mené à la création des bases de données et des systèmes de gestion structurés de base de données (Les SGBD). Les systèmes de gestion efficaces de bases de données ont été des capitaux très importants pour la gestion d'un corpus volumineux des données et particulièrement pour la recherche pertinente et efficace d'information particulière dans une grande collection lorsque nécessaires. Aujourd'hui, nous avons bien plus d'informations que nous pouvons manipuler : des transactions et des données scientifiques, aux images satellites, aux états des textes ...
Confronté avec les collectes des données énormes. Il a ainsi fallu développer des outils permettant une appréhension rapide de l'information contenue dans de "grands tableaux de données". C'est l'object de l'*analyse et la classification des données*. Ces outils se sont développés d'abord à partir de méthodes préexistantes à cette période, (analyse en composante principales, analyse des correspondances,...), les méthodes descriptives fournissant

des résumés numériques et graphiques permettant de synthétiser l'information la plus importante contenue dans un tableau de données.

Plan de la thèse

Ce document est organisé sur deux parties :
- La première : Analyse et Classification des Données
- La deuxième : Quelques contributions de la Classification des données.

Dans le deuxième chapitre, nous présentons brièvement quelques notions et techniques de l'analyse des données (**ACP, AFC, AFM**). Ensuite, nous présentons l'analyse des données symboliques (**ADS**) afin de montrer l'intérêt de ce type d'analyse et ces propres propriétés. Nous donnons la définition des objets symboliques qui jouent un rôle essentiel en (**ADS**), nous présentons quelques types d'objets symboliques.

Dans le troisième chapitre nous abordons la classification des données, nous commençons par quelques définitions de certains outils mathématiques : (partition, hiérarchie,...), et comme la plupart de techniques de classification des données sont basées sur des mesures de proximité, nous présentons quelques indices de dissimilarité classés selon le type de variable. Les différentes techniques de classification seront présentées en les divisant en plusieurs familles (hiérarchique, par partition, par la densité, par la grille,...). Pour chaque technique, nous citons les algorithmes les plus utilisés et nous citons pour chaque technique les points forts et les points faibles.

Dans la deuxième partie, nous proposons quelques approches de la classification des données. Ces approches sont basées sur des concepts mathématiques tel que la prétopologie, la théorie de graphe et l'agrégation des préférences...

Objectifs de cette thèse

L'objectif principal de ce travail est de trouver de nouvelles méthodes de classification où les données ne sont pas forcément métriques. Le but de la classification est de diviser un ensemble des données en plusieurs classes homogènes. Cette tâche n'est pas facile en cas des données complexes où nous ne pouvons pas appliquer une mesure de proximité entre les individus sans passer par d'autres procédures. Notre objectif dans cette thèse est de trouver une méthode de classification des données sans passer par le calcul d'une mesure de proximité utilisée dans le cas des données numériques et en se basant sur d'autres

outils mathématiques comme la prétopologie ou l'agrégation de préférences.

Dans cette thèse, nous développons trois approches de la classification.

Dans la première approche, nous présentons une méthode de classification basée sur l'agrégation d'opinions. La méthode proposée consiste à associer à chaque variable une fonction de classement qui va jouer le rôle de juge. Ce dernier va classer les individus selon ses propres critères. A partir de l'ensemble de classement de toutes les variables, on cherche à construire sur l'ensemble des individus un classement collectif qui soit la meilleure agrégation possible. Afin de résoudre le problème d'optimisation rencontré, nous utilisons la méthode de recuit simulé.

La Deuxième approche s'appuie sur la construction d'espaces prétopologiques pour la manipulation des relations entre les individus. Ces espaces utilisent la notion de base de voisinages afin de réaliser un outil mathématique pour mesurer la proximité entre les individus.
De nombreux contextes issus de problématiques propres aux sciences sociales nécessitent que l'on soit en mesure d'établir des liens ou d'apprécier des proximités ou des ressemblances entre des individus. L'outil mathématique connu de formalisation du concept de proximité est la topologie, dont souvent on ne connaît que le cas particulier qu'est la métrique. Mais, en réalité, on s'aperçoit assez rapidement que l'axiomatique de la topologie est trop contraignante, parce que trop riche, et n'est pas en mesure d'apporter des solutions intéressantes aux problèmes posés. D'où l'idée de chercher à affaiblir l'axiomatique de la topologie, pour espérer une meilleure adaptation aux problèmes rencontrés : telle est l'idée qui conduit à ce qu'on appellera ici la "prétopologie".

Dans le domaine de la classification, la plupart des méthodes ont plus ou moins implicitement recours à un concept de mesure de "proximité". Dans le cas où le contexte est tel qu'une telle "proximité" n'a rien de naturel les outils habituels ne sont guère satisfaisants. Supposons par exemple, qu'on dispose d'une population Ω sur laquelle on a observé divers caractères et que, pour chacun de ces caractères, l'observation se traduise par une variable X définie sur Ω et prenant ses valeurs dans l'ensemble E des modalités du caractère. Si le caractère est quantitatif alors E peut naturellement être muni d'une mesure de proximité qui pourra être utilisée pour apprécier la ressemblance ou non de deux objets de Ω. Mais si le caractère est purement qualitatif alors ceci n'est plus possible, sauf à prendre le risque de malmener sérieusement la réalité. Il faut alors construire une procédure spécifique au

contexte qualitatif, en vue de proposer une formalisation adaptée du concept de "proximité". La prétopologie peut être utile à la réalisation de cet objectif.

Dans la troisième approche, nous proposons d'analyser le problème de la classification d'une population à partir de la mesure de k variables quantitatives X_i sur ses individus à la manière des modèles d'agrégation des préférences. L'objectif n'est donc pas de développer une méthode originale de classification mais plutôt d'interpréter le problème de fond de la classification par les concepts de l'agrégation des préférences et de produire des résultats qui permettent de qualifier la qualité d'une procédure de classification et, peut être d'introduire dans cette procédure une part d'intervention humaine permettant d'en contrôler le déroulement.

L'idée qui sous-tend cette approche s'appuie sur la constatation qu'il est en général aisé de classifier une population à partir de la donnée des observations d'une seule variable. En effet, les diverses méthodes habituellement utilisées de classification fournissent, la plupart du temps, des résultats quasiment identiques lorsqu'on les applique dans des situations où n'intervient qu'une seule variable. Il est alors naturel de se dire que lorsque k variables $X_1, X_2, \ldots, X_k$ sont en jeu, on obtient k classifications associées à chacune de ces variables et qu'obtenir une classification finale revient à agréger ces k classifications en une seule. On se ramène ainsi à un problème d'agrégation. Les travaux dédiés aux problèmes d'aide à la décision apportent de nombreux résultats à ce sujet, notamment les travaux situés dans la lignée de ceux menés par **K. J. Arrow** [Arr63]. Nous nous situerons dans cette lignée tout en notant que le principal résultat est qu'il n'existe pas de méthode d'agrégation, mais des méthodes d'agrégation, partant du même constat qu'il n'existe pas une méthode de classification mais des méthodes de classification, nous souhaitons produire une réflexion s'appuyant sur le parallélisme qui existe entre les deux problèmes : classification et agrégation pour obtenir des résultats qui peuvent éclairer, sur le fond, l'utilisateur des méthodes de classification dans sa démarche et l'aider à se poser des questions.

Le dernier chapitre de cette thèse est une application de la méthode que nous avons proposé dans l'approche prétopologique. Les données de cette application ont déjà été utilisé par *Grötschel* [GW89]. L'intérêt d'utiliser cet exemple est de comparer notre résultat avec celui de *Grötschel* qu'il qualifie de solution "optimale".

Chapitre 2

Analyse des Données

Sommaire

2.1 Introduction **10**

2.2 Analyse des données classiques **10**

2.2.1 Analyse en composantes principales **(ACP)** : 10

2.2.2 Analyse factorielle des correspondances **(AFC)** 11

2.2.3 Analyse Factorielle Multiple **(AFM)** 12

2.3 Analyse des données symboliques **13**

2.3.1 Introduction 13

2.3.2 Présentation formelle de l'analyse de données symboliques . . . 14

2.3.3 Les apports de l'analyse de données symboliques 16

2.3.4 Les principaux avantages des objets symboliques : 16

2.1 Introduction

L'expansion des techniques et des matériels informatiques dans de nombreux domaines (Économie, Médecine, Biologie...) a permis le stockage et la gestion de grandes masses de données. En conséquence, il est difficile, lorsque le volume de données est important, de tirer des enseignements utiles sans traitement au préalable.
Les méthodes d'analyse des Données (AD) se sont alors développées. Elles représentent un ensemble de méthodes pour la synthèse et la recherche de relations dans de grands ensembles de données. Elles permettant aujourd'hui d'organiser, de synthétiser et d'extraire des connaissances à partir de ces données généralement à travers une représentation graphique particulière des données de volume important. L'exploitation d'un graphique ainsi constitué se résume, en somme, en une compression de données où la lecture de l'information permet l'aide à la décision.
Les premiers travaux sur l'analyse factorielle et de la classification des données remontent aux années 60 [SS63], [Ben73],...

Aujourd'hui, on peut recenser énormément de travaux dans ce domaines, favorisés par la quantité gigantesque de données disponibles.

2.2 Analyse des données classiques

2.2.1 Analyse en composantes principales (ACP) :

L'analyse en composantes principales est la suite des travaux de **Pearson** [Pea01] qui cherchait à approcher un système de points dans l'espace, selon sa terminologie, par un sous-espace linéaire de dimension inférieure.
Plus précisément, **Pearson** étudiait le problème d'approcher des données multivariées par une droite telle qu'elle minimise la somme des écarts des points à la droite au carré.
Nous allons dans ce paragraphe présenter brièvement le fondement théorique de l'ACP. Le lecteur désirant de plus amples détails pourra consulter [Sap90] chap. 8.

Méthodologie de l'ACP :

Le tableau des données $(n \times p)$ forme un nuage de n points dans un espace à p dimensions, ou un nuage de p points dans un espace à n dimensions. Un diagramme binaire consiste à projeter ces points sur deux des dimensions choisies plus ou moins arbitrairement.

L'**ACP** consiste à projeter les points sur une droite, un plan... un sous-espace à d dimensions (avec $d \leq p$) choisi de façon à optimiser un certain critère. Intuitivement, on cherchera le sous-espace donnant la meilleure visualisation possible de notre nuage de points. Un bon choix consiste à rechercher la plus grande dispersion (le plus grand étalement) possible des projections dans le sous-espace choisi. On est amené ainsi à chercher une rotation de notre système d'axes initial (les variables) permettant de mieux voir notre nuage.

Remarque :
L'ACP, étant une méthode de réduction de dimension, il est important de savoir qu'elle ne peut pas retenir la totalité de l'information contenue dans le nuage de points initial. Enfin, l'ACP prend uniquement en compte les dépendances linéaires entre les variables et ne peut donc pas fournir une projection fidèle pour une distribution non-linéaire de points.

2.2.2 Analyse factorielle des correspondances (AFC)

L'analyse des correspondances, présentée sous ce nom et développée par Benzécri (1969), a un certain nombre de précurseurs, parmi lesquels il faut citer Guttman (1941), Hayashi (1956).
L'analyse des correspondances peut être présentée selon divers points de vue. Il est d'ailleurs difficile de faire l'historique précis de cette méthode. Les principes théoriques remontent probablement aux travaux de Fisher (1940) sur les tables de contingences, dans un cadre de statistique inférentielle classique. Depuis les travaux de Benzécri (1973) [Ben73] et de Escofier-Cordier (1965), on utilise surtout les propriétés algébriques et géométriques de l'outil descriptif que constitue l'analyse. Cette méthode n'est pas un cas particulier de l'analyse en composantes principales bien que l'on puisse se ramener à cette technique en faisant des changements de variables appropriés (à condition de traiter chaque espace séparément). On peut la présenter comme un cas particulier de l'analyse canonique lorsque les données ont un codage "disjonctif" et également comme un cas particulier de l'analyse discriminante. On peut enfin montrer qu'il s'agit de la recherche de la meilleure représentation simultanée de deux ensembles constituant les lignes et les colonnes d'un tableau de données positives.

Dans le cas d'une **AFC** les données sont transformées afin de mettre en évidence la répartition relative de l'individu par rapport aux variables et d'établir les corrélations entre les profils obtenus. En fait, l'**AFC** correspond à une ACP réalisée sur un tableau

qui a subi un traitement supplémentaire, en prenant en compte la marginale des variables et des individus. Ce traitement permet de représenter les individus et les variables dans le même espace (les vecteurs propres de la matrice $n \times p$ et sa transposée sont identiques). Elle permet donc d'observer la typologie des individus par rapport aux variables ou la typologie des variables par rapport aux individus [Ben73].

2.2.3 Analyse Factorielle Multiple (AFM)

L'analyse des correspondances peut se généraliser de plusieurs façons au cas où plus de deux ensembles sont mis en correspondance. Une des généralisations la plus simple et la plus utilisée est l'analyse des correspondances multiples qui permet de décrire de vastes tableaux binaires, dont les fichiers d'enquêtes socio-économiques constituent un exemple privilégié : les lignes de ces tableaux sont en général des individus ou observations (il peut en exister plusieurs milliers) ; les colonnes sont des modalités de variables nominales, le plus souvent des modalités de réponses à des questions. Il s'agit en fait d'une simple extension du domaine d'application de l'analyse des correspondances, avec cependant des procédures de calcul et des règles d'interprétation spécifiques. L'analyse des correspondances multiples est une analyse des correspondances simple appliquée non plus à une table de contingence, mais à un tableau disjonctif complet. Les propriétés d'un tel tableau sont intéressantes, les procédures de calculs et les règles d'interprétation des représentations obtenues sont simples et spécifiques. L'extension du domaine d'application de l'analyse des correspondances se fonde sur l'équivalence suivante : si pour n individus, on dispose des valeurs (réponses) prises par deux variables nominales ayant respectivement p1 et p2 modalités, il est alors équivalent de soumettre à l'analyse des correspondances le tableau de contingence (p1 , p2) croisant les deux variables ou d'analyser le tableau binaire à n lignes et (p1 + p2) colonnes décrivant les réponses. L'analyse de ce dernier tableau est plus coûteuse, mais plus intéressante, car elle se généralise immédiatement au cas de plus deux variables nominales.

L'AFM permet de comparer les relations qui existent entre les individus et les variables en fonction d'un critère. Elle est basée sur l'étude de différents tableaux représentant les mêmes individus selon les mêmes variables mais pour une valeur de critère différente à des instants différents. L'AFM consiste alors à réaliser une Analyse Factorielle (AFC ou ACP) soit à partir de la concatenation des différents tableaux, soit à partir d'un des tableaux, les éléments des autres tableaux étant alors considérés comme des éléments supplémentaires (simplement projetés dans l'espace de représentation obtenu).

2.3 Analyse des données symboliques

2.3.1 Introduction

Dans la pratique, les données peuvent être plus complexes et chaque case du tableau peut contenir des valeurs multiples, parfois pondérées, qui peuvent être liées entre elles par des règles ou des taxonomies.
Différentes méthodes de codage ont été créées pour ce genre de données. Ces recodages peuvent entraîner des pertes d'information. La structuration des données complexes implique alors une modélisation loin de la réalité. Le traitement de ces données dans leur format d'origine peut donc s'avérer essentiel [Vra02].

Des données de cette forme sont dites "symboliques" car elles ne sont pas purement numériques [DKBM00]. Concrètement, une case d'un tableau de données symboliques peut contenir différents types de données, par exemple :

- une valeur quantitative ou qualitative
- des valeurs quantitatives ou qualitatives
- un intervalle
- un histogramme ordinal ou nominal
- une loi de probabilité, etc

L'analyse des données classiques est un cas particulier de l'Analyse des Données Symboliques (ADS). L'ADS donne une importance aux individus en donnant un cadre dans lequel ils peuvent être représentés et analysés en tenant compte de leur complexité et de leur variation interne de manière proche de la réalité.

De nombreux travaux ont déjà fait en analyse des données symboliques. On peut citer parmi eux les algorithmes dits de "Conceptual Clustering" [DGLS80], puis [MSD81]. Les applications sont nombreuses , diagnostic des maladies [Man88], pour l'aide au diagnostic des causes de surdité [MD88], climatologiques [Vra02] et beaucoup de travaux en classification des données symboliques comme, classification pyramidale [Bri91], classification automatique [Cha97], classification conceptuelle[Gir97], treillis de Galois [Pol98] et d'autres travaux [DKBM00],[Kod92],...

L'ADS s'appuie donc sur deux notions définies sur le même ensemble de description. La première est définie par un ensemble d'entités non décomposables du monde réel, ces entités sont nommées "*individus*" et sont modélisés par des descriptions de propriétés qu'elles satisfont, sous forme de vecteurs. La seconde est définie par des concepts représentés à

l'aide de classes d'individus. Les classes sont décrites sous forme de vecteurs de même dimension que ceux des individus, en prenant en compte leur variation. Les concepts sont modélisés par des objets symboliques [Vra02],[DKBM00].

2.3.2 Présentation formelle de l'analyse de données symboliques

L'analyse des données symboliques est définie sur deux espaces, l'espace des individus et l'espace des descriptions. Nous posons les notations suivantes :

- Ω l'ensemble des individus
- D l'ensemble des descriptions
- Y une application de Ω dans D, associant à chaque individu de Ω ou à chaque classe d'individus de Ω sa description.

Ainsi D permet de décrire une classe d'individus ou un seul individu considéré comme une classe réduite à un seul élément. Avec ces notations, nous définissons ce qu'est un objet symbolique.

Définition 2.1. *Un objet symbolique est un triplet (a,R,d) avec :*

- *R un opérateur de comparaison entre 2 descriptions (le résultat peut être à valeurs dans $\{0,1\}$ ou dans $[0,1]$)*
- *d est une description*
- *a une application de Ω à valeurs dans $L = \{0,1\}$ ou dans $L = [0,1]$ qui associe à un individu w de Ω un degré d'adéquation entre sa description et la description d. La fonction a est nommée "fonction de reconnaissance"*

Nous notons :

$$a(w) = [Y(w)Rd]$$

Selon que $L = \{$ vrai, faux $\}$ ou $L = [$ 0,1 $]$, nous définirons deux types d'objets symboliques :

- **les objets symboliques booléens :**
  lorsque la fonction de reconnaissance a est à valeurs dans $\{0,1\}$. Un individu w vérifie ou ne vérifie pas les propriété de l'objet symbolique.
- **les objets symboliques modaux :**
  lorsque la fonction de reconnaissance a est à valeurs dans $[0,1]$. Un individu w a une probabilité de vérifier les propriétés de l'objet symbolique.

De façon générale, on définit un objet symbolique comme une description qui s'exprime sous forme d'une conjonction d'événements portant sur les valeurs prises par des variables.[Bri91], on peut dire aussi que les objets symboliques permettent une adéqua-

tion plus grande à la réalité multidimensionnelle que les objets habituellement utilisés en analyse des données.

les objets symboliques booléens

Si la base des données symboliques contient p variables, nous notons :

- $Y(w) = (Y_1(w), \ldots, Y_p(w))$
- $D = D_1 \times \ldots \times D_p$
- $d = (d_1, \ldots, d_p) \in D$
- $R = (R_1, \ldots, R_p)$ avec R_i la relation sur D_i.

Nous appelons "assertion" une conjonction d'objets symboliques. L'assertion est un cas particulier d'un objet symbolique défini par $s = (a, R, d)$ avec

$$a(w) = \wedge_{i=1}^{p}[Y_i(w)R_i d_i]$$

Exemple 2.1. *Soit une classe $C = (w_1, w_2, w_3)$ de Comme dans le tableau.*
L'assertion booléenne adossée à C peut être :

$$a(w) = [Age(w) \subseteq [37, 45]] \wedge [Taille(w) \subseteq [1.75, 1.82]] \wedge [Continent(w) \subseteq \{Afrique, Europe\}]$$

Pays	Age	Taille	continent
w_1	37	1.75	Afrique
w_2	45	1.82	Afrique
w_3	42	1.80	Europe

TAB. 2.1 – Tableau des données Symboliques

Nous posons $d_1 = [1.75, 1.82], d_2 = [37, 45], d_3 = \{Afrique, Europe\}$ et $R_1 = R_2 = R_3 = \subseteq$ et l'objet symbolique associé à C est :

$$a(w) = \begin{cases} 1(= vrai) \ Si \ pour \ tout \ i, X_i(w) \in d_i \\ 0(= faux) \ Sinon \end{cases}$$

les objets symboliques modaux

Les objets symboliques modaux sont définis avec des fonctions d'agrégation et de comparaison entre descriptions modales des individus de Ω et celles de l'espace des descriptions

D.
Si un individu w de Ω a des descriptions probabilistes $F_1^w, \ldots, F_p^w$ par rapport à p variables et qu'on veut les comparer aux descriptions probabilistes $G_1, \ldots, G_p$, l'objet symbolique modal associé est

$$a(w) = f(g(F_1^w, G_1), \ldots, g(F_p^w, G_p))$$

avec g une fonction de comparaison entre les deux descriptions probabilistes F_i^w et G_i, f est une fonction d'agrégation sur toutes les variables descriptives. La fonction f induit un degré d'adéquation sur toutes les variables. Par exemple, f peut être un produit et g une distance entre fonctions de répartition.
Les objets symboliques permettent donc de comparer des individus à des descriptions.

2.3.3 Les apports de l'analyse de données symboliques

Par rapport aux approches classiques, l'analyse des données symboliques présente les caractéristiques et ouvertures suivantes :

- Elle s'applique à des données plus complexes. En entrée, elle part de données symboliques (variables à valeurs multiples, intervalle, histogramme, distribution de probabilité,...) munies de règles et de taxonomies et peut fournir en sortie des connaissances nouvelles sous forme d'objets symboliques.
- Elle utilise des outils adaptés à la manipulation d'objets symboliques de généralisation et spécialisation, d'ordre et de treillis, de calcul d'extension, d'intention et de mesures de ressemblance tenant compte des connaissances sous-jacentes basées sur les règles et taxonomies.
- Elle fournit des représentations graphiques exprimant entre autres la variation interne des descriptions symboliques. Par exemple, en analyse factorielle, un objet symbolique sera représenté par une zone, elle-même exprimable sous forme d'objet symbolique, et pas seulement par un point.

2.3.4 Les principaux avantages des objets symboliques :

Nous citons ici quelques avantages à utiliser des objets symboliques :

- Ils fournissent un résumé de la base plus riche que les données agrégées habituelles car ils tiennent compte de la variation interne et des règles sous-jacentes aux classes décrites, ainsi que des taxonomies fournies. (proche du langage de l'utilisateur), en exprimant des descriptions basées sur des propriétés concernant les variables initiales ou les variables significatives.

- Ils sont explicatifs puisqu'ils s'expriment sous forme de propriétés des variables initiales ou de variables significatives obtenues (axes factoriels).
- En étant indépendant de la base des données, ils peuvent identifier n'importe quel individu vérifiant leurs descriptions dans n'importe quelle base de données.
- En utilisant leur partie descriptive, ils permettent de construire un nouveau tableau de données de plus haut niveau sur lequel une analyse de données symbolique de second niveau peut s'appliquer.
- Afin de modéliser des concepts, ils peuvent aisément exprimer des propriétés joignant des variables provenant de plusieurs tableaux associés à différentes populations. Par exemple, pour construire un objet symbolique associé à une ville, on peut utiliser des propriétés issues d'une relation décrivant les habitants de chaque ville et une autre relation décrivant les foyers de chaque ville.
- Afin de modéliser des concepts, ils peuvent aisément exprimer des propriétés joignant des variables provenant de plusieurs tableaux associés à différentes populations. Par exemple, pour construire un objet symbolique associé à une ville, on peut utiliser des propriétés issues d'une relation décrivant les habitants de chaque ville et une autre relation décrivant les foyers de chaque ville.
- Ils peuvent être facilement transformés sous forme de requête d'une base des données. Ceci a au moins les deux conséquences suivantes : Ils peuvent propager les concepts qu'ils représentent d'une base à une autre (par exemple, d'un pays à l'autre de la communauté européenne).
- Au lieu d'analyser une gigantesque base de données constituée de plusieurs bases, nous pouvons résumer chaque base de données par des objets symboliques et appliquer l'analyse de données symboliques à l'ensemble des objets symboliques.
- Les objets symboliques formés à partir de la base par les outils de l'Analyse des données symboliques permettent de définir des requêtes et donc de fournir des questions qui peuvent être pertinentes.

L'analyse des données symboliques dispose maintenant d'un cadre théorique et informatique permettant d'étendre et de renouveler toutes les méthodes de la statistique multidimensionnelle afin de les mettre en adéquation avec les données complexes et en quantité gigantesque dont nous disposons maintenant du fait des progrès de la technologie informatique. Cela va permettre de décupler les recherches et les applications dans tous les domaines où l'on dispose de données de niveau et de complexité plus élevés puis d'en extraire des connaissances.

Chapitre 3

La classification

Sommaire

3.1 Introduction **20**

3.2 Définitions **22**

3.3 Notion de ressemblance entre objets **25**

3.3.1 Les mesures de proximité sur un ensemble 27

3.3.2 Différents types d'indices de dissimilarité 27

3.4 Présentation des méthodes de classification de données . . . **31**

3.5 Méthodes hiérarchiques **34**

3.5.1 Classification Ascendante Hiérarchique (CAH) 35

3.5.2 Classification Hiérarchique Descendante(CHD) 40

3.6 Classification par partition (CPP) **44**

3.6.1 Méthodes k-means 46

3.6.2 Méthodes k-medoids 49

3.6.3 TAXY 53

3.7 Classification par la densité **57**

3.7.1 Approche basée sur la connexité de densité 58

3.7.2 Approche basée sur la fonction de densité 60

3.8 Classification basée sur la quantification par grille **61**

3.9 Autres Méthodes **64**

3.10 Comparaison de classification **66**

3.11 Conclusion **69**

3.1 Introduction

La notion de classification est essentielle en science, elle permet aux scientifiques de mettre de l'ordre dans les connaissances qu'ils ont sur le monde. Ainsi, depuis longtemps des chercheurs et des scientifiques ont essayé de classer les espèces animales. De nombreuses classifications ont été créées. Face à ces classifications, les scientifiques sont souvent incapables de désigner la meilleur d'entre elles. Car chacune présente un intérêt par rapport aux autres et à la tâche considérée.

Le terme de classification est associé à la notion d'abstraction. En effet, une classification permet de synthétiser des informations dans des groupes plus généraux. C'est une forme d'abstraction, puisque l'on va mettre de côté les descriptions exactes des objets et ne faire ressortir que les traits particuliers que certains d'entre eux ont en commun.

L'importance de la classification dans les sciences se reflète dans la grande variété des domaines où tant leur nature que leur construction ont fait l'objet des recherches.

Dans le cadre d'un problème de classification, on dispose d'un ensemble des données qui reprend une collection d'individus (objets) non étiquetés. Les classes sont encore inexistantes. L'objectif est alors d'obtenir des classes d'objets homogènes, en favorisant l'hétérogénéité entre ces différentes classes.

L'un des problèmes essentiels en classification consiste donc à définir quelle classification pour une personne et à un instant donné, a plus de "*sens*".

On peut donc dire que toutes les techniques de classification suivent un même principe général qui consiste à minimiser la distance entre deux individus d'une même classe et à maximiser la distance entre deux individus de classes distinctes.

Une définition formelle de la classification, qui puisse servir de base à un processus automatisé, amène a se poser les questions suivantes :

- Comment les objets à classer sont-ils définis ?
- Comment définir la notion de ressemblance entre objets ?
- Qu'est-ce qu'une classe ?
- Comment sont structurées les classes ?

– Comment juger une classification par rapport à une autre ?

A toutes ces questions, des réponses exhaustives peuvent être trouvées dans l'excellent ouvrage de Gordon [Gor81, Eve93].

(Les références générales concernant la classification des données les plus cités dans la littérature sont [Gor81, Eve93, Fas99, GKR00, HK01, Har75, JD88, JMF99, KR90, Mir96]).

Une introduction à l'exploitation contemporaine des données et des techniques de classification peut être trouvées dans Han et Kamber [HK01].

La classification est liée à beaucoup d'autres champs. La classification a été employée couramment dans les statistiques [AH96], la science [MK83], la reconnaissance de forme [DH73] et segmentation et traitement d'images [DLR77], [GGR99].

Les méthodes et les algorithms des classification ont été utilisés dans d'autres domaines d'application. Nous citons brièvement :
– La recherche documentaire (texte mining) [DFG01, SKK00] ;
– Analyse spatiales des données [EFKS00, XEKS98] ;
– Applications aux données de Web [CMS99, FWZ01, HC01] ;
– Analyse d'ADN dans la bi-informatique [BDSY99].

Ce chapitre est organisé en plusieurs paragraphes, nous présentons d'abord dans le premier paragraphe quelques définitions formelles et essentielles pour comprendre les différentes techniques de classification en passant par les objectifs principaux d'une classification et les étapes principales d'une classification comme une procédure automatisée.

Dans le deuxième paragraphe, nous présentons la notion d'une distance entre les individus à classer car cette notion est essentielle pour la majorité des techniques de classification en citant quelques mesures souvent citées dans les littératures spécialisées, nous citons ces mesures selon le type de variable à savoir dans les cas d'une variable binaire, numérique, nominale, etc

Le troisième paragraphe sera consacré à présenter les différentes techniques de classification. Nous divisons ces techniques en plusieurs familles :

- Méthodes hiérarchiques,
- Méthodes de partitionnement,
- Méthodes basées sur la densité,
- Méthodes basées sur la grille,
- Autres méthodes.

Nous présentons aussi pour chaque technique les algorithmes les plus connus, leurs avantages et leurs inconvénients. La figure (3.2) montre le diagramme de différentes techniques avec les algorithmes cités dans ce document.

A la différence de la classification supervisée, qui, étant donné un ensemble de classe déjà identifiées, il s'agit de trouver la meilleure classe à laquelle un individu appartient. La classification non supervisée consiste à structurer des classes non encore identifiées qui groupent ces individus.
Dans cette thèse nous nous intéressons à la classification non supervisée.

Les étapes d'une classification automatique
Les différentes étapes sd'une classification automatique sont :

1. Le choix des données.
2. Le calcul des ressemblances entre les individus à partir du tableau initial.
3. Le choix d'un algorithme de classification et exécution.
4. L'interprétation des résultats :
   - évaluation de la qualité de la classification,
   - description des classes obtenues.

Dans le paragraphe suivant nous présentons les définitions essentielles pour bien comprendre la classification.

3.2 Définitions

La classification est une méthode d'analyse des données qui vise à regrouper en classes homogènes un ensemble d'observations. Ces dernières années, les besoins d'analyse de données et en particulier de classification ont augmenté significativement. En effet, de plus en plus de domaines scientifiques nécessitent de catégoriser leurs données dans un

but descriptif ou décisionnel.

La notion de partition est un élément fondamental dans la pratique de la classification. Classification sur $\mathcal{I}$ signifie déterminer sur $\mathcal{I}$ une ou plusieurs partitions.

Définition 3.1. *Partition*

On appelle partition de $\mathcal{I}$, l'ensemble $\mathcal{P}$, $\mathcal{P} = \{C_i, i \in I\}$
I : ensemble d'indices.
C_i : partie de $\mathcal{I}$ (ou une classe) possédant les propriétés :

1. $\forall i \in I, C_i \neq \phi$
2. $\forall i \in I, \forall j \in I, i \neq j \qquad \longrightarrow C_i \cap C_j \neq \phi$
3. $\bigcup_{i \in I} C_i = \mathcal{I}$

Comparaison de partitions :

Soit $\mathcal{P}$ et $\mathcal{P}'$ deux partitions de $\mathcal{I}$: $\mathcal{P} = \{C_i, i \in I\}$ et $\mathcal{P}' = \{C'_j, j \in I'\}$

$\mathcal{P}$ est plus fine que $\mathcal{P}'$ (ou $\mathcal{P}'$ est moins fine que $\mathcal{P}$) si et seulement si $\forall i \in I$, $\exists j \in I', C_i \subset C'_j$
(Chaque élément de $\mathcal{P}'$ est une réunion d'éléments de $\mathcal{P}$)

$\mathcal{P}_D$ est dite une partition discrète si $\mathcal{P}_D = \{\{x\}, x \in \mathcal{I}\}$
$\mathcal{P}_G$ est dite une partition grossière si $\mathcal{P}_G = \{\mathcal{I}\}$

Définition 3.2. *Recouvrement*

On appelle un recouvrement de $\mathcal{I}$, l'ensemble $\mathcal{P}$, $\mathcal{P} = \{C_i, i \in I\}$
I : ensemble d'indices.
C_i : partie de $\mathcal{I}$ (ou une classe)
possédant les propriétés :

1. $\forall i \in I, C_i \neq \phi$
2. $\bigcup_{i \in I} C_i = \mathcal{I}$

Hiérarchies de parties :

Une hiérarchie est une ensemble de parties de $\mathcal{I}$ issues d'une suite emboitée de partitions, et qu'on peut représenter par un dendrogramme.

Définition 3.3. *Hiérarchie*

On appelle hiérarchie de parties de $\mathcal{I}$, tout sous-ensemble, $\mathcal{H}$ de $\mathcal{P}(I)$ tel que :

1. $\Phi \notin \mathcal{H}$
2. $\forall x \in \mathcal{I}, \qquad \{x\} \in \mathcal{H}$
3. $\mathcal{I} \in \mathcal{H}$
4. $\forall H_1 \in \mathcal{H}, \forall H_2 \in \mathcal{H} \Rightarrow H_1 \cap H_2 = \phi \quad ou \quad H_1 \subset H_2 \quad ou \quad H_2 \subset H_1$

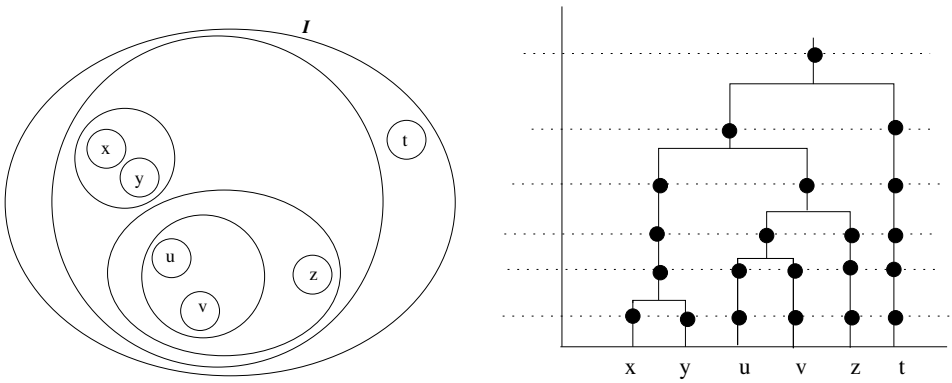

FIG. 3.1 – Exemple d'une hiérarchie de parties de $\mathcal{I}$

Hiérarchie indicée de parties :

Définition 3.4. *on appelle hiérarchie indicée de parties de $\mathcal{I}$, la donnée $(\mathcal{H}, h)$ où $\mathcal{H}$: Hiérarchie de parties*
$h : \mathcal{H} \longrightarrow \mathbb{R}^+$ tel que
$\forall x \in \mathcal{I}, h(\{x\}) = 0 \forall H_1 \in \mathcal{H}, \forall H_2 \in \mathcal{H}, H_1 \subseteq H_2 \Rightarrow h(H_1) \leq h(H_2)$
$h(H)$ s'appelle niveau hiérarchique de H.

Hiérarchie indicée et partition

une hiérarchie indicée permet de construire des familles de partition totalement ordonnées :
Soit $s \in \mathbb{R}$ et $\mathcal{P}$, l'ensemble des élémentes h de $\mathcal{H}$ tel que :

- $h(H) \leq s$
- $\forall H' \in \mathcal{H}, \quad$ si $\quad h(H') \leq s \quad$ et $\quad H' \neq H \Rightarrow H' \subset H$

alors on montre que l'ensemble $\mathcal{P}(s)$ est l'ensemble des élémentes de $\mathcal{H}(s)$

$$\mathcal{H}(s) = \{H \in \mathcal{H}, \qquad h(H) \leq s\}$$

Réciproquement, à partir de toute famille totalement ordonnée de partition de $\mathcal{I}$ (y compris, $\mathcal{P}_D$ et $\mathcal{P}_G$), on peut construire une hiérarchie indicée.

En effet, soit $P_G \leq P_1 \leq P_2 \leq ..... \leq P_m \leq P_D$
$\Rightarrow \mathcal{H} = P_G \cup P_D \cup (\bigcup_{i=1}^{m} P_i)$
on construit $\mathcal{F}$ tel que :
$\mathcal{F}(\mathcal{P}_D) = 0$
$\forall i \in \{1, ..., m\}, \forall j \in \{1, ..., m\}$
$\mathcal{P}_i \leq \mathcal{P}_j \Rightarrow \mathcal{F}(\mathcal{P}_i) > \mathcal{F}(\mathcal{P}_j)$

et on définit h tel que :
$\forall H \in \mathcal{H}, \qquad h(H) = \mathcal{F}(\mathcal{P}_i)$
où $\mathcal{P}_i$: est le plus fine des partitions contenant H.

Définition 3.5. *La classification automatique est un ensemble des techniques qui fournissent directement une ou plusieurs partitions d'un ensemble ; certaines d'entre elles, dites de classification hiérarchique, permettent d'obtenir des partitions "non contradictoires" qui sont présentées sous forme d'un arbre de classification.*

Définition 3.6. *Une taxinomie est un système par lequel des classes sont reliées les unes aux autres par inclusion de classes. Plus une classe est inclusive dans une taxinomie, plus élevé est le niveau d'abstraction de cette classe. Chaque classe d'une taxinomie est entièrement incluse dans une autre classe (à moins qu'elle ne soit la classe du niveau le plus élevé), mais n'épuise pas cette classe plus inclusive.*

La notion de distance est centrale en classification, pour cela nous présentons d'abord les différentes définitions d'une distance et nous présentons ensuite quelques distances selon le type de variables.

3.3 Notion de ressemblance entre objets

Nous avons dit que l'objectif de la classification est de constituer des groupes d'objets homogènes et différenciés, i.e. des groupes d'objets tels que :

– Les objets soient les plus similaires possibles au sein d'un groupe

– Les groupes soient aussi dissemblables que possible

La ressemblance ou la dissemblance étant mesurée sur l'ensemble des variables descriptives.
Pour mesurer la ressemblance entre deux objets, deux démarches sont envisageables :

– On peut dire que deux objets sont semblables s'ils partagent une certaine caractéristique. (Considérons le nombre de doigts d'un être vivant et comparons, les deux espèces seront jugées semblables. Ce genre de démarche aboutit à une classification **monothétique**. Tous les objets d'une même classe partagent alors un certain nombre de caractéristiques.
  (e.g : "tous les hommes sont mortels").
– On peut aussi mesurer la ressemblance en utilisant une mesure de proximité (distance, dissimilarité). Dans ce cas la notion de ressemblance est mesurée de façon plus floue et deux objets d'une même classe posséderont des caractéristiques "proches" au sens de la mesure utilisée. Cette démarche est dite **polythétique**.

Nous avons dit que les méthodes de classification traditionnelles procèdent au rapprochement d'objets sur une base de mesures de distances calculées à partir des valeurs prises par les attributs de l'objet. En réalité, en substitut à la notion de distance, d'autres notions peuvent être utilisées : similarité ou dissimilarité.
La mesure de proximité pourra être effectuée soit entre des objets, soit entre des classes. Ainsi, selon les méthodes de classification, on peut être confronté à trois types de rapprochement : entre deux objets, entre un objet et une classe, entre deux classes.
La notion de proximité est centrale dans les méthodes de classification traditionnelles. Il est donc important de bien comprendre sa raison d'être, ses variantes et les impacts du choix d'une mesure particulière sur les classes.
Pour la suite nous désignons par la distance ou (la dissimilarité) entre deux classes la même définition entre deux individus (objets). Sachant qu'un individu peut être une classe.

Généralement quatre groupes de mesures de dissimilitude peuvent être distingués :

1. Coefficients de corrélation, également marqués mesures d'association.

2. Mesures de distance.

3. Mesures dérivées. Elles sont dérivées de coefficients de corrélation ou des distances.

4. D'autres mesures de dissimilarité ou de similarité. Elles ont été développées pour des buts spéciaux, principalement pour des variables binaires.

3.3.1 Les mesures de proximité sur un ensemble

On appelle similarité ou dissimilaruté, toute application à valeurs numériques qui permet de mesurer le lien entre les individus d'un même ensemble ou entre les variables.

Définition 3.7. *Un indice de similarité sur $\mathcal{I}$ est une application*

$$Sim : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}^+$$

ayant la propriété :
$\forall x \in \mathcal{I}, \forall y \in \mathcal{I}$ *avec* $x \neq y$

- *P(1) :* $Sim(x, y) = Sim(y, x)$
- *P(2) :* $Sim(x, x) = Sim(y, y) = Sim_{max} > Sim(x, y)$

Définition 3.8. *Un indice de dissimilarité sur $\mathcal{I}$ est une application*

$$Diss : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}^+$$

ayant la propriété :
$\forall x \in \mathcal{I}, \forall y \in \mathcal{I}$ *avec* $x \neq y$

- *P(3) :* $Diss(x, y) = Diss(y, x)$
- *P(4) :* $Diss(x, x) = 0$

Nous avons conservé les termes "similarité" et "dissimilarité" pour respecter l'usage mais il serait plus correct de les remplacer respectivement par "ressemblance" et "dissemblance".

Seuls les indices de dissimilarité seront considérés par la suite, car à un indice de similarité Sim peut toujours être associé l'indice de dissimilarité défini par :
$diss(x, y) = Sim_{max} - Sim(x, y)$.

3.3.2 Différents types d'indices de dissimilarité

Les propriétés retenues pour définir un indice de dissimilarité ont l'avantage d'être peu contraignantes ; elles ont par contre le désavantage de permettre ce que certains appelleront des incohérences :
Rien n'empêche par exemple d'avoir simultanément
$x \neq y$ et $Diss(x, y) = 0$
il existe z tel que $Diss(x, z) \neq Diss(y, z)$

Cette incohérence disparaît si l'on impose à l'indice de dissimilarité de posséder l'une au moins des deux propriétés suivantes :

- P(5) : $Diss(x, y) = 0 \Rightarrow x = y$
- P(6) : $Diss(x, y) \leq Diss(x, z) + Diss(y, z)$ pour tout $(x, y, z) \in \mathcal{I}^3$ (Inégalité triangulaire)

Les indices de dissimilarité vérifiant la propriété P(5) sont appelés des "*indices de distance*" ; ceux qui vérifient la propriété P(6) sont appelés des "*écarts*" ; ceux, enfin, qui vérifient ces deux propriétés sont appelés des "*distances*".

En analyse linéaire, on utilise des écarts euclidiens ; en classification automatique, on rencontre des "*distances ultramétriques*" ; ces dernières vérifient la propriété :

- P(7) : $\forall (x, y, z) \in \mathcal{I},\ Diss(x, y) \leq max\{Diss(x, z), Diss(y, z)\}$

Le tableau suivant donne un bilan des définitions :

Indices de dissimilarité	Propriétés		
	P(5)	P(6)	P(7)
Indice de distance	x		
écart		x	
écart ultramétrique		x	x
Distance	x	x	
distance ultramétrique	x	x	x

TAB. 3.1 – Bilan des définitions des indices de dissimilarité

Les mesures de proximité entre les objets à classer dépendent de la nature des variables mesurées. Nous allons donner plusieurs indices souvent cités dans la littérature et nous citons ces indices par rapport de la nature de variables.

Mesures de proximité dans le cas des données numériques

Une mesure générale de la distance dans le cas des données numériques est donnée par l'indice de *Mikowski* défini de la manière suivante : $\forall x, y \in \mathcal{I}^p$

$$d(x, y) = [\sum_{i=1}^{p} \mid x_i - y_i \mid^q]^{\frac{1}{q}}$$

Remarque 3.1. *Les trois distances suivantes d_1, d_2 et d_3 sont des cas particuliers de l'indice de* Minkowsky *:*

$$d_1(x,y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$$

$$d_2(x,y) = \sum_{i=1}^{p} \mid x_i - y_i \mid$$

$$d_3(x,y) = \max\{\mid x_i - y_i \mid\} \qquad i = 1, \ldots\ldots, p$$

$q = 1 \Rightarrow d = d_2$ *(distance* Manhattan (city-block)*)*
$q = 2 \Rightarrow d = d_1$ *(distance* euclidienne*)*
$q = \infty \Rightarrow d = d_3$ *(distance* Tchebycev*)*

Dissimilarités sur un ensemble de parties :

Soit $\mathcal{P}(\mathcal{I})$, l'ensemble des parties de $\mathcal{I}$:
On suppose que pour tout $x \in \mathcal{I}$, il existe un nombre réel p_x (poids de x) tel que : $\forall x \in \mathcal{I}, p_x \geq 0$ et $\sum_{x \in \mathcal{I}} p_x = 1$
Si $A \in \mathcal{P}(\mathcal{I})$, on notra p_A la somme des poids $p_x (x \in A)$.
Soit $F(\mathcal{I}, \mathbb{R})$ l'ensemble des applications de $\mathcal{I}$ dans $\mathbb{R}$ et f une application injective : $f : \mathcal{P}(\mathcal{I}) \to F(\mathcal{I}, \mathbb{R})$

On peut définir une disimilarité de la manière suivante : $d : \mathcal{P}(\mathcal{I}) \times \mathcal{P}(\mathcal{I}) \longrightarrow \mathbb{R}^+$
$\forall A, B \in \mathcal{P}(\mathcal{I})$

$$d(A,B) = \Big[\sum_{x \in \mathcal{I}} p_x |f_A(x) - f_B(x)|^q\Big]^{\frac{1}{q}} \qquad (q \geq 1)$$

Mesures de proximité dans le cas des données binaires

Si on s'intéresse qu'aux caractères présence-absence, ce que l'on peut coder par 1 et 0 respectivement. Le calcul de l'indice de dissimilarité entre deux individus compte tenu les occurrences du couple d'individus considérés.

Les indices de distance pour les données dichotomiques sont dérivés de la construction du tableau suivant :

Où :
a est le nombre de presence (cas 1) pour le couple d'individus.
b est le nombre de presence pour l'individu i et absence pour l'individu j.
c est le nombre d'absence pour l'individu i et présence pour l'individu j.
d est le nombre d'absence (cas 0) pour le couple d'individus .
p le nombre total de variables.

		Individu j		
		$x_{jk} = 1$	$x_{jk} = 0$	
Individu i	$x_{ik} = 1$	a	b	$a + b$
	$x_{ik} = 0$	c	d	$c + d$
		$a + c$	$b + d$	p

TAB. 3.2 – Tableau de caractères pour les données dichtomiques

Une grande variété d'indices ont été développée pour des données binaires (environ 27). Nous citons dans le tableau 3.3 des indices souvent cités dans la littérature spécialisée pour mesurer la ressemblance entre deux individus i et j

Indice	Forme	Commentaire
$I_{jaccard}$	$\frac{a}{a+b+c}$	le nombre d'absence est ignoré
I_{Sokal_1}	$\frac{2(a+d)}{2(a+d)+b+c}$	Pondération double des cooccurences.
I_{Sokal_2}	$\frac{a}{a+2(b+c)}$	Pondération double des non cooccurences, le nombre d'absence est ignoré.
I_{Ochiai}	$\frac{a}{\sqrt{(a+b)(a+c)}}$	Forme dichotomique d'un cosinus.
I_{Rogers}	$\frac{a+d}{a+d+2(b+c)}$	Contrairement à l'indice I_{Dic}
I_{Russel}	$\frac{a}{a+b+c+d}$	Proportion d'occurrences positives
I_{Simple}	$\frac{a+d}{a+b+c+d}$	Proportion d'occurrences positives et négatives
I_{Dic}	$\frac{2a}{2a+b+c}$	le nombre d'absence est ignoré et le nombre de présence est doublé

TAB. 3.3 – Tableau des indices en cas des données binaires

Mesures de proximité dans le cas des données nominales

$$d(x, y) = \frac{p - m}{p} \quad ; \quad m = a + d$$

Tels que m est le nombre d'attributs dont la valeur est pareille dans tous les deux objets, p est le nombre total d'attributs.

3.4 Présentation des méthodes de classification de données

Les premières approches proposées en classification étaient algorithmiques, heuristiques ou géométriques et reposaient essentiellement sur la dissimilarité entre les objets à classer. L'approche statistique, plus récente, se base sur des modèles probabilistes qui formalisent l'idée de classe. Cette approche permet en outre d'interpréter de façon statistique la classification obtenue.

Formalisation générale du problème de la classification

Soit $\mathcal{I}$ un ensemble des individus (Objets),$\mathcal{I} = \{w_1, w_2, \ldots, w_n\}$

A partir de critères donnés, le problème consiste à réaliser une partition de $\mathcal{P} = \{C_1, C_2, \ldots, C_k\}$ tell que :

$$C_1 \cup C_2 \ ... \ \cup C_k = \mathcal{P}$$

$$C_i \cap C_j = \phi \text{ avec } i, j = 1, 2, \ldots, k$$

Suivant la nature du problème posé, on se fixe à l'avance ou non le nombre k des classes C_i désirées.

Remarque :
Une classification automatique obtenue sur un ensemble n'est jamais **la classification** de cet ensemble, mais une classification (parmi beaucoup d'autres) établie à partir des variables et des méthodes choisies intentionnellement.

Questions importantes
Le choix d'un algorithme de classification est très compliqué et dans le plus part des cas nous devons répondre à plusieurs questions avant de faire notre choix :

- Le type d'attributs qu'un algorithme peut manipuler
- Capacité de travail avec des données d'une très grande dimension
- Possibilité de travail avec une grande base des données
- Capacité de trouver des groupes dans un ensemble des données irrégulières
- Complexité de temps
- Dépendance d'ordre de données
- Confiance dans la connaissance a priori et des paramètres définis pour l'utilisateur

– Possibilité d'interpréter des résultats

Après avoir exposé quelques définitions, nous allons présenter quelques méthodes de classification. Notre présentation sera basée sur le figure (3.2).

Ce qui distingue les différentes méthodes

Trois éléments permettent de caractériser les différentes méthodes :

1. La classification procède séquentiellement en regroupant les individus les plus "semblables" en premier lieu (méthodes hiérarchiques) ou elle regroupe en k groupes tous les individus simultanément (méthodes de division).
2. Le critère de "ressemblance" entre deux individus.
3. Le critère de "ressemblance" entre deux classes.

Nous allons présenter quelques méthodes de classification en divisant ces méthodes comme suit 3.2 :

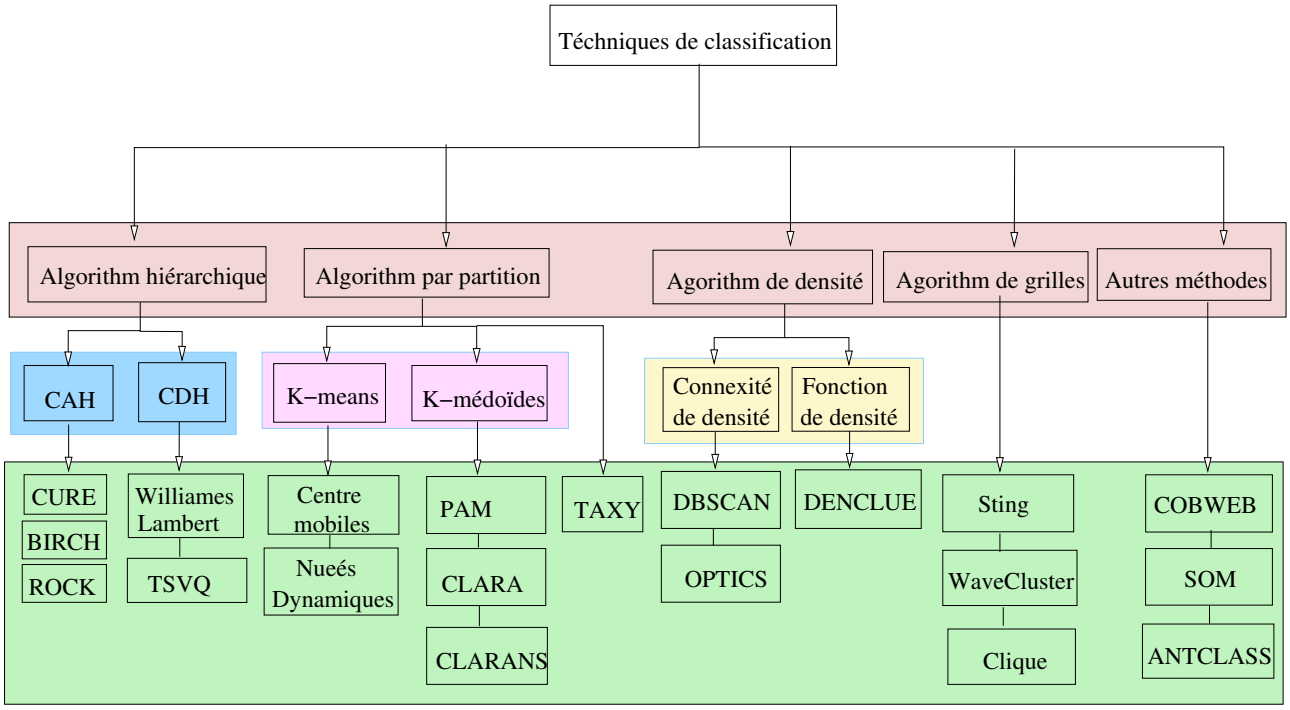

FIG. 3.2 – Techniques de classification

I- **Algorithmes hiérarchiques,**

– **Classification Hiérarchique Ascendante (CHA)**, (CURE, BIRCH, ROCK)

- **Classification Hiérarchique Descendante(CHD)**,
  (Williames et Lambert, TSVQ)

II- **Algorithmes par partition,**
- **k-means,**
  (*Centres mobiles, Nuées Dynamiques*)
- **k-médoïdes**,
  (*PAM, CLARA, CLARANS*)
- **TEXY**

III- **Algorithmes basés sur la densité,**
- Approche basée sur la connexité de densité
  (*DBSANS, OPTICS, DBCLASD*)
- Approche basée la fonction de densité
  DENCLUE

IV- **Algorithmes de Grilles,**
(*STING, CLIQUE, WaveCluster*)

V- **Autres méthodes.**
(*ANTCLASS, COBWEB*)

Les algorithms recensés de classement ne représentent pas un travail exhaustives. Nous les citons ici à titre d'exemple et dans le cadre historique du développement des techniques de classification.

Dans les littérature, **J. Sidi** en 1980 a présenté [Sid80] une méthode de classification par division en utilisant la méthode des nuées dynamiques de Diday [DGLS80] où la distance entre éléments dépend de la classe. **MICHALSKI ET STEPP (1983)** [MS83a] ont proposé une méthode de classification hiérarchique connue sous le nom d'*"algorithme de l'étoile"*. Il s'agit d'un algorithme à deux couches. La couche intérieur basée sur la classification par division qui détermine une partition d'un ensemble d'individus en k classes. La couche extérieure basée sur la classification hiérarchique consiste à appliquer récursivement l'algorithme intérieur, pour obtenir une classification de type hiérarchique.

Par la suite, **BENAMOU ET KODRATOFF (1986)** [BK86] ont proposé une méthode de classification ascendante hiérarchique avec l'indice de maximum qui est basé sur la détermination d'un vecteur de similarité, qui liste les valeurs communes à deux objets, et d'un vecteur de dissimilarité qui tient compte de la partie non commune. La valeur de

l'indice de dissimilarité entre deux objets est alors fonction de ces deux valeurs.

En 1988, **LERMAN et Nicolas** [LNP88], ont proposé une méthode de classification purement numérique et constituée en deux étapes :

- Classification hiérarchique de l'ensemble des individus en utilisant une méthode de la famille VL (Vraisemblance du Lien),
- Caractérisation statistique des nœuds significatifs, traduisant l'influence d'une même variable sur un ensemble organisé de classes d'individus.

3.5 Méthodes hiérarchiques

Il y a deux genres de méthodes de classification méthodes de classification hiérarchiques . La **C**lassification **A**scendante **H**iérarchique **CAH** et la **C**lassification **D**escendante **H**iérarchique **CDH**.

- La **CAH** permet de construire une hiérarchie entière des objets sous la forme d'un "arbre" dans un ordre ascendant. On commence en considérant chaque individu comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe. Le processus est itéré jusqu'à ce que tous les individus se trouvent dans une même classe. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre des classes plus ou moins grand.

  Différentes mesures de la distance interclasses peuvent être utilisées : la distance euclidienne, la distance inférieure (qui favorise la création de classes de faible inertie) ou la distance supérieure (qui favorise la création de classes d'inertie plus importante) etc...

  Dans le cas de la classification ascendante hiérarchique, à partir des éléments, on forme des petites classes ne comprenant que des individus très semblables, puis à partir de celles-ci, on construit des classes de moins en moins homogènes, jusqu'à obtenir la classe tout entière.

  On trouve dans [Ben73], [CDLR81],[Jam99] ,[Sap90] et beaucoup d'autres auteurs des détails plus riches.
- Dans la **CDH**, on considérant tous les individus comme une seule classe au début, on divise successivement les classes en classes plus raffinées. Le processus marche jusqu'à ce que chaque classe contienne un seul point ou bien si l'on atteint un nombre de classes désiré.

3.5.1 Classification Ascendante Hiérarchique (CAH)

Le but du module de **CAH** est d'obtenir une classification automatique de l'ensemble d'individus. Elle commence par déterminer parmi les n individus, quels sont les 2 individus qui se ressemblent le plus par rapport à l'ensemble des p variables spécifiées. Elle va alors regrouper ces 2 individus pour former une classe. Il existe donc à ce niveau $(n-1)$ classes, une étant formée des 2 individus regroupés précédemment, les autres ne contenant qu'un unique individu. Le processus se poursuit en déterminant quelles sont les 2 classes qui se ressemblent le plus, et en les regroupant. Cette opération est répétée jusqu'à l'obtention d'une unique classe regroupant l'ensemble des individus.
Cette procédure est basée sur 2 choix :

- La détermination d'un critère de ressemblance entre les individus. La méthode laisse à l'utilisateur le choix de la dissimilarité.
- La détermination d'une dissimilarité entre classes : procédé appelé un critère d'agrégation. La méthode laisse à l'utilisateur le choix de ce critère.

Les critères d'agrégation

De nombreux critères d'agrégation ont été proposés les plus nonnus sont :

- **Le critère du saut minimal**
  La distance entre 2 classes C_1 et C_2 est définie par la plus courte distance séparant un individu de C_1 et un individu de C_2.

$$D(C_1, C_2) = min(\{d(x, y)\}, x \in C_1, y \in C_2)$$

- **Le critère du saut maximal**
  La distance entre 2 classes C_1 et C_2 est définie par la plus grande distance séparant un individu de C_1 et un individu de C_2.

$$D(C_1, C_2) = max(\{d(x, y)\}, x \in C_1, y \in C_2)$$

- **Le critère de la moyenne**
  Ce critère consiste à calculer la distance moyenne entre tous les éléments de C_1 et tous les éléments de C_2

$$D(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y)$$

  Avec :

- n_{C_1} : Le cardinal de C_1
- n_{C_2} : Le cardinal de C_2

- **Le critère de Ward**
  Ce critère ne s'applique que si on est muni d'un espace euclidien. La dissimilarité ρ entre 2 individus doit être égale à la moitié du carré de la distance euclidienne d. Le critère de Ward consiste à choisir à chaque étape le regroupement de classes tel que l'augmentation de l'inertie intra-classe soit minimal.

$$D(C_1, C_2) = \frac{n_{C_1} n_{C_2}}{n_{C_1} + n_{C_2}} d^2(g_{C_1}, g_{C_2})$$

  Avec :
  - g_{C_1} : Le centre de gravité de C_1
  - g_{C_2} : Le centre de gravité de C_2

- **Le critère des centres de gravité**
  La distance entre 2 classes C_1 et C_2 est définie par la distance entre leurs centres de gravité.

$$D(C_1, C_2) = d(g_{C_1}, g_{C_2})$$

La difficulté du choix du critère d'agrégation réside dans le fait que ces critères peuvent déboucher sur des résultats différents. Selon les plus parts des références le critère le plus couramment utilisé est celui du Ward.

Remarque :
Dans le cas de stratégie du saut minimal, plus une classe contient d'éléments, plus elle est attractive pour des éléments isolés. au contraire, dans le cas d'une stratégie du saut maximal, plus une classe contient d'éléments, moins elle est attractive pour les éléments isolés.

Le critère de Ward, aisée à mettre en oeuvre lorsque la classification est effectuée après une analyse factorielle (les objets à classer étant repérés par leurs coordonnées sur les premiers axes factoriels), constitue une excellente méthode de classification ascendante hiérarchique. En effet, il est basé sur sur la réunion des 2 classes qui minimise l'augmentation de l'inertie intra-classes.

Nous présentons dans la suite quelques algorithms basés sur le principe de **CAH** :

- Clustering Using REpresentatives(**CURE**)
- Balanced Iterative Reducing and Clustering using Hierarchies(**BIRCH**)
- Robust Clustering using links (**ROCK**)

Algorithme CURE

Son nom vient de **Clustering Using REpresentatives**, il a été proposé par *Guha et al* dans [GRS98], cet algorithme utilise un échantillon représentatif de l'échantillon total pour réduire la complexité temporelle des calculs. Cet échantillon sera divisé en sous-ensembles qui sont regroupés en sous-classes. Les sous-classes seront agrégées hiérarchiquement en utilisant la distance entre deux sous-classes C_1 et C_2, la plus petite distance entre un représentant de C_1 et un représentant de C_2 jusqu'à obtenir k de classes demandées.
Les détail de cette méthode est décrit dans l'algorithme suivant :

Données : k le nombre maximum de classes désiré.
Début
(1) Extraire un échantillon $\mathcal{I}_s$ de taille s de l'ensemble $\mathcal{I}$ d'individus
(2) Diviser l'échantillon $\mathcal{I}_s$ en p sous ensemble de taille s/p
(3) Déterminer une partition partielle de chaque sous-ensemble en s/pq sous-classes avec $q > 1$
(4) Éliminer les sous-classes d'effectif faible.
(5) Déterminer dans chaque sous-classes un ensemble de c points (la caractérisant points bien répartis dans la sous-classe).

(6) Agréger de façon hiérarchique les sous-classes $d(C_1, C_2) = min[d(x, y); \forall x$ représentant de C_1 et $\forall y$ représentant de $C_2]$
(7) Arrêter la procédure d'agrégation quand on obtient k classes.
(8) Classer l'ensemble $\mathcal{I}$ total en utilisant les c points représentant les k classes obtenues après l'agrégation. Chaque objet est affecté à la classe possédant le représentant qui lui est le plus proche.
FIN

Algorithme BIRCH

BIRCH (**B**alanced **I**terative **R**educing and **C**lustering using **H**ierarchies) a été développé par *Zhang et al* en 1996 [ZRL96, ZRL97]. C'est un algorithme qui travaille efficacement sur de gros jeux de données. L'idée principale du **BIRCH** est d'effectuer une classification sur un résumé compact des données, au lieu des données originales. C'est pourquoi il peut traiter un grand volume de données en utilisant une mémoire limitée.
Il est incremental, i.e. il a besoin d'un seul balayage du jeu de données. Il essaie de minimiser le coût d'entrée/sortie en organisant les données traitées en une structure d'arbre équilibré avec une taille limitée.
La communauté du domaine de classification trouve que **BIRCH** est l'un des meilleurs algorithmes qui peuvent traiter de gros jeux de données. Initialement, **BIRCH** a été créé pour traiter des données numériques, une extension a été proposée pour traiter des données de type catégoriel. Nous en parlerons au niveau algorithmique sans aborder en détail la mesure (le calcul) utilisée dans l'algorithme.

Deux phases principales caractérisent cette méthode :

- Phase (1) : Construction en mémoire d'un arbre de sous-classes et de leurs vecteurs caractéristiques (CF-tree) résumant les données volumineuses sur disque
- Phase (2) : application en mémoire d'une technique classique de clustering sur les sous-classes les plus fines de l'arbre déterminés en (1)

Le vecteur caractéristique d'une classe CF peut prendre les valeurs N, LS, SS

- N : C'est le nombre d'objets O_i dans la classe
- $LS = \sum X_i$
- $SS = \sum X_i^2$
- X_i : Le vecteur des variables décrivant l'objet O_i

Le vecteur caractéristique d'une classe C composée de deux sous-classes C_1 et C_2 est la somme des vecteurs caractéristiques de C_1 et C_2 permet un calcul incremental des vecteurs caractéristiques des classes construites.

A partir des vecteurs caractéristiques d'un ensemble de classes, on peut déterminer sans accéder aux données :

- Les centres de gravités et diamètres des classes,
- Les distances inter-classes moyennes et distances intra-classes classiques (moyennes ou entre centres de gravité) Ce qui permet de réaliser la phase (2) à partir des vecteurs caractéristiques des classes, sans accéder aux données de base

Principe de construction

Deux paramètres :

- Le nombre maximal B de fils par nœud
- Le diamètre maximal R d'une sous-classe

Au départ : Un arbre comportant une seule racine avec une seule sous-classe comportant un objet unique.
A chaque étape : Insertion d'un nouvel objet O dans l'arbre construit :

* Par une recherche en profondeur depuis la racine, on détermine la sous-classe la plus fine dont O est le plus proche et on incorpore O à cette sous-classe

* Si le diamètre de la sous-classe après insertion dépasse R :
  - On scinde la sous-classe en deux
  - On ajoute donc une sous-classe et un vecteur caractéristique à la feuille contenant la sous-classe initialement la plus proche de O
  - Si le nombre de sous-classes de ce nœud dépasse B, on le scinde en deux

* Dans le cas où un nœud est scindé en deux, on augmente le nombre de fils de son père de 1.

Si un arbre ne tient plus en mémoire, on augmente R et on construit un nouvel arbre à partir de l'arbre déjà construit et des données encore non examinées (utilisation de la propriété de réduction).

Algorithme ROCK

ROCK (**RO**bust **C**lustering using lin**K**s) a été développé par *Guha et al.* en 1999 [GRS00]. C'est un algorithme de classification hiérarchique. Il utilise le nouveau concept appelé liens (*links* en anglais) pour mesurer la similarité entre des individus au lieu des métriques pour les données numériques ou le coefficient de *Jaccard*.
Deux individus sont voisins si leur similarité dépasse un seuil θ : $sim(p, q) > \theta$
La similarité entre un pair d'individus peut être mesurée en se basant sur les distances métriques, le coefficient de *Jaccard* ou n'importe quelle fonction de similarité non métrique.
Le nombre de liens entre une paire d'individus est le nombre de leurs voisins communs. En général, les individus se trouvent dans une classe ont de nombreux voisins, par conséquent, de nombreux liens. En se basant sur le concept lines *"links"* entre 2 individus, on définit le lien ou l'interconnexité entre 2 classes par le nombre de liens croisés (*cross links*) entre eux. Ce nombre est calculée par la somme de liens entre tous les points situés dans 2

classes :

$$link(C_r, C_s) = \sum_{x_i \in C_r, x_j \in C_s} link(x_i, x_j)$$

Le nombre de liens dans une classe C_r donné par $n_r^{1+2f(\theta)}$ où $f(\theta)$ est souvent de la forme $\frac{1-\theta}{1+\theta}$ et la classification est effectuée en maximisant une fonction de critère :

$$E = \sum_{s=1}^{k} n_s \sum_{x_i, x_j \in C_s} \frac{link(x_i, x_j)}{n_s^{1+f(\theta)}}$$

On définit ainsi une fonction qui permet de décider si 2 classes sont fusionnées dans le processus de classification hiérarchique. Cette fonction est appelée *"goodness function"* :

$$g(C_r, C_s) = \frac{link(C_r, C_s)}{(n_r + n_s)^{1+2f(\theta)} - n_r^{1+2f(\theta)} - n_s^{1+2f(\theta)}}$$

Pendant la classification, deux classes dont *la goodness function* est maximale sont fusionnées.

Après un calcul initial de nombre de liens entre les individus, l'algorithme démarre en considérant chaque individu comme une classe et tente de fusionner les classes en basant sur *la goodness function.* L'algorithme se poursuit jusqu'à ce que le nombre spécifié de classes soit obtenu ou il n'y a plus de liens qui restent entre les individus.

ROCK ne traite pas toutes les données. Il prend un échantillon aléatoire d'un jeu de données pour faire la classification. Il est donc efficace pour faire la classification de gros jeux de données.
La complexité de **ROCK** est $O(n^2 + n\,(m_m m_a) + n^2\, logn)$ où m_m est le nombre maximum de voisins et m_a est le nombre moyen de voisins d'un individu, n est la taille de l'échantillon de données qu'il traite.

Les Avantage et les inconvénients de CAH

- **Avantage** : facile à implémenter,
- **Défaut** : méthode très coûteuse, complexité temporelle en $O(n^2)$.

3.5.2 Classification Hiérarchique Descendante(CHD)

Les méthodes de classification descendante hiérarchique partent d'un ensemble d'individus $\mathcal{I}$ et construisent, de manière itérative une partition de l'ensemble d'individus.

A l'inverse de la classification ascendante hiérarchique, à chaque étape de l'algorithme il y a deux processus à faire :

1. Chercher une classe à scinder.
2. Choisir un mode d'affectation des objets aux sous-classes.

parmi les algorithme les plus anciens, l'algorithme de *Williams et Lambert*, divise la classe la plus grande en deux classes. *Hubert* a proposé de diviser la classe de plus grand diamètre. Aucun des deux n'a pas justifié son choix de division.

Cette méthode de classification construit sa hiérarchie dans le sens inverse, en commençant par une grande classe contenant tous les objets. A chaque étape, elle divise une classe en deux classes plus petites jusqu'à ce que toutes les classes ne contiennent qu'un seul individu. Ceci veut dire que pour n individus, la hiérarchie est construite en $n-1$ étapes. Dans la première étape, les données sont divisées en deux classes au moyen des dissimilarités. Dans chacune des étape suivantes, la classe avec le diamètre le plus grand se divise de la même façon. Après $n-1$ divisions, tous les individus sont bien séparés. La dissimilarité moyenne entre l'individu x qui appartint à la classe C qui contient n individus et tous les autres individus de la classe C est définie par :

$$d_x = \frac{1}{n-1} \sum_{x \in C, y \neq x} d(x, y)$$

Nous présentons dans la suite quelques algorithms basés sur le principe de **CDH** :

- **Williams et Lambert**
- **TSVQ (*Tree Structured Vector Quantization*)**

Algorithme Williams et Lambert

L'algorithme de *Williams et Lambert* (1959) [WL59] que nous décrivons ici, est particulièrement rudimentaire. Il n'est applicable que sur des variables qualitatives, il sélectionne l'une des variables pour servir de critère d'affectation : tous les individus présentant, pour cette variable, la même modalité sont rangés dans la même classe (si les variables sont à plus de deux modalités le nœud correspondant aura plus de deux branches). La variable retenue est celle qui, dans la classe C à scinder, est la plus corrélée à toutes les autres.
Comme il s'agit de variables qualitatives la corrélation est mesurée par le χ^2 de contingence. On calcule donc les χ^2 de contingence de toutes les variables prises deux à deux, et l'on retient celle pour laquelle la somme de ses χ^2 est maximum.
La méthode de *Williams et Lambert* est bien adaptée au traitement de tableaux présentant un grand nombre d'observations et peu de variables qualitatives. La table des χ^2

de contingence entre variables est alors rapide à obtenir, par comparaison au temps qu'il faudrait pour calculer, par exemple, la matrice de *Jaccard* relative aux individus.

`Début`

`(1) On divise l'ensemble des individus` $\mathcal{I}$ `suivant la variable` V_j `telle que :`
$\delta(V_j, J) = max\{\delta(V_s, J), s = 1, 2, \ldots, p$ `et` $s \neq j\}$
`Sachant que` $\delta(V_j, J) = \sum_{l \neq j}(\chi^2_{jl};\ l = 1, 2, \ldots, p)$
`où` $\chi^2_{jl} = \frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}$
`(2) On obtient une partition de` $\mathcal{I}$ `en deux sous-ensembles`
$\mathcal{I}_1 = \{V \in \mathcal{I}; V_j = 1\}$ `et` $\mathcal{I}_2 = \{V \in \mathcal{I}; V_j = 0\}$ `tels que` $\mathcal{I} = \mathcal{I}_1 \oplus \mathcal{I}_2$
`(3) Chacun de ces sous-ensembles` $\mathcal{I}_1$ `et` $\mathcal{I}_2$ `est divisé à son tour en deux,`
`et l'ensemble` J `est remplacé par` $J_1 = J - V_j$
`(4)` $\mathcal{I}_1$ `est divisé suivant la variable` V_{k1} `telle que :`
$\delta(V_{k1}, J_1) = max\{\delta(V_s, J_1), s \in J_1$ `et` $s \neq k1\}$
`(5)` $\mathcal{I}_2$ `est divisé suivant la variable` V_{k2} `telle que :`
$\delta(V_{k2}, J_1) = max\{\delta(V_s, J_1), s \in J_1$ `et` $s \neq k1\}$
`On arrête de diviser un sous-ensemble lorsque sa taille est inférieure à`
`un nombre fixé au départ`
`(6) Éditer la partition obtenue.`

`FIN`

La facilité à interpréter le résultat est une propriété parmi les avantages de cet algorithme. Par contre, il possède les inconvénients suivants :

- les résultats sont en général grossiers. Cela tient au fait que les groupes d'individus se définissent rarement par leurs réponses strictement identiques à une série de questions mais bien plutôt par un pourcentage élevé de réponses semblables sur l'ensemble des questions.
- Notons encore que les niveaux des nœuds de la hiérarchie ne sont plus définis que par l'ordre dans lequel ils apparaissent et il n'est pas naturel de leur associer un indice montrant la cohésion du groupe d'objets associés à ce nœud..

On pourrait imaginer un programme semblable travaillant sur des variables quantitatives. Il y faudrait ajouter une étape supplémentaire : une fois choisie la variable de scission, il faudrait choisir une valeur-seuil pour cette variable ; en dessous de ce seuil les objets seraient rangés dans l'une des sous-classes, au dessus de ce seuil les objets seraient affectés à l'autre sous-classe. Toutefois une telle procédure présenterait les mêmes avantages et les mêmes inconvénients que celle de *Williams et Lambert.*

Algorithme TSVQ

TSVQ (*Tree Structured Vector Quantization*) a été proposé par *Gersho et Gray* (1992)[GG92]. Cet algorithme utilise l'algorithme K-moyennes (k=2) pour le partitionnement. Utilise la somme des distances par rapport au centroïde (au lieu de la distance moyenne).

Les Avantage et les inconvénients de CDH

* **Avantage :**
  Par rapport à la plus part des algorithmes en classification automatique, l'algorithme de classification descendante hiérarchique ne nécessite pas l'utilisation d'une seuil arbitraire pour la formation des classes qui peut éventuellement mener la recherche d'une partition dans une direction non réaliste.

  Si l'algorithme d'échange ne privilégie que les aspects locaux, il est initialisé avec une partition liée par des relations de filiations avec les partitions précédemment obtenues. Cela donne à l'algorithme un certain aspect global.

* **inconvénients :**
  Les résultats sont en général grossiers, les niveaux des nœuds de la hiérarchie ne sont plus définis que par l'ordre dans lequel ils apparaissent.

Conclusion

Les constructions hiérarchiques par divisions successives ont un aspect séduisant : Elles commencent par le haut de l'arbre, c'est à dire par la partie sur laquelle repose essentiellement l'interprétation. Malheureusement les simplifications drastiques qu'elles exigent, pour maintenir des temps de calcul raisonnables, font que les résultats obtenus sont souvent décevants. Cependant les dichotomies basées sur des variables bien choisies ont l'avantage d'être rapides et de fournir des interprétations aisées. Elles permettent donc de traiter facilement de très grands jeux de données avec peu de variables.

Les avantages des méthodes hiérarchiques sont :

- Flexibilité concernant le niveau du granularité,
- Facilité de manipuler toute forme de similitude ou de distance,
- Applicabilité à tout type d'attribut.

Les inconvénients des méthodes hiérarchiques sont :

– La difficulté de choisir la droite arrêtant des critères,
– La plupart des algorithmes hiérarchiques ne révisent pas des classes (d'intermédiaire) une fois ils sont construits.

3.6 Classification par partition (CPP)

Contrairement aux méthodes de classifications hiérarchiques qui construisent les classes progressivement, les algorithmes de partitionnement construisent directement une partition de l'ensemble d'individus en k classes. Dans le cas général, par définition d'une partition, cela signifie que chaque classe doit contenir au moins un individu, et que chaque individu doit appartenir à une classe unique (cependant, les algorithmes dits "flous" n'imposent pas cette condition). Pour ce faire, étant donné le nombre k de classes requises, ces algorithmes génèrent une partition initiale, puis cherchent à l' "améliorer" en réattribuant les individus d'une classe à l'autre. Il n'est bien entendu pas possible d'énumérer toutes les partitions possibles. Ces algorithmes recherchent donc des maximums locaux en optimisant une fonction objectif qui traduit que les objets doivent être "similaires" au sein d'une même classe, et "dissimilaires" d'une classe à une autre. La méthode la plus classique et qui reste très utilisée est celle des *k-means* [Mac67] et les nombreuses variations qui en découlent. Les classes sont représentées par leur "centroïde", qui correspond à la moyenne de l'ensemble des objets contenus dans la classe. Dans sa version la plus classique, l'algorithme consiste à sélectionner aléatoirement k individus initiales qui représentent les centroïdes initiaux. Un individu est assigné au cluster pour lequel la distance entre l'individu et le centroïde est minimale. Les centroïdes sont alors recalculés et l'on passe à l'itération suivante. La fonction objectif traditionnellement utilisée est :

$$F = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)$$

avec :
– m_i les centroïdes courants,
– x : objet à classer.

L'avantage de cet algorithme est avant tout sa grande simplicité. Sa complexité algorithmique est également intéressante, puisqu'elle est en $O(nkt)$ où t est le nombre d'itérations, avec $t \ll n$ habituellement. De plus, les classes sont facilement interprétables et représentées naturellement par les centroïdes. Cependant, ses inconvénients sont nombreux. Tout d'abord, il nécessite évidemment de pouvoir définir une moyenne entre les données. Deuxièmement, le calcul des moyennes est très sensible aux données aberrantes.

Troisièmement, il est nécessaire de définir le nombre de clusters k, et le résultat de la classification est très dépendant du choix des centroïdes initiaux.

Afin de résoudre notamment le problème de sensibilité aux données aberrantes lors de l'initialisation de la méthode des *k-means*, un autre type d'algorithme a été développé, à savoir les *k-médoïdes*, dont l'algorithme **PAM** (*Partition Around Medoids*) est un exemple typique [KR90]. La principale différence avec les *k-means* se situe au niveau du choix du représentant d'une classe. Dans les *k-médoïdes*, chaque classe est représentée par un de ses membres appelé **médoïde**, et non plus par un **centroïde**, moyenne de l'ensemble des membres. Ainsi, après un choix aléatoire de k médoïdes initiaux, l'algorithme passe en revue tous les couples d'individus tels que l'un est un médoïde et l'autre un non-médoïde, en évaluant si l'échange des deux objets permet d'améliorer la fonction objectif. L'échange qui permet la plus grande amélioration de la fonction objectif est réalisé. Une nouvelle itération a alors lieu. Au final, les différents objets sont affectés à la classe du médoïde le plus proche. Cet algorithme présente l'avantage d'une bonne résistance aux données erronées, puisque les médoïdes ne sont sensibles qu'aux données proches du "centre" de la classe. De plus, il est possible d'utiliser tout type de distance avec les k-médoïdes, puisqu'il n'est pas nécessaire de définir la moyenne des objets. Cependant, l'absence de centroïde qui "résume" les données se fait au détriment de la complexité, puisqu'il apparaît d'après le descriptif précédent de l'algorithme que chaque itération est en $O(k(n-k)^2)$. De plus, comme pour les *k-means*, il est nécessaire de spécifier le nombre de clusters k. Afin de résoudre partiellement le problème des temps élevés de calcul avec les algorithmes de type **PAM**, l'algorithme **CLARANS** (*Clustering large applications based upon randomized search*) [NH94, NH02] est très souvent utilisé. Plutôt que d'essayer tous les échanges entre chaque médoïde et chaque non-médoïde, **CLARANS** choisit aléatoirement un des k médoïdes et un des $(n-k)$ autres objets et regarde si l'échange permet d'améliorer la fonction objectif. Si ce n'est pas le cas au bout d'un nombre d'itérations maximales fixé, l'optimum local est considéré comme atteint. On relance alors l'algorithme en partant d'autres centroïdes initiaux choisis aléatoirement. La meilleure partition parmi les différents essais est alors retenue. La limitation de l'algorithme à un nombre limité d'itérations permet d'obtenir une complexité en $O(n)$.

Ces deux familles de méthodes sont exposées dans les deux paragraphes suivants :

3.6.1 Méthodes k-means

Ces méthodes construisent k classes à partir d'un ensemble de n individus, tout en minimisant la quantité :

$$\sum_{r=1}^{k} \sum_{x_i \in C_r} (x_i - g_r)^2$$

où :

- C_r est la classe numéro r
- x_i est un individu dans une classe
- g_r est le centre de classe C_r.

L'algorithme général de ces méthodes :

```
Données : k le nombre maximum de classes désiré.
Début
    (1) Choisir k individus au hasard (comme centre des classes initiales)
    (2) Affecter chaque individus au centre le plus proche
    (3) Recalculer le centre de chacune de ces classes
    (4) Répéter l'étape (2) et (3) jusqu'à stabilité des centres
    (5) Éditer la partition obtenue.

FIN
```

Nous citons ici deux méthodes connues sur le principe de k-means sont :

- Méthodes de centres mobiles
- Méthodes des nuées dynamiques

Méthodes de centres mobiles

Cette méthode consiste à construire une partition en k classes en sélectionnant k individus comme centres des classes tirés au hasard de l'ensemble d'individus. Après cette sélection, on affecte chaque individu au centre le plus proche en créant k classes, les centres des classes seront remplacer par les centres de gravité et nouveaux classes seront crées par le même principe. L'auteur de cette méthode est `Forgy` en 1965[For65].

```
Données : k le nombre maximum de classes désiré.
```

```
Début
    (1) Choisir k individus au hasard (comme centre des classes initiales)
    (2) Affecter chaque individus au centre le plus proche
    Ce qui donne une partition en k classes P_1 = {C_1, ..., C_k}
    (3) On calcule les centres de gravité des chacune des classes de P_1
    Ce qui donne k nouveaux centres de classes.
    (4) Répéter l'étape (2) et (3) jusqu'à deux itérations successives
    donnent la même partition
    (5) Éditer la partition obtenue.
FIN
```

Généralement la partition obtenue est localement optimale car elle dépend du choix initial des centres. Pour cela les résultats entre deux exécutions de l'algorithme sont significativement variés.
Parmi les méthodes proches de celle des centres mobiles, on peut citer la méthode des nuées dynamiques.

Méthodes des nuées dynamiques

Dans ce cas, le problème posé est la recherche d'une partition en k (k fixé) classes d'un ensemble de n individus. Comme les centres mobiles, les Nuées Dynamiques sont des algorithmes itératifs qui optimisent un critère mathématique.
Nous exposons ici le principe général des algorithmes des nuées dynamiques tel qu'il a été présenté par `Diday` dans [DGLS80].

Soit $\mathcal{I}$ une population d'individus, cette population est représentable sur $\mathbb{R}$ et forme un nuage de n points.

On cherche à constituer une partition en k classes sur $\mathcal{I}$. Chaque classe est représentée par son centre, également appelé noyau, constitué du petit sous-ensemble de la classe qui minimise le critère de dissemblance.

Les deux fonctions de base sur lesquelles repose l'algorithme sont définie de la manière suivante :

- On définit une application f de l'ensemble des classes d'une partition dans un ensemble de représentation (dont les éléments sont appelés les noyaux)
- On définit aussi une application g qui à une famille d'individus de l'ensemble de représentation, associe une partition.

Ainsi, si $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ est un ensemble des classes, f associe à cet ensemble un

ensemble de noyaux $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$ par $f(\mathcal{C}) = \mathcal{N}$ et N_j noyau de C_j.

De même, pour une famille de noyau $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$, l'application g associe à cette famille une partition $\mathcal{C}$ par $\mathcal{C} = g(\mathcal{N})$
où C_j est la classe associée à N_j.

Enfin, on se donne un critère (à valeurs positives) qui permet de mesurer l'adéquation d'une famille de noyaux à une partition :

$$H(\mathcal{C}, \mathcal{N}) = \sum_{j=1}^{k} h(C_j, N_j)$$

Remarquons que H, g, h ne sont pas définis indépendamment :
- Pour $\mathcal{C}$ donné, f associe la famille de noyaux $\mathcal{N}$ qui minimise H.
- Pour $\mathcal{N}$ donné, g associe la partition $\mathcal{C}$ qui minimise H.

Comme les centres mobiles, cet algorithme est une succession d'appels à ces deux fonctions, on notera que le résultat change selon le choix des conditions initiales. Il faut donc exécuter plusieurs fois l'algorithme et comparer les résultats de manière à extraire les classes stables, c'est à dire à dégager des formes fortes.
L'exécution de l'algorithme se termine après un nombre fixe d'itérations. Ce nombre est fixé par l'utilisateur au lancement de l'algorithme.

Données : k le nombre maximum de classes désiré.
i le nombre maximum d'itérations désiré.
    j=1 Début
    (1) Donner une famille de noyaux $\mathcal{N}_0$
    (2) Déduire par g une famille de classes $\mathcal{C}_0$ (une partition)
    (3) Calculer la valeur de $H(\mathcal{C}_0, \mathcal{N}_0)$
    (4) A l'aide de h déterminer une nouvelle famille de noyaux $\mathcal{N}_1$ telle que :
        $H(\mathcal{C}_0, \mathcal{N}_1) \leq H(\mathcal{C}_0, \mathcal{N}_0)$
    (5) Calculer $g(\mathcal{N}_1) = \mathcal{C}_1$ tel que :
        $H(\mathcal{C}_1, \mathcal{N}_1) \leq H(\mathcal{C}_0, \mathcal{N}_1)$
        j=j+1
    jusqu'à j=i
FIN

Avantages des méthodes k-means

Nous pouvons citer quelques avantages des méthodes k-means par :

- Simple,
- Compréhensibles,
- Complexité des calculs en $O(k.n)$,
- Applicables à des données de grande taille,

Inconvénients des méthodes k-means

- Le nombre des classes doit être fixé au départ,
- Ne détectent pas les données bruitées,
- Le résultat dépend du tirage initial des centres des classes,

Ces méthodes donnent la plus part de temps une partition localement optimale, il est donc conseillé d'effectuer plusieurs exécutions et comparer les différentes résultats obtenues. Une suggestion consiste [BFR98] à appliquer la méthode des *k-means* dans une première étape à plusieurs sous-ensembles de données extraits de l'ensemble total, et la meilleure partition obtenue fournit les centres à utiliser au départ de l'algorithme appliqué à l'ensemble total des données.

3.6.2 Méthodes k-medoids

Dans des méthodes de k-medoids une classe est représentée par un de ses individus (médoïde). C'est une méthode itérative combinant la réaffectation des individus dans des classes avec une intervention des médoïdes et des autres individus. C'est une méthode simple parce qu'elle couvre n'importe quel type de variables. Quand des medoids sont choisis, des classes sont définis comme sous-ensembles des individus près des médoïdes les plus proches par rapport à une mesure de distance choisie.

Il est alors plus judicieux de choisir comme centre de groupe un individu présent dans le groupe et non un individu calculé. La médoïde d'un groupe est l'individu possédant la dissimilarité moyenne la plus faible d'avec les autres individus du groupe.

Nous présentons ici quelques versions des méthodes de k-medoids :

- PAM (Partitioning Around Medoids)
- CLARA (Clustering LARge Applications)
- CLARANS (Clustering Large Applications based upon RANdomized Search)

PAM

K-medoids ou "Partition Around Medoids" (PAM) [KR90] :

- Chaque classe est représentée par l'un de ces individus (médoïde)
- Principe
  1. Choisir un ensemble de médoïdes,
  2. Affecter chaque individu au médoïde le plus proche,
  3. Itérativement remplacer chaque médoïde par un autre si cela permet de réduire la distance globale

---

Données : k le nombre maximum de classes désiré.
Début
(1) Sélectionner aléatoirement k représentants $M_1, M_2, \ldots, M_K$
(2) Calculer le coût global CG_{ih} d'une permutation pour chaque paire (M_i, O_h) où M_i est le représentant d'une classe et O_h un autre objet ($CG_{ih} = \sum_j dist(i,h) - dist(i,j)$ c'est le gain en distance globale que l'on va avoir en remplaçant h par i)
(3) Sélectionner la paire (M_i, O_h) pour laquelle CG_{ih} est minimal
(4) Si le coût différentiel CG_{ih} pour la paire sélectionnée est négatif, alors échanger les rôles de M_i et O_h et retourner en (2)
(5) Retourner les classes correspond aux représentants $M_1, M_2, \ldots, M_K$
FIN

L'avantage de l'algorithme **PAM** est qu'il est plus robuste que les méthodes *k-means* en présence de bruit et l'inconvénient vient de la complexité des calculs ($O[k(n-k)^2]$) pour chaque itération ce qui est plus coûteux en cas de k et n assez grand. Ce qui nous conduit à dire que cet algorithme est intéressant et efficace en cas de données de petite taille. D'où la proposition d'autres algorithmes comme **CLARA** pour classer les données de taille moyenne peut être importante.

CLARA

Cet algorithme a été introduit par *Kaufman* et *Rousseuw* en 1990 pour traiter des données de taille moyenne ou grande, elle effectue une recherche locale des représentants en opérant sur plusieurs échantillons de données de taille s extraits de l'échantillon total.

Ensuite l'algorithme de **PAM** sera appliqué à chacun d'entre eux et le résultat obtenu est le meilleur parmi les différents résultats. Les points de l'échantillon total sont alors affectés aux représentants résultants. L'expérience montre qu'en pratique 5 échantillons, de taille $40 + 2k$ points [KR90], suffisent pour obtenir des résultats satisfaisants.

Algorithme CLARA

```
Données : k le nombre maximum de classes désiré.
Début
    Pour i = 1 à 5
        Répéter
    (1) Tirer un échantillon de 40 + 2.k objets
    (2) Appliquer PAM sur l'échantillon
    (3) Calculer la qualité ( Comme dans l'algorithme PAM) de la partition
    engendrée sur l'ensemble des données
    (4) Si la qualité trouvée est la plus faible déjà déterminée,
    (5) Mémoriser les représentants déterminés à l'étape 3.
FIN
```

Points faibles

- Paramètres d'échantillonnage choisis expérimentalement,
- Exploration réduite de l'espace de recherche.

CLARANS

Cette méthode a été introduite par *Ng et Han [1994]*, elle consiste à construire un graphe $G_{n,k}$ (graphe de classification de n individus et de k classes) dont chaque sommet (ou nœud) est représenté par un ensemble de représentants de l'ensemble des individus de taille k : $\{R_{m1}, \ldots, R_{mk}\}$
Deux sommets sont voisins(reliés par une ligne) si leurs ensembles diffèrent par un seul représentant.

Algorithme CLARANS

```
Début
    (1) Fixer les paramètres NUMLOCAL et MAXNEIGHBOR,
    puis Initialiser I à 1 et MINCOST au coût maximal possible
    où :
        NUMLOCAL est le nombre de minimaux locaux,
        MAXNEIGHBOR est le nombre de maximum de voisins d'un
        sommet de graphe G_{n,k},
    (2) Choisir aléatoirement un nœud X du graphe à explorer
    (3) J = 1
    (4) Considérer un voisin Y de X et calculer le coût différentiel TC_{X->Y}
    (5) Si (TC_{X->Y} < 0), alors X = Y et retourner en 3)
    (6) Sinon, J = J + 1
    (7) Si (J < MAXNEIGHBOR), alors retourner en 4)
    (8) Si (MINCOST > COST(X)), alors MINCOST = COST(X) et BEST = X
    (9) I = I + 1. Si (I < NUMLOCAL), retourner en 2)
    (10) Sinon, retourner la solution correspondant à BEST
FIN
```

Points positifs

1. Classes de meilleure qualité par rapport aux méthodes **PAM**, **CLARA** ;

Points négatifs

1. Sensibilité aux paramètres choisis,
2. Complexité de l'ordre de $O(k.n^2)$,
3. Données numériques de petite dimension.

Avantages des méthodes k-medoids

- Bonne résistance aux données erronées,
- Flexibles avec tout type de distance,

Inconvénients des méthodes k-medoids

- Nécessité de spécifier le nombre de clusters k,
- Complexité de chaque itération est en $O(k(n-k)^2)$.

3.6.3 TAXY

Cette méthode a été proposé par *R. Fages*. Elle est basée sur la notion de la dispersion.[EF79],[Fag74]

Avant de donner l'algorithme de cette méthode, nous donnons quelques définitions. Ces définitions sont plus détaillées dans [Fag78],[EF80].

Notions :

- $\mathcal{I}$: l'ensemble des objets à classer supposé fini : $Card(\mathcal{I}) = n$
- p le nombre total des variables observées.
- F un ensemble contenant le produit cartésien $Q_1 \times Q_2 \times \ldots \times Q_p$ où Q_j $1 \leq j \leq p$ désigne l'ensemble des valeurs possibles de la j^{ieme} variable.

Définitions :

Fonction d'ensemble :
Soit C une classe de partie de $\mathcal{I}$, $C \subset \mathcal{P}(\mathcal{I})$. On appelle fonction d'ensemble à valeurs réelles non négatives, toute application de $C \in \mathbb{R}^+$ qui vérifie les propriétés suivantes :

a) Additivité, sur-additivité, sous-additivité
Une fonction d'ensemble F à valeurs réelles non négatives définie sur C est dite additive sur C si :

$$\forall A \in C, \forall B \in C, \quad \text{tels que} \quad A \cap B = \phi \quad \text{et} \quad A \cup B \subset C,$$

on a :

(1) $F(A \cup B) = F(A) + F(B)$
La sur-additivité ("resp. sous-additivité") se traduit dans (1) par le signe "$\geq$" ("resp. "$\leq$")

b) σ-additivité, σ-sur-additivité, σ-sous-additivité
Une fonction d'ensemble F est σ-additive dans C si pour toute la famille (A_i) ; $i \in I \subset \mathbb{N}$ d'élément de C disjoints 2 à 2 dont le réunion est encore dans C ,on a :

(2) $F(\bigcup_{i\in I} A_i) = \sum_{i\in I} F(A_i)$

La σ-sur-additivité ("resp. σ-sous-additivité") se traduit dans (2) par le signe "$\geq$" ("resp. "$\leq$")

c) Monotonie dans l'inclusion

Une fonction d'ensemble F à valeurs réelles est monotone croissante ("resp. décroissante) si, pour tout couple A, B d'éléments de C, on a :

$$A \subset B \Longrightarrow F(A) \leq F(B)$$

("resp. $\geq$")

d) Continuité

Soit E_n une suite monotone d'éléments de $C(E_1 \subset E_2 \subset \ldots$ ou $E_1 \supset E_2 \supset \ldots)$ telle que $\lim_{n\to\infty} E_n \in C$, alors on dira que F est continue si :

$$\lim_{n\to\infty} F(E_n) = F(\lim_{n\to\infty} E_n)$$

Dispersion

Une dispersion est une fonction d'ensemble D à valeurs réelles non négatives, définie sur une classe C de parties de $\mathcal{I}$ contenant l'ensemble vide ϕ, vérifiant les axiomes suivants :

1. $D(\phi) = 0$
2. Si A est un atome alors $D(A) = 0$
3. D est sur-additive "ou σ-sur-additive"
   $\forall A \subset \mathcal{I}, \forall B \subset \mathcal{I}$ tels que $A \cap B = \phi$, nous avons : $D(A \bigcup B \geq D(A) + D(B)$

Dispersion associée à une partition

Soit D une dispersion définie sur une classe C de parties de $\mathcal{I}$

Soit $P = \{C_1, \ldots, C_q\}$ une partition de $\mathcal{I}$ telle que pour tout indice k on ait $C_k \in C$

La quantité désignée par $D[P]$ et qui est définie par :

$$D[P] = \sum_{k=1}^{q} D(C_k)$$

est appelée dispersion de la partition P.

Exemple de dispersion :

- Si on définie ρ comme dissimilarité sur $\mathcal{I}$ on peut définir la dispersion suivante :

$$D(A) = min[\sum_{a \in A} \rho(a, x) | x \in \mathcal{I}]$$

- $\mathcal{I}$ désigne eu sous-ensemble des fonctions continues sur un intervalle $[a, b]$

$$D(A) = Card(A) \int_a^b (max[f(t)|f \in A] - min[f(t)|f \in A])dt$$

- $\mathcal{I} \subset \mathbb{R}^p, d$: distance euclidienne.

$$D(A) = Card(A) max[d(a, b)|a, b \in A]$$

- $\|.\|$ est la norme euclidienne, $y = \mathbb{R}^2$ est l'espace de représentation de $\mathcal{I}$, Soit : $A \subset \mathcal{I}, X_i = X(i)$

$$D(A) = min \sum_{y \in \mathbb{R}^2, i \in A} \| X_i - y \|^2$$

- $Y = \{0, 1\}^p$

$$D(A) = Card(A).[\prod_{j=1}^{p} f_{A_j}(1 - f_{A_j})]^{1/p}$$

  où f_{A_j} est la fréquence du caractère j dans la classe A.

- $y = \mathbb{R}^p$

$$D(A) = Card(A).[detV_A]^{1/p}$$

  où V_A est la matrice des variance-covariances du nuage X(A).

Définition 3.9. ***Partition optimale***
Une partition $P_q = \{C_1, \ldots, C_q\}$ en q classes est optimale pour la dispersion D ssi $D[P_q]$ est minimum absolu sur l'ensemble des partitions à nombre fixé q de classes.

Définition 3.10. ***Partition localement optimale***
Une partition P_q est localement optimale ssi :

$\forall i, j \in \{1, 2, \ldots, q\}, \forall x \in C_k$ *on a :*

$$D(C_i) + D(C_j) \leq D(C_i - \{x\}) + D(C_j + \{x\})$$

Définition 3.11. ***Ecart entre deux classes disjointes :***
On appelle écart entre deux classes disjointes C_i et C_j, la quantité $\delta(C_i, C_j)$ définie par :

$$\forall C_i, C_j \in \mathcal{P}(\mathcal{I}), C_i \cap C_j = 0, \qquad \delta(C_i, C_j) = D(C_i \cup C_j) - D(C_i) - D(C_j)$$

Définition 3.12. ***Elémnt lâche d'une classe.***
On appelle élément lâche de A tout élément $\bar{\bar{x}}_A$ tel que :

$$D(A - \{\bar{\bar{x}}_A\}) = min[D(A - \{x\}) \,|\, x \in A]$$

(c'est l'élément qui contribue le plus à la dispersion)

Définition 3.13. ***Élément centré d'une classe :***
On appelle élément centré de A tout élément $\bar{x}_A$ tel que :

$$D(A - \{\bar{x}_A\}) = max[D(A - \{x\}) \,|\, x \in A]$$

(c'est l'élément qui contribue le moins à la dispersion)

Définition 3.14. ***Attraction d'une classe sur un élément :***
Soit $A, B \in \mathcal{P}$, $A \cap B = \phi$ et $x \in A$,

par définition, l'attraction de A sur x est définie par :

$$att(A, x) = 0$$

l'attraction de B sur x est définie par :

$$att(B, x) = D(A) + D(B) - D(A - \{x\}) - D(b \cup \{x\})$$

notons que, si $att(B, x) > 0$ la dispersion de la partition diminue quand on passe x dans B.

Algorithme de TAXY

`Données :` q_{max} `le nombre maximum de classes désiré.`

```
Début
    (1) Création d'une classe unique q = 1 𝒫 = {ℐ}

    (2) Pour les q classes actuelles,

        rechercher celle P_k qui a la dispersion maximale :

            D(P_k) = max[D(P_i) | P_i ∈ 𝒫]

    (3) q := q + 1
        Création de P_q = {x̄̄} élément lâche de P_k

    (4) Etape de transfert
        répéter
            change :=false ;
            pour i := 1 à n faire
                chercher P_{K_0} ∈ 𝒫 tel que att(P_{k_0}, a_i) = max[att(P_k, a_i) | P_k ∈ 𝒫]
                si att(P_{k_0}, a_i) > 0
                    alors
                        P_{α(i)} = P_{α(i)} − {a_i}
                        P_{k_0} = P_{k_0} ∪ {a_i}
                        change :=vrai
                FinSi
            Fin Pour
        Jusq'à NOT change.
        Editer la partition obtenue.

    (5) Si q < k alors aller en (2) sinon FIN

FIN
```

3.7 Classification par la densité

Les méthodes de la classification basées sur la densité sont des méthodes hiérarchiques, dans lesquelles les classes sont considérées comme des régions en haute densité qui sont

séparées par des régions en faible densité. La densité est représentée par le nombre d'individus de l'ensemble des données. C'est pourquoi ces méthodes sont capables de chercher des classes de forme arbitraire. Elles ne travaillent que dans un espace métrique.

Le principe de ces méthodes est de caractériser une classe comme étant une zone où le nombre de données initiales est plus important qu'ailleurs.

Il y a deux approches dans ce type de méthodes :
- Approche basée sur la connexité de densité
  DBSCAN, OPTICS, DBCLASD
- Approche basée sur la fonction de densité
  DENCLUE

3.7.1 Approche basée sur la connexité de densité

DBSCAN

L'algorithmes **DBSCAN** (*Density Based Spatial Clustering of Applications with Noise*) [EKSX96], et ses dérives tels que **OPTICS** [ABKS99] ou **DBCLASD** [XEKS98] sont basés sur l'idée de définir la notion de voisinage de rayon ε d'un point : tous les points situés à une distance de ce point inférieure a ε appartiennent au voisinage. Pour qu'une classe soit identifiée, il est nécessaire qu'un voisinage contienne plus de ($MinPoints$) autres points. Les clusters sont alors agrandis en associant de proche en proche les points de voisinages qui respectent les conditions précédentes.

DBSCAN utilise la notion *connecté-densité* pour former des classes :
- Un point est dit *directement accessible-densité* ($\varepsilon - MinPoints$) d'un autre point s'il se trouve dans le voisinage ($\varepsilon - MinPoints$) de ce point.
- Un point est dit *accessible-densité* ($\varepsilon - MinPoints$) d'un autre point s'il y a une chaîne de points entre eux dont tous les 2 points successifs sont directement accessible-densité ($\varepsilon - MinPoints$).
- Un point est dit *connecté-densité* ($\varepsilon - MinPoints$) d'un autre point s'il y a un point duquel tous les deux points sont accessibles-densité.

Une classe avec ε et $MinPoints$ prédéfinis est définie comme un ensemble non vide d'individus qui satisfait 2 conditions :
- La condition de *connectivité*, i.e. tous les points de la classe doivent être connectés-densité,

– La condition de *maximum*, i.e. tous les points qui se trouvent dans le voisinage ($\varepsilon - MinPoints$) d'un point de la classe doivent appartenir à cette classe.

Le bruit est défini comme un ensemble des points qui n'appartiennent à aucune classe.

Il y a deux points différents qui sont pris en compte dans la classification :

– Un point de noyau : C'est le point qui a un voisinage ($\varepsilon - MinPoints$).
– Un point non noyau : C'est celui qui n'a pas un tel voisinage.

Un point non noyau peut être un point de frontière ou un bruit.

L'algorithme commence en prenant en compte d'un point arbitraire et cherche tous les objets accessibles densité. S'il est point de noyau, alors cette phase forme une classe. S'il est un objet de frontière et qu'il n'y a aucun point qui est accessible densité depuis lui, alors c'est un bruit, l'algorithme passe à un autre objet.

Algorithme

Données : $\varepsilon, MinPoints$
Début
    Répéter
    (1) Prendre un point $x \in \mathcal{I}$ aléatoirement
    (2) Mettre dans une classe C tous les point *accessible-densité* à partir de x
    (3) Si le point x est noyau, alors C est une classes ;
    (4) Si x est un point non noyau (frontière) :
        Passer à un autre point et retourner en (2)
    Jusqu'à passer tous les points
FIN

Ce type d'algorithme peut a priori gérer tout type de données. Cependant le calcul d'un voisinage de taille ε pose problème au niveau de la complexité. C'est uniquement dans le cas de données spatiales de faible dimension que l'on peut calculer ces voisinages en $O(log\, n)$. Ainsi, la complexité de l'algorithme est en $O(n\, log\, n)$ pour de telles données, et en $O(n^2)$ dans le cas général.

Avantages des méthodes DBSCAN

Cet algorithme présente l'intérêt de trouver lui-même une évaluation du nombre de classes. Celles-ci peuvent avoir des formes arbitraires.
L'algorithme permet également de bien gérer les données aberrantes, qui ne sont pas affectées aux clusters détectes.

Inconvénients des méthodes DBSCAN

Il requiert des paramètres ε et $(MinPoints)$, et l'expérience montre que les résultats obtenus sont très sensibles aux choix de ces paramètres.

OPTICS

L'algorithme **OPTICS** *(Ordering Points To Identify Clustering Structure)* a été proposé par *(Ankerst et al)* en 1999 [ABKS99]. C'est une extension de l'algorithme **DBSCAN**. L'idée générale est d'identifier les régions potentielles de début de classe et de fin de classe et ensuite, de combiner ces régions pour former une hiérarchie. L'algorithme ordonne les points pour des valeurs de voisinage croissantes, et permet une exploration interactive de la hiérarchie ainsi produite.

3.7.2 Approche basée sur la fonction de densité

DENCLUE

L'algorithme **DENCLUE** (*Density-based clustering*) a été proposé par *Hinneburg et al* en 1998 [HK98]. Il généralise l'approche de **DBSCAN** puisque celui-ci en est un cas particulier. **DENCLUE** modélise l'influence de chaque point sur son voisinage par une fonction d'influence (dépendant de la distance entre les objets et au moins d'un paramètre σ réglant l'échelle du voisinage influence).
Une fonction d'influence sert à mesurer l'impact d'un point dans son voisinage. Une fonction d'influence très connue est la fonction *gaussienne* :

$$f_{Gauss}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \quad ; \ \sigma^2 = \frac{1}{N-1}\sum_i (x_i - \bar{x})^2$$

Une fonction de densité est alors définie comme étant la somme de toutes les fonctions

d'influence de tous les points :

$$f^D_{Gauss}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

Les classes apparaissent comme étant les zones de l'espace ou la fonction de densité est supérieure a un seuil σ. Afin de permettre une gestion d'un grand nombre de données, l'algorithme utilise dans une première étape une quantification de l'espace en hypercubes de taille fixe (cf. plus bas les algorithmes basés sur une quantification par grille) qui servent à accéder rapidement à des zones voisines de l'espace. L'utilisation de cette quantification permet d'obtenir une complexité linéaire en $O(nlog\,n)$. Cependant, il est nécessaire de pouvoir notamment calculer des moyennes des données.

Avantages des méthodes DENCLUE

Cet algorithme permet de trouver des classes aux formes arbitraires, sans fixer a priori le nombre de classes cherché. Il est également robuste au bruit. Plus rapide que **DBSCAN** (facteur d'approximatif 45)

Inconvénients des méthodes DENCLUE

Cependant, il nécessite un choix adapté des paramètres σ et ε.

3.8 Classification basée sur la quantification par grille

L'idée de ces méthodes est qu'on divise l'espace de données en un nombre fini de cellules formant une grille. Ce type d'algorithme est conçu pour des données spatiales. Une cellule peut être un cube, une région, un hyper rectangle.
En fait, avec une telle représentation des données, au lieu de faire la classification dans l'espace de données, on la fait dans l'espace spatial en utilisant des informations statistiques des points dans la cellule. Les méthodes de ce type sont hiérarchiques ou de partitionnement. Les algorithmes les plus connus sont **STING, CLIQUE, WaveCluster.**

STING

L'algorithme **STING** (***ST****atistical* **IN***formation* ***G****rid*) [WYM97] utilise ainsi une grille multirésolution. Les données dans chaque cellule sont résumées par le nombre d'ob-

jets dans la grille et par leur moyenne.

Chaque cellule est récursivement découpée en 4 sous-cellules. On commence en considérant l'espace entier comme une cellule ancêtre de la hiérarchie et on s'arrête si l'on atteint un critère de terminaison : (le nombre des points dans une cellule est inférieur à un seuil).

Chaque cellule possède certaines informations nécessaires pour l'affectation d'un point :
- La moyenne de toutes les valeurs dans la cellule,
- La variance de toutes les valeurs d'attribut dans la cellule,
- La valeur d'attribut minimum, maximum dans la cellule
- Le type de distribution de la cellule (normale, uniforme, exponentielle ou simplement une énumération).

Quand on remonte dans la hiérarchie, ces informations statistiques sont calculées à l'aide de celles du niveau inférieur.
STING est extensible car il a besoin d'une seule passe sur les données et un nouvel objet qui arrive peut être facilement inséré dans la grille en visitant les cellules appropriées à chaque niveau de la hiérarchie.

La classification est effectuée dans ces cellules de feuille au lieu de toutes les données, le coût de calcul est donc réduit ($O(k)$ au lieu de $O(N)$, k étant le nombre de cellules et N est le nombre de points de données).

Cet algorithme gère également bien les données aberrantes, puisque seules les cellules très peuplées sont importantes. Cependant, il nécessite la définition de la moyenne des données. Il est dépendant du paramètre fixant la résolution de la grille, et obtient forcément des classes "rectangulaires".

WaveCluster

WaveCluster a été développé par *Sheikholeslami et al* en 2000 [SCZ00]. C'est un algorithme qui est conçu pour travailler sur des données spatiales. Dans l'algorithme, on considère les données spatiales comme les signaux multidimensionnels et on applique une technique de traitement de signal qui s'appelle la transformation wavelet pour transformer l'espace de données dans un espace de fréquences. L'idée est basée sur le fait que les parties de haute fréquence du signal correspondent aux régions de l'espace spatial de données où il y a un changement brusque dans la distribution des objets. Ce sont les frontières des

classes. Au contraire, les parties de basse fréquence du signal correspondent aux régions de l'espace de données où les objets sont concentrés. (les classes).

Tout d'abord, l'algorithme quantifie l'espace de données (divise l'espace en cellules) et affecte les objets aux cellules. Ensuite, il applique la transformation wavelet pour l'espace quantifié afin d'obtenir un espace transformé. Puis il cherche les composantes connectées (les classes) dans les sous bandes (l'espace transformé est un espace de fréquence). Deux composantes sont dites connectées si le nombre d'objets de chacune dans l'espace transformé est supérieur à un seuil et si les 2 composants sont $\varepsilon - voisines$ (la distance entre elles est inférieure à ε. Enfin, il étiquette les cellules et réaffecte les objets à leurs classes correspondantes.

WaveCluster est capable de traiter efficacement les gros jeux de données spatiales donc en basse dimension. Il peut découvrir des classes de forme arbitraire et bien traiter le bruit. En plus, grâce au traitement du signal appliqué, **WaveCluster** nous donne la multi résolution, i.e. il peut donner des classes de différents niveaux du détail. La complexité de la phase de quantification est $O(N)$ et celle de la phase de transformation wavelet est au plus de $O(4/3k)$ tel que k est le nombre de cellules.

CLIQUE

CLIQUE *(**CL**ustering **I**n **QUE**st)* a été développé par *Agrawal et al* en 1998 [AGGR98]. Dans cet algorithme au lieu de construire les classes dans l'espace original comme les autres algorithmes, **CLIQUE** le fait dans des sous-espaces en dimension la plus grande possible. L'idée est basée sur le fait que si une collection de points S est une classe dans un espace de k dimensions, alors S est ainsi une partie d'une classe dans n'importe quelle projection en $k-1$ dimensions de cet espace.

L'algorithme travaille en plusieurs niveaux. Tout d'abord, il détermine toutes les unités denses en une dimension en balayant une fois les données. Après avoir déterminé les unités denses en $(k-1)$ dimensions, il détermine les unités denses candidates en k dimensions en utilisant une procédure qui génère les unités candidates.

La procédure de génération utilise l'ensemble de toutes les unités denses en $k-1$ dimensions D_{k-1}. Elle joint D_{k-1} avec elle-même sous une condition où les unités qui sont jointes partagent les premières $k-2$ dimensions pour obtenir un ensemble de candidates C_k. On supprimera les unités denses de C_k qui ont une projection en k dimensions qui ne

sont pas incluses dans C_{k-1}. L'algorithme se termine lorsqu'il n'y a plus de candidatures générées.

CLIQUE est une solution pour la classification de données en haute dimension. Il est insensible à l'ordre des données. Cependant, sa complexité est exponentielle. Supposons que k est le nombre le plus haut de dimensions du sous-espace auquel l'algorithme se termine, m est la cardinal de l'ensemble des données, l'algorithme doit accéder au jeu de données k fois. La complexité est $O(c^k + mk)$ où c est une constante.

3.9 Autres Méthodes

COBWEB

L'algorithme **COBWEB** a été proposé par *Fisher* en 1987 [Fis87a]. Il construit dynamiquement un dendrogramme en passant en revue les individus un à un. Au cours de la construction de dendrogramme, chaque nouvel individu parcourt l'arbre qui est au fur et à mesure mis à jour. **COBWEB** est rapide avec une complexité de calculs en $O(t.n)$ où t est une constante dépendant des caractéristiques de l'arbre.

SOM

Les réseaux neuronaux ont développé leur propre méthodes de classification, les **SOM** *(Self Organizing Maps)* [Koh88, Koh82, Koh00]. Le principe s'appuie sur la théorie des réseaux à compétition, c'est-à-dire sur l'établissement d'une liaison de compétition entre les neurones autres que ceux d'entrée. En pratique, cela signifie que des liens inhibiteurs relient les neurones. Lors de la phase d'apprentissage, le réseau spécialise ses neurones en reconnaissance de catégories d'entrées, ce qui est en effet une façon d'apprendre des classes. Une classe est définie comme l'ensemble des exemples reconnus par un neurone de sortie d'un réseau à compétition.
Les SOM sont composés, comme tous les réseaux à compétition, d'une couche d'entrée et d'une couche de compétition. La couche de compétition est structurée par une distance permettant de définir une notion de voisinage. La compétition, au départ, ne se fait alors plus entre un neurone et les autres, mais entre un voisinage et le reste de la carte. Cette distance est en fait une fonction du temps, de sorte qu'on retrouve une compétition par neurone en fin d'apprentissage. L'avantage de cette approche par voisinages variables est de séparer effectivement les exemples très différents les uns des autres. Les dernières classes

sont un raffinement des celles obtenues au départ.

ANTCLASS

D'autres méthodes ont vu le jour, dont l'hypothèse sous-jacente, bien que non clairement formulée, les rattache à une approche de type non paramétrique. Par exemple, dans l'approche multi-agents de **ANTCLASS** [MVS99], des microprogrammes autonomes nommés *fourmis* agissent localement en "déplaçant" des données de manière à faire des empilements. Le comportement émergent de la colonie fournit des paquets de données compacts, interprétés comme des clusters. Si l'idée est séduisante, il n'a pas été prouvé que la mise en oeuvre effective de cette approche par application à de gros volumes de données soit un succès, comparativement aux autres algorithmes du domaine.

SUPER-PARAMAGNETIC-CLUSTERING (SPC)

Un autre algorithme d'inspiration physique fonctionne sur un principe de *recuit simulé*, mais cette fois appliqué à une fonction objectif favorisant la formation de clusters de forme quelconque : le **SUPER-PARAMAGNETIC-CLUSTERING** [BWD97]. Cet algorithme fait varier un paramètre de température, et pour chacune de ses valeurs, calcule les zones denses dans les données selon cette fonction objectif. Le résultat est une séquence de partitions imbriquées, donc l'équivalent d'une hiérarchie.

GGA

Les algorithmes génétiques ont aussi été appliqués à la classification [RB79, GKR00, JB91], comme par exemple pour le système **GGA** (*Genetically Guided Algorithm*) de [BBHB94, HÖB99]. Le principe des approches génétiques est d'encoder une classification sous forme de chromosomes virtuels [JMF99, JD88] pour quelques exemples de représentation). Ceux-ci sont soumis à des opérations de mutations et de cross-over, l'idée étant de faire évoluer la population de chromosomes-classifications et de tester leur adéquation aux données à l'aide d'une fitness fonction basée sur l'erreur quadratique. A chaque génération, seuls les meilleurs au sens de cette fonction servent de base pour la génération suivante, selon le principe darwinien de sélection naturelle. Ces méthodes sont extrêmement sensibles aux paramètres tels que la taille de la population, les opérations de modification des chromosomes, et bien entendu la fitness fonction, et ont en général un coût en calculs très élevé.

3.10 Comparaison de classification

Indice de Rand

On peut examiner visuellement en comparant les dendrogrammes pour différentes solutions. Cette technique est ennuyeuse pour un nombre d'individus assez grand. Un indice ou une test statistique pour comparer les différentes classifications peut résoudre ce problème.

Cet indice compare deux classifications (même si les nombres des classes ne sont pas les mêmes) [Ran71]. Il est donné par le formule :

$$Rand(CL_i, CL_j) = \frac{2}{n(n-1)} \sum_{x} \sum_{y>x} r_{xy}$$

Où r_{xy} égale à 1 si x et y sont dans la même classe pour les deux classifications (CL_i, CL_j). si l'indice de **Rand** est égal à 1, donc nous avons un accord parfait entre les deux classifications. Une valeur de 0.7 est considérée comme suffisante pour dire que les deux classifications sont assez en accord [Dre86, FS92]. En 1984 (Morey et Agresti) [MA84] ont proposé une modification à l'indice de **Rand**. En 1985 (Hubert et Arabie) (cités dans Gordon [Gor98]) ont modifié l'indice de sorte que son maximum soit 1 et sa valeur prévue soit zéro, si les classifications sont choisies aléatoirement.

Critères pour une bonne classification

L'objectif principal des techniques de classification est de trouver une partition où les objets d'une classe devraient être semblables (entre eux), les objets de différents classes devraient être différents. Une bonne classification devrait accomplir différents critères [Gor98, Eve93]

- Validité interne :
  1. Chaque classe d'une partition doit être homogène : Les objets qui appartiennent à la même classe doivent être semblables.
  2. Les classes doivent être isolés entre eux : Les objets de différents classes doivent être différents.
  3. La classification doit s'adapter aux données : La classification doivent pouvoir expliquer la variation des données.
- interprétabilité :

1. Les classes doivent avoir une interprétation substantive : Il est possible de donner des noms aux classes. Dans le meilleur des cas, ces noms doivent correspondre aux types déduits d'une certaine théorie.

– Stabilité :

1. Les classes doivent être stables : Les petites modifications dans les données et dans les méthodes ne doivent pas changer les résultats.

– Validité externe :

1. Les classes doivent être valides (validité externe) : Les classes doivent se corréler avec les variables externes qui sont connues pour être corrélées avec la classification et qui ne sont pas employées pour grouper.

– Validité relative :

1. La classification doit être meilleure que d'autres classifications.

– D'autres critères :

1. Parfois la taille et le nombre de classes sont employés en tant que critères additionnels : Le nombre de classes doit être aussi petit que possible. La taille des classes ne doit pas être trop petite.

Les critères ne sont pas complètement indépendants. Mais Parfois, on s'intéresse seulement aux quelques critères. Par exemple, les critères de la validité interne et l'interprétabilité. Une meilleurs classification est celle qui pourra accomplir les critères de la validité interne et de la validité externe. Cependant, toutes les classifications doivent accomplir la critère de l'interprétabilité, elles doivent être essentiellement interprétabes.

Le nombre des classes

Il n'y a pas des méthodes complètement satisfaisantes pour déterminer le nombre des classes de l'ensemble des individus et pour tout type de classification [Eve79, Eve93]

Dans les méthodes de classification hiérarchiques, couper l'arbre par une droit horizontale donne une partition de l'ensemble des individus. Donc on peut fixer le nombre des classes désirées selon l'endroit où on fait notre coupe.

Dans les méthodes de classification par partition, le nombre des classes est fixé à l'avance par l'utilisateur.

Il n'y a pas de méthodes analytiques pour déterminer le nombre de classes. Nous allons citer quelques indices souvent cités dans les littérateurs pour déterminer le nombre optimal des classes.

On part dans tous les cas de l'hypothèse où l'on dispose des résultats de classification en k classes pour $k \in (1, ..., M)$. Une premiere grande classe de méthodes consiste a utiliser une fonction objectif, qui atteint son maximum pour le nombre optimal de classes, en cherchant typiquement a maximiser les distances interclasses tout en minimisant les distances intraclasses. Il suffit donc de calculer cette fonction objectif pour chaque k et de choisir le k qui donne le résultat le plus élevé. Les fonctions objectifs dépendent parfois uniquement des distances intraclasses [Har75, KL88], mais aussi des distances interclasses [KR90].

Un autre type de méthodes consiste à chercher également le maximum d'une fonction objectif, mais avec une approche probabiliste différente : il s'agit d'évaluer si l'ajout de nouveaux paramètres au modèle (c'est-à-dire utiliser un k plus grand) conduit à un meilleur modèle ou non. Ces techniques cherchent donc à évaluer un compromis entre l'adéquation du modèle aux données et la complexité du modèle. Différents critères ont été proposés, tels que le critère de la longueur de description minimum ("MDL", Rissanen en 1978 [Ris78]), la longueur du message minimum ("MML", Wallace et Freeman en 1987 [WF87, WKD96]), le critère de l'information bayésienne ("BIC", Schwarz 1978 [Sch78]), ou d'autres critères d'évaluation d'hypothèses statistiques (par exemple, Hamerly et Elkan 2004 [HE03]).

Un troisième type de méthodes correspond aux méthodes par rééchantillonage (Tibshirani et al. 2001 [TWH00]), Dudoit and Fridlyand 2002 [DFS02]). Elles consistent à comparer les classifications obtenues sur les données réelles à des classifications "de référence", correspondant à l'absence de classes. Pour cela, des données sont simulées à partir de distributions uniformes, et des classifications sont réalisées sur ces échantillons. Le nombre de classes k pour lequel l'écart entre la classification réelle et la classification de référence est le plus grand est le nombre retenu de classes. Ces méthodes présentent l'inconvénient majeur de nécessiter de multiplier les rééchantillonages afin d'obtenir des classifications de référence fiables. Cela n'est possible que lorsque l'algorithme de classification n'est pas trop consommateur en temps de calcul en lui-même.

Les mesures de homogénéité

Les mesures de homogénéité supposent que la moyenne des dissimilarités dans les classes (d_{in}) est plus petite que la moyenne des dissimilarités entre les classes (d_{out})pour une bonne classification. Différentes approches ont été proposé pour calculer ces moyennes [Kla83].
Par exemple, les dissimilarités à l'intérieur et entre les classes sont calculées dans cette approche comme :

$$d_{in} = \sum_{k} d(k)_{in}/K$$

$$d_{out} = \sum_{k} \sum_{k*>k} d(k, k*)_{out}/(K(K-1)/2)$$

$$d(k)_{in} = \sum_{g \in k} \sum_{\substack{g* \in k \\ g*>g}} d_{g,g*}/(n_k(n_k-1)/2)$$

et

$$d(k, k*)_{out} = \sum_{g \in k} \sum_{g* \in k*} d_{g,g*}/(n_k n_{k*})$$

Où K est le nombre des classes. le n_k est le nombre des individus dans la classe K.
La teste de la homogénéité est la différence :

$$g = d_{out} - d_{in}$$

ou

$$g = d_{in}/d_{out}$$

La première expression a l'avantage qu'il est facile de construire une teste statistique. L'inégalité standard de la distribution normale ou de *Chebyschev* peut être employée pour calculer un niveau de signification évaluant si les deux dissimilarités sont égales ou une plus grande que l'autre dissimilarité [Kla83].

3.11 Conclusion

Plusieurs méthodes sont proposées pour le problème général de la classification. Ils diffèrent par les mesures de proximité qu'ils utilisent, la nature des données qu'ils traitent et le objectifs finale de la classification. Chacune des ces méthodes possède ses points

Nom	Type d'algorithme	Paramètres d'entrée	Caractéristiques de l'algorithme
k-means	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité : $O(Ikn)$ Il existe certaines versions qui diffèrent par la mise à jour des céntroïds
PAM	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité : $O(Ik(n-k)^2)$
CLARA	Partition	Nombre de classes	Ne traite pas les aberrants. Complexité : $O(ks^2 + k(nk))$
CLARANS	Partition	Nombre de classes. Maximum nombre de voisins	Ne traite pas les aberrants. Complexité : $O(kn^2)$ Combinaison entre PAM et CLARA donc donner une meilleure qualité de classes.
CURE	Hiérarchique	Nombre de classes, Nombre de représentatifs	Traite les aberrants. Complexité : $O(n^2 logn)$ Utiliser le résumé de données
BIRCH	Hiérarchique	Nombre d'embranchement, Seuil de compacité	Traite les aberrants Complexité : $O(n)$ Utiliser le résumé de données. Trois versions pour données numériques
DBSCAN	Basé sur densité	Rayon d'une classe, nombre minimum de points dans une classe	Traite les aberrants. Complexité : $O(n \, log \, n)$
DENCLUE	Basé sur densité	Rayon d'une classe, nombre minimum d'objets	Traite les aberrants. Complexité : $O(n \, log \, n)$
OPTICS	Basé sur densité	Rayon d'une classe, nombre minimum, maximum d'une classe,Nombre minimum d'objets	Traite les aberrants. Complexité : $O(n \, log \, n)$
STING	Basé sur grille	Nombre de cellules au niveau le plus bas, Nombre d'objets dans une cellule	Traite les aberrants, Complexité $O(n)$
WaveCluster	Basé sur grille	Nombre de cellules pour chaque dimension, Wavelet, Nombre d'applications de transformation	Traite les aberrants, Complexité $O(n)$
CLIQUE	Basé sur grille	Taille de la grille, Nombre minimum de points dans une cellule	Traite les aberrants Complexité $O(c^k + kn)$ Classification faite dans sous-espaces

Tab. 3.4 – Sommaire des algorithmes de classification

forts et ses points faibles. Les méthodes hiérarchiques ascendantes sont utilisées en cas des données de petite taille car la complexité est très élevée. Si au contraire, des problèmes de temps d'exécution se posent, alors c'est les méthodes des k-means qui sont utilisées. Enfin, si l'objectif est de fournir des classes de forme quelconque, alors ce sont les méthodes basées sur la densité ou sur des grilles qui sont utilisées.
C'est donc, le choix d'une méthode approprié dépend fortement de l'application, la nature des données et les ressources disponibles. Une analyse attentive des données aide à bien choisir le meilleur algorithme. Il n'existe pas un algorithme qui peut répondre à toutes les demandes.
Malgré le nombre important de méthodes existantes, plusieurs problématiques restent encore ouvertes dans le cadre de la classification. Un problème très sauvent rencontré concerne la difficulté de fixer les paramètres en entrée des méthodes par l'utilisateur. De même, la présentation des résultats sous une forme facilement intérprétable par l'utilisateur est rarement abordée malgré son intérêt. La complexité des méthodes mises en œuvre est aussi problématique dans certains cas. La prise en compte du contexte dans lequel les classes sont créés constitue un sujet d'étude récent. Enfin, une problématique ouverte importante dans le cadre de la classification concerne l'évaluation des résultats et la comparaison des différentes méthodes.
Dans la suite de cette thèse, nous présentons quelques approches concernes la classification. La partie II présente ainsi nos travaux dans la mise en œuvre de nouvelles méthodes de classification, minimisant le nombre de paramètres requis, fournissant en sortie un résultat compréhensible et ayant une complexité

Deuxième partie

Quelques Approches de Classification des Données

Chapitre 4

Approche basée sur l'agrégation d'opinions

Sommaire

4.1 Introduction **76**

4.2 Présentation du problème **77**

4.2.1 Etude du critère d'optimisation 78

4.2.2 Calcule du nombre de désaccords 80

4.3 Formulation du problème **81**

4.4 Résolution du problème **82**

4.4.1 Application 84

4.5 Extension de la méthode **87**

4.6 Cohérence et ressemblance entre groupes **89**

4.7 Structuration de l'ensemble des variables **92**

4.7.1 Formulation classique du problème 92

4.7.2 Une méthode heuristique pour la résolution du problème 94

4.1 Introduction

Dans ce chapitre, nous présentons une méthode de classification des données basée sur le principe d'agrégation d'opinions.
Ce problème a été posé par *Régnier* [Reg65] repris ensuite par *Marcotorchino et al.* [MM82], *Amorim et al.* [DABR92] et *Barthélemy et al.* [BL95].
La méthode proposée consiste à associer à chaque variable une fonction de classement qui va jouer le rôle d'un juge. Ce dernier va classer les individus selon ses propres critères.
A partir de l'ensemble de classement de toutes les variables, on cherche à construire sur l'ensemble des individus un classement collectif qui soit la meilleure agrégation possible.
Nous sommes ensuite confrontés à un problème d'optimisation, que nous avons résolu par l'utilisation de l'algorithme du recuit simulé.
Nous considérons deux ensembles :

- L'ensemble des variables, $\mathcal{V} = \{V_1, V_2, \ldots, V_p\}$
- L'ensemble des individus, $\mathcal{I} = \{w_1, w_2, \ldots, w_n\}$

Pour tout couple d'individus $(w_i, w_j) \in \mathcal{I}^2$, on associe à chaque variable une fonction de classement qui va jouer le rôle d'un juge, celui de classer deux individus dans une même classe ou dans deux classes différentes.
Cette fonction pour une variable V_k est une application

$$A_k(.,.) : \mathcal{I}^2 \rightarrow \{0, 1\}$$

tel que : $\forall (w_i, w_j) \in \mathcal{I}^2; w_i \neq w_j$

- $A_k(w_i, w_j) = 1$ Si pour la variable V_k les deux individus w_i et w_j sont dans une même classe.

- $A_k(w_i, w_j) = 0$ Si pour la variable V_k les deux individus w_i et w_j sont dans deux classes différentes.

Nous supposons que chaque fonction de classement a une rationalité fixe, nous pouvons traduire cette rationalité par la symétrie de A^K :

$$\forall (w_i, w_j) \in \mathcal{I}^2, A_k(w_i, w_j) = A_k(w_j, w_i)$$

4.2 Présentation du problème

Étant donné les classements des p variables $A_1, A_2, \ldots, A_p$ nous cherchons à construire sur l'ensemble $\mathcal{I}$, un classement collectif qui soit la meilleure agrégation possible qui, à la fois va engendre sur l'ensemble $\mathcal{I}$ une relation d'equivalence et qui va maximiser le nombre des concordances entre le classement collectif et les ensembles de classement des variables.

Une partition de l'ensemble $\mathcal{I}$ peut être assimilée à un vote défini sur $\mathcal{I}$ qui peut prendre autant des modalités qu'il y a d'éléments dans la partition.
Nous identifierons cette partition à une application $X(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$.

Étant données, deux applications $X(.,.)$ et $Y(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$, on appelle concordance de $X(.,.)$ et de $Y(.,.)$ notée $C(X,Y)$ le nombre entier défini par :

$$C(X,Y) = \frac{1}{n^2} Card\{(w_i, w_j)\,;\, X(w_i, w_j) = Y(w_i, w_j)\}$$

Cette concordance peut se décomposer en deux catégories de concordances :

- Une concordance positive C^+

$$C^+(X,Y) = \frac{1}{n^2} Card\{(w_i, w_j)\,;\, X(w_i, w_j) = Y(w_i, w_j) = 1\}$$

- Une concordance négative C^-

$$C^-(X,Y) = \frac{1}{n^2} Card\{(w_i, w_j)\,;\, X(w_i, w_j) = Y(w_i, w_j) = 0\}$$

Étant donnée une partition caractérisée par l'application $X(.,.)$ de $\mathcal{I}^2$ dans $\{0,1\}$ et le classement d'une variable V_k, la concordance entre cette partition et le classement de la variable V_k est donnée par $C(X, A_k)$.

Les différents classements de variables $A_k; k = 1, \ldots, p$ étant données, nous cherchons X qui maximise l'expression :

$$C(X, \mathcal{A}) = \sum_{k=1}^{p} C(X, A_k) \tag{4.1}$$

Ce problème de maximisation peut être formulé en terme de programmation linéaire en nombre entiers.

En effet, pour tout $(w_i, w_j) \in \mathcal{I}^2$ on pose $x_{ij} = X(w_i, w_j)$ et pour tout $k = 1, \ldots, p$ on pose $a_{ij}^k = A_k(w_i, w_j)$

$$\begin{aligned} C(X, A_k) &= \frac{1}{n^2} Card\{(w_i, w_j)\,;\, X(w_i, w_j) = A_k(w_i, w_j)\} \\ &= \frac{1}{n^2} Card\{(w_i, w_j)\,;\, x_{ij} = a_{ij}^k\} \end{aligned}$$

Le couple (w_i, w_j) contribue donc à la cohérence dans tous les cas suivants :

- Cas 1 : $x_{ij} = a_{ij}^k = 1$
- Cas 2 : $x_{ij} = a_{ij}^k = 0$

Le premier cas est équivalent à : $x_{ij} a_{ij}^k = 1$
Le deuxième cas est équivalent à : $(1 - x_{ij})(1 - a_{ij}^k) = 1$
Par suite :

$$C(X, \mathcal{A}) = \frac{1}{n^2} \sum_{k=1}^{p} \sum_{(i,j) \in \mathcal{I}^2} (1 - x_{ij})(1 - a_{ij}^k) + x_{ij} a_{ij}^k$$

4.2.1 Etude du critère d'optimisation

Cas de données complètes :

Ce problème est connu sous le nom de "Clique partitioning" et un algorithme utilisant la programmation linéaire a été proposé dans [GW89] .
Pour résoudre le problème, nous proposons une approche heuristique : l'algorithme du recuit simulé.

Nous associons à une application X définie sur $\mathcal{I}^2$ deux vecteurs (x_{ij}) et $(\bar{x}_{ij})$ définis comme suit : $\forall (w_i, w_j) \in \mathcal{I}^2$

$$\begin{cases} x_{ij} = 1 \text{ Si } X(w_i, w_j) = 1 \\ x_{ij} = 0 \text{ Sinon} \end{cases}$$

et

$$\begin{cases} \bar{x}_{ij} = 1 \text{ Si } X(w_i, w_j) = 0 \\ \bar{x}_{ij} = 0 \text{ Sinon} \end{cases}$$

Nous associons à la variable $V_k; (k = 1, \ldots, p)$ les deux vecteurs (a_{ij}^k) et $(\bar{a}_{ij}^k)$ définis comme suit $\forall (w_i, w_j) \in \mathcal{I}^2$

$$\begin{cases} a_{ij}^k & = & 1 \text{ Si } A_k(w_i, w_j) = 1 \\ \\ a_{ij}^k & = & 0 \text{ Sinon} \end{cases}$$

et

$$\begin{cases} \bar{a}_{ij}^k & = & 1 \text{ Si } A_k(w_i, w_j) = 0 \\ \\ \bar{a}_{ij}^k & = & 0 \text{ Sinon} \end{cases}$$

Il y a une concordance entre la partition donnée par X et la partition donnée par A_k si :

$(x_{ij} = 1 \text{ et } a_{ij}^k = 1)$ ou si $(\bar{x}_{ij} = 1 \text{ et } \bar{a}_{ij}^k = 1)$
Nous notons

$$\begin{cases} q_{ij}^k & = & 1 \text{ s'il y a une concordance entre } X(w_i, w_j) \text{ et } A_k(w_i, w_j) \\ \\ q_{ij}^k & = & 0 \text{ Sinon} \end{cases}$$

le nombre de concordances $C(X, A_k) = \sum_{(i,j)\in\mathcal{I}^2} q_{ij}^k$ et le nombre de concordance entre X et les partitions données par $A_k; (k = 1, \ldots, p)$ peut s'écrire :

$$\begin{aligned} C(X, \mathcal{A}) = \sum_{k=1}^{p} C(X, A_k) &= \sum_{k=1}^{p} \sum_{(i,j)\in\mathcal{I}^2} [x_{ij} a_{ij}^k + \bar{x}_{ij} \bar{a}_{ij}^k] \\ &= \sum_{k=1}^{p} \sum_{(i,j)\in\mathcal{I}^2} [x_{ij} a_{ij}^k + (1 - x_{ij}) \bar{a}_{ij}^k] \\ &= \sum_{k=1}^{p} \sum_{(i,j)\in\mathcal{I}^2} \bar{a}_{ij}^k + \sum_{k=1}^{p} \sum_{(i,j)\in\mathcal{I}^2} [a_{ij}^k - \bar{a}_{ij}^k] x_{ij} \end{aligned}$$

$\forall (w_i, w_j) \in \mathcal{I}^2$ nous posons ;

$$r_{ij} = \sum_{k=1}^{p} a_{ij}^k \text{ (Nombre de variables qui classent } w_i \text{ et } w_j \text{ dans une même classe)}$$

et

$$\bar{r}_{ij} = \sum_{k=1}^{p} \bar{a}_{ij}^{k} \text{ (Nombre de variables qui classent } w_i \text{ et } w_j \text{ en deux classes différentes)}$$

$$s_{ij} = r_{ij} - \bar{r}ij$$

L'expression (4.1) devient alors

$$C(X, \mathcal{A}) = \sum_{(i,j)\in\mathcal{I}^2} \bar{r}_{ij} + \sum_{(i,j)\in\mathcal{I}^2} s_{ij} x_{ij} \tag{4.2}$$

Cas de données manquantes :

Si la fonction de classement de variable V_k juge que deux individus sont non classifiables, sa réponse sera M (Manquant).
Dans ce cas, toutes les fonctions de classement $A_k; k = 1, \ldots, p$ seront des applications de l'ensemble $\mathcal{I}^2$ dans l'ensemble $\{0, 1, M\}$ avec la convention suivante :
$A_k(w_i, w_j) = M \Leftrightarrow$ la fonction de classement pour la variable V_k juge que (w_i, w_j) est non classifiable.

En plus des vecteurs $(a_{ij}^k)_{(w_i,w_j)\in\mathcal{I}^2}$ et $(\bar{a}_{ij}^k)_{(w_i,w_j)\in\mathcal{I}^2}$, nous associons à la variable V_k le vecteur $(v_{ij}^k)_{(i,j)\in\mathcal{I}^2}$ défini comme suit :

$$\begin{cases} v_{ij}^k &= 1 \text{ Si } A_k(w_i, w_j) = M \\ \\ v_{ij}^k &= 0 \text{ Sinon} \end{cases}$$

Nous remarquons que la propriété $a_{ij}^k = 1$ ou $\bar{a}_{ij}^k = 1$ n'est vraie que si la variable V_k a donné un avis sur le couple (i, j).
D'autre part, Nous avons : $r_{ij} + \bar{r}_{ij} \leq p$ (avec égalité si et seulement si toutes les variables ont exprimé un avis sur le couple $(w_i, w_j) \in \mathcal{I}^2$.

La prise en compte de ces remarques rendra tous les calcules effectués précédemment valables, et la relation (4.2) sera toujours vraie.

4.2.2 Calcule du nombre de désaccords

Le nombre N d'avis exprimés par l'ensemble des variables peut s'écrire :

$$N = \begin{cases} \frac{pn(n-1)}{2} \text{ S'il n'y a pas des données manquantes} \\ \\ \sum_{(i,j)\in\mathcal{I}^2} (r_{ij} + \bar{r}_{ij}) \text{ S'il y a des données manquantes} \end{cases} \tag{4.3}$$

Le nombre de désaccords $N_D(X)$ entre X et les avis $A_k; (k = 1, \ldots, p)$, est donc donné par :

$$\begin{aligned} N_D(X) &= N - C(X, \mathcal{V}) \\ &= N - \sum_{(i,j)\in\mathcal{I}^2} \bar{r}_{ij} - \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij} \\ &= \sum_{(i,j)\in\mathcal{I}^2} r_{ij} + \sum_{(i,j)\in\mathcal{I}^2} \bar{r}_{ij} - \sum_{(i,j)\in\mathcal{I}^2} \bar{r}_{ij} - \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij} \\ &= \sum_{(i,j)\in\mathcal{I}^2} r_{ij} - \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij} \end{aligned}$$

4.3 Formulation du problème

La solution de notre problème revient à maximiser l'expression $C(X, \mathcal{A})$ donné par (4.2) qui revient à maximiser la quantité linéaire :

$$\sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$$

En fin, pour poser la propriété de transitivité imposée à X par des contraintes linéaires sur les x_{ij}, notons :

$$T = \{(i, j, k) \in I^3 | 1 \leq i < j < k \leq n\}$$

Donc la propriété de transitivité est équivalente aux trois conditions linéaires suivantes :

$$\forall (i, j, k) \in T \begin{cases} x_{ij} + x_{jk} - x_{ik} \leq 1 \\ x_{ij} - x_{jk} + x_{ik} \leq 1 \\ -x_{ij} + x_{jk} + x_{ik} \leq 1 \end{cases} \tag{4.4}$$

Le problème devient un problème de programmation linéaire en nombres entiers suivant :

$$max \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$$

Sachant que $x_{ij} \in \{0,1\}$ avec les contraintes linéaires citées dans le formule (4.4)

Remarque :
Le nombre de variables est de l'ordre de n^2, et le nombre de contraintes est de l'ordre de n^3 et ce problème est connu sous le nom de "click partitioning problem" et c'est un NP-complexe.
Grötschel et Wakbayashi dans [GW89] ont proposé des algorithmes pour la résolution de ce problème.

Dans le cadre de ce travail nous allons présenter une autre solution en basant sur une technique heuristique (le recuit simulé) facile à appliquer, avec un temps du calcul raisonnable.

4.4 Résolution du problème

Nous avons trouvé que la donnée d'une partition X vérifiant la propriété de la transitivité est équivalente à celle d'un vecteur $(x_{ij})_{(i,j)\in\mathcal{I}^2}$, chaque élément x_{ij} étant à valeur dans $\{0,1\}$ et vérifiant le condition (4.4).

Remarquons que la donnée d'un tel vecteur x_{ij} est équivalente à celle d'une partition $\mathcal{P}$ sur I, selon le schéma suivant :

$$x_{ij} = 1 \Leftrightarrow \exists G \in \mathcal{P} \text{ tel que } i \in G \text{ et } j \in G$$

Conformément à (4.2), notons :

$$val(\mathcal{P}) = \sum_{(i,j)\in\mathcal{I}^2} s_{ij}x_{ij}$$

si $\mathcal{P} = \{G_1, \ldots, G_m\}$, nous pouvons écrire :

$$val(\mathcal{P}) = \sum_{h=1}^{m} \sum_{\substack{(i,j)\in G_h \\ i<j}} s_{ij}$$

Notre problème prend alors la formulation suivante :

$$max\{val(\mathcal{P})|\mathcal{P} \in \mathbb{P}\} \tag{4.5}$$

où $\mathbb{P}$ désigne l'ensemble des partitions de I.

Avant de résoudre ce problème (4.5) par la méthode de recuit simulé, nous allons préciser la description d'un voisinage $\mathcal{V}(\mathcal{P})$ d'une partition $\mathcal{P} = \{G_1, \ldots, G_m\}$.

Définition 4.1. *Une partition $\mathcal{P}'$ appartient à $\mathcal{V}(\mathcal{P})$, si elle dérive de $\mathcal{P}$ en déplaçant un seul individu i_0 comme suit :*
Supposons que $i_0 \in G_h$ alors :

- *Soit i_0 est transféré dans un groupe $G_l (l \neq h)$, de $\mathcal{V}(\mathcal{P})$,*
- *soit i_0 constitue un nouveau groupe G_l à lui tout seul.*

Il s'en suit :

$$\Delta = val(\mathcal{P}') - val(\mathcal{P})$$

et, en posant, pour simplifier l'écriture :

$$S_{ij} = \begin{cases} s_{ij} & Si \quad i < j \\ s_{ji} & Si \quad i > j \\ 0 & Si \quad i = j \end{cases}$$

$$\Delta = \sum_{j\in G_l} S_{i_0 j} - \sum_{j\in G_h - \{i_0\}} S_{i_0 j}$$

Nous donnons ci-après le schéma de la procédure :

Algorithme

```
Données : T_0, K_Max, r_0
Résultat : P*
Begin
    Initialisation
        T := T_0
```

```
        choisir au hasard une partition P de I ; P* = P
    Répéter
        change :=FAUX
        Pour k := 1 à K_Max faire
            choisir au hasard P' dans V(P)
            calculer delta = val(P') − val(P)
            tirer au hasard une valeur R dans [0,1] (loi uniforme)
            si delta > 0 ou R < exp(delta/T) alors
                    P := P' ; change :=vrai ;
                si val(P) > val(P*) alors P* := P ;
                fin si
            fin Si
        Fin Pour
    T := T * r_o
jusqu'à Not change
end
```

Remarques :

1. Remarquons que la définition de $\mathcal{V}(\mathcal{P})$ permet d'atteindre toutes les partitions possibles à partir d'une quelconque partition initiale en utilisant le procédé itératif.
2. Les paramètres T_0, K_{Max}, r_o sont respectivement la température initiale, le nombre d'itérations dans la boucle interne, et le coefficient de refroidissement.
3. Nous avons choisi les valeurs de ces paramètres conformément aux recommandations de [KGV83], et plus précisément $K_{Max} = 10 \times card(I), r_o = 0.98$.
4. Quant à T_0, sa valeur est choisie de la façon suivante :
   Soit $\mathcal{P}$ une partition choisie au hasard, et $\mathcal{P}'$ une partition telle que :

$$\mathcal{P}' \in \mathcal{V}(\mathcal{P}), \text{ et } delta = \mathcal{V}(\mathcal{P}') - \mathcal{V}(\mathcal{P}) < 0$$

   Alors la valeur de T_0 doit garantir que $exp(delta/T_0)$ soit proche de 1 en moyenne.

4.4.1 Application

Les données de cette application sont les mêmes qui étaient utilisées par Grötschel et al. dans [GW89]. Nous reprenons cette application car une solution optimale a été fournit dans [GW89] et comme ça nous pouvons comparer notre résultat avec la solution optimale. L'ensemble des individus est constitué de 36 cétacés (Tab. 7.1) et l'ensemble des variables

(juges) est constitué de 15 variables nominales $V_1, \ldots, V_{15}$ pouvant prendre de 2 à 5 valeurs. (Les données manquantes sont représentées par "*").
Ces variables sont les suivantes (groupées en 3 groupes)

1. Morphological parameters
   - V_1 : Neck (0 : does not exist, 1 : exist)
   - V_2 : Form of the head (0 : cylindrical, 1 : conical, 2 : with a curved forehead, 3 : globular, 4 : bat, 5 : convex)
   - V_3 : Size of the head (0 : very big, 1 : medium size)
   - V_4 : Beak (0 : missing, 1 : large, 2 : narrow and short, 3 : narrow and long)
   - V_5 : Dorsal fin (0 : missing, 1 : triangular, 2 : falciform, 3 : backward and falciform)
   - V_6 : Flippers (0 : small, 1 : large and short, 2 : medium size, 3 : long and narrow)
   - V_7 : Set of teeth (0 : on the lower jaw, 1 : on the lower and upper jaw, 2 : without teeth but long baleens, 3 : without teeth but thick baleens, 4 : without teeth but large baleens)
   - V_9 : Blow hole (0 : on the lift side, 1 : on the right side, 2 : on the middle line, 3 : on the middle line with two holes)
   - V_{10} : Color (0 : central part are clearer than dorsal part, 1 : blackish, 2 : no pigmentation, 3 : spotted)
   - V_{14} : Longitudinal furrows on the throat (0 : does not exist, 1 : a small number exists, 2 : a big number exists)
2. Osteological parameters
   - V_{11} : Cervical vertebrae (0 : free, 1 : partly or complectly welded)
   - V_{12} : Lachrymal and jugal bones (0 : form one piece, 1 : are independent)
   - V_{15} : Head bones (0 : symmetrical, 1 : slightly unsymmetrical, 2 : unsymmetrical, 3 : very unsymmetrical)
3. Behavioral parameters
   - V_8 : Feeding (0 : feed of squish, 1 : feed of fish, 2 : feed on seal, 3 : feed of plankton)
   - V_{13} : Habitat (0 : rivers, 1 : temperate or warm seas, 2 : cold seas, 3 : coasts, 4 : variable)

L'application de notre méthode sur les données du tableau (Tab. 7.1.), donne une partition en 7 classes qui coïncide avec la partition optimale selon [GW89].

```
---Recherche de partitions---
---CARACTÉRISTIQUES DES DONNÉES---
Nombre d'individus : 36 Nombre de variables : 15 Nombre d'avis
exprimés : 9096 Pourcentage de concordances : 0.705
```

Individu	Variables														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Balaena	0	5	0	0	0	1	2	3	3	4	1	*	2	0	0
Balaenoptea	0	4	0	0	3	0	4	3	3	0	0	*	4	2	0
Balaenoptea Mus	0	4	0	0	3	3	4	3	3	3	0	*	4	2	0
Berardius	1	2	1	2	2	0	0	0	2	0	1	1	2	1	2
Cephalorhynehus	0	1	1	1	2	2	1	0	1	4	1	1	2	0	1
Delphinapterus	1	3	1	0	0	1	1	1	1	2	0	1	2	0	2
Delphinus	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1
Eschrichtius	0	1	0	0	0	3	3	3	3	3	0	*	4	1	0
Eubalaena	0	5	0	0	0	1	2	3	3	1	1	*	2	0	0
Globicephala	0	3	1	0	2	3	1	0	1	3	1	1	1	0	1
Grampus	0	3	1	0	2	3	0	0	1	0	1	1	4	0	1
Hyperoodon	1	2	1	2	2	0	0	0	2	0	1	1	4	1	2
Inia	1	2	1	3	1	1	1	1	2	0	0	0	0	0	*
Kogia	0	0	0	0	1	0	0	0	0	0	1	*	4	0	3
lagenorhynchus	0	2	1	2	2	2	1	1	1	0	1	1	2	0	1
Lipotes	1	2	1	3	1	1	1	1	0	0	0	0	0	0	*
Lissodelphis	0	1	1	1	0	2	1	1	1	0	1	1	1	0	1
Megaptera	0	4	0	0	3	3	4	3	3	3	0	*	4	2	0
Mesoplodon	1	1	1	2	2	0	0	0	2	0	1	1	4	1	2
Monodon	1	3	1	0	0	1	0	0	1	3	0	1	2	0	2
Neophalaena	0	1	0	0	3	0	2	3	3	0	1	*	2	0	0
Neophocaena	0	1	1	0	0	0	1	1	1	3	1	1	4	0	1
Orcaella	1	3	1	0	2	0	1	1	1	1	1	1	3	0	1
Orcinus	0	3	1	0	2	1	1	2	1	3	1	1	2	0	1
Phocaena	0	1	1	0	1	0	1	1	1	0	1	1	4	0	1
Physeter	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3
Platanista	1	2	1	3	1	1	1	1	2	1	0	0	0	0	1
Pseudorea	0	3	1	0	2	3	1	0	1	1	1	1	4	0	1
Sotalia	0	2	1	2	2	2	1	1	1	2	1	1	3	0	1
Sousa	0	2	1	2	2	2	1	1	1	0	1	1	3	0	1
Stenella	0	2	1	2	2	2	1	1	1	3	1	1	4	0	1
Steno	0	1	1	2	2	2	1	1	1	0	1	1	1	0	1
Stenodelphis	1	2	1	2	1	1	1	1	2	1	0	0	0	0	*
Tasmacetus	0	3	1	2	2	0	0	0	2	0	1	1	2	1	2
Tussio	0	2	1	2	2	2	1	1	1	0	1	1	4	0	1
Ziphus	0	1	1	2	2	0	0	0	2	3	1	1	4	1	2

Tab. 4.1 – Données cétacés

```
--- RÉSULTATS ---
Partition obtenue
--------------------------------
Classe 1 {Delphinapterus, Monodon}

Classe 2 {Balaenoptea, Balaenoptea Mus, Eschrichtius,  Megaptera}

Classe 3 {Balaena, Eubalaena, Neophalaena}

Classe 4 {Inia, Lipotes, Platanista, Stenodelphis}

Classe 5 {Kogia, Physeter}

Classe 6 {Berardius, Hyperoodon, Mesoplodon, Tasmacetus, Ziphus}

Classe 7 {Cephalorhynehus, Delphinus, Globicephala, Grampus,
lagenorhynchus, Lissodelphis, Neophocaena, Orcaella, Orcinus,
Phocaena, Pseudorea, Sotalia, Sousa, Stenella, Steno, Tussio}
```

Remarques :

Du point de vue de l'agrégation des opinions, nous remarquons les points suivants :

- Du fait des données manquantes, le nombre total N d'avais exprimés est de 9096, (soit 6409 le nombre d'accords $NA(X)$ entre l'opinion X, correspondant à la partition $\mathcal{P}$, et celle de 15 juge et soit 2687 le nombre de désaccords)
- Le pourcentage de concordances $\tau = NA(X)/N$ est de 0.705. Cette information est importante pour situer la qualité de la partition obtenue.
- Le pourcentage de concordances par juge $\tau_k = NA(X, A^k)/N_k$, $k = 1, \ldots, 15$ est très variable de 0.579 pour V_1 et de 0.928 pour V_{15}. Cette information éclaire sur le poids des juges dans la contribution de la classification obtenue.

4.5 Extension de la méthode

Nous pouvons résumer le problème de la classification des données de la manière suivante :

Étant donné une ensemble d'individus $\mathcal{I}$ avec une mesure de dissimilarité sur l'ensemble

des couples d'individus. Plus précisément, une dissimilarité est une application ρ de $\mathcal{I}^2$ dans $\mathbb{R}^+$ qui vérifie les propriétés :

- $\forall (i,j) \in \mathcal{I}^2, \rho(i,j) = \rho(j,i)$
- $\forall i \in \mathcal{I}, \rho(i,i) = 0$

Une dissimilarité est donc totalement définie par sa restriction à $\mathcal{I}^2$.
Etant donné une partition $\mathcal{P}\{G_1, \ldots, G_m\}$ de $\mathcal{I}$, nous pouvons définir comme précédemment une mesure qui, dans ce cas, est fonction décroissante de la qualité de $\mathcal{P}$:

$$Qval(\mathcal{P}) = \sum_{h=1}^{m} \sum_{\substack{(i,j) \in G_h \\ i<j}} \rho(ij)$$

Mais le problème de la recherche d'une partition qui minimise $Qval(\mathcal{P})$ sur $\mathbb{P}$ n'a pas de sens, car du fait de la positivité de $\rho(i,j)$, la solution de ce problème est triviale, à savoir la partition comportant un individu par classe.

C'est pourquoi, on fixe le nombre de classes de la partition cherchée, et, nous notons $\mathbb{P}_m$ l'ensemble des partitions de $\mathcal{I}$ en m classes, donc nous cherchons à résoudre le problème :

$$min[Qval(\mathcal{P}|\mathcal{P} \in \mathbb{P}_m)]$$

Nous pouvons ensuite faire varier m pour trouver la meilleure partition mais cela pose la problème de la recherche du "bon" nombre de classes qui est généralement assez délicat.

Le fait que le choix du nombre de classes soit réglé par le problème d'optimisation tel que nous l'avons traité dans ce chapitre est un avantage très important de la présente formulation.
Nous pouvons donc ramener un problème de classification où l'on est en présence d'une dissimilarité ρ, à notre présente formulation (4.5) de la façon suivante :
Notons

$$\bar{\rho} = \frac{1}{|\mathcal{I}|} \sum_{(i,j) \in \mathcal{I}^2} \rho(i,j)$$

Nous posons

$$s_{ij} = \bar{\rho} - \rho(i,j)$$

La quantité s_{ij} n'est plus un entier relatif mais c'est un réel vérifiant :

$$\bar{\rho} - \rho_{max} \leq s_{ij} \leq \bar{\rho}$$

où $\rho_{max} = max[\rho(i,j)|(i,j) \in \mathcal{I}^2]$
Pour une partition $\mathcal{P} = \{G_1, \ldots, G_m\}$ de $\mathcal{I}$, on définit :

$$val(\mathcal{P}) = \sum_{h=1}^{m} \sum_{\substack{(i,j) \in G_h \\ i<j}} s_{ij}$$

et le problème :

$$max[val(\mathcal{P})|\mathcal{P} \in \mathbb{P}]$$

est de même nature celui que nous avons traité ; la même procédure basée sur le Recuit Simulé s'y applique aussi bien.

4.6 Cohérence et ressemblance entre groupes

Définition

Définition 4.2. *ressemblance*
Étant donné deux groupes A et B de $\mathcal{I}$ vérifiant :

$$A, B \neq \phi, A, B \subset \mathcal{I}, A \cap B = \phi$$

nous appelons ressemblance entre le groupe A et le groupe B, la quantité $\delta(A, B)$

$$\delta(A, B) = \sum_{i \in A} \sum_{j \in B} S_{ij}$$

évidement, nous avons les propriétés suivantes

$$\delta(A, B) = \delta(B, A)$$

$$\begin{cases} \forall A, B, C \subset \mathcal{I}, A \cap B = B \cap C = A \cap C = \phi \\ \delta(A \cup B, C) = \delta(A, C) + \delta(B, C) \end{cases} \tag{4.6}$$

Définition 4.3. *Cohérence*
Étant donné une partie $A \in \mathcal{I}$, nous appelons cohérence de A, la quantité $\gamma(A)$:

$$\begin{cases} si\ card(A) = 1 \Rightarrow \gamma(A) = 0 \\ si\ card(A) > 1 \Rightarrow \gamma(A) = \sum_{i \in A} \sum_{j \in A, i<j} S_{ij} \end{cases}$$

Et nous pouvons remarquer que la relation entre ressemblance et cohérence est donnée par :

$$\delta(A \cup B) = \gamma(A) + \gamma(B) + \gamma(A, B) \tag{4.7}$$

Considérons maintenant une partition $\mathcal{P} = \{G_1, \ldots, G_m\}$ en m groupes.
Il est naturel de lui associer la matrice de cohérence ressemblance $S(\mathcal{P})$, symétrique, de dimension (m, m), définie comme suite :

$$\begin{cases} S_{hl}(\mathcal{P}) = \delta(G_h, G_l)\, \forall h, l \in \{1, \ldots, m\} h \neq l \\ S_{hh}(\mathcal{P}) = \gamma(G_h)\, \forall h \in \{1, \ldots, m\} \end{cases}$$

Nous allons noter $\mathcal{P}_n$ la partition discrète de $\mathcal{I}$:

$$\mathcal{P}_n = \{\{1\}, \{2\}, \ldots, \{m\}\}$$

et nous avons évidemment :

$$S_{ij}(\mathcal{P}_n = S_{ij}\, \forall i, j \in \{1, \ldots, n\}$$

Par ailleurs, nous pouvons associer à la partition $\mathcal{P} = \{G_1, \ldots, G_m\}$ les deux indicateurs suivants :

$$\gamma[\mathcal{P}] = \sum_{h=1}^{m} \gamma(G_h)$$

$$\delta[\mathcal{P}] = \sum_{h=1}^{m} \sum_{l=h+1}^{m} \delta(G_h, G_l)$$

$\gamma[\mathcal{P}]$ est une mesure de la cohérence intra-groupes, et $\delta[\mathcal{P}]$ est une mesure de la cohérence inter-groupes.

Remarques

1. Nous pouvons remarquer :

$$val(\mathcal{P}) = \gamma[\mathcal{P}]$$

   Ainsi le problème (4.5) revient à chercher une partition qui maximise la cohérence intra-groupes.

2. D'autre part :

$$\gamma[\mathcal{P}] + \delta[\mathcal{P}] = \sum_{(i,j)\in\mathcal{I}} s_{ij} \tag{4.8}$$

   et le problème (4.5) est équivalent au problème suivant :

$$min[\delta[\mathcal{P}] \,|\, \mathcal{P} \text{ partition de } \mathcal{I}]$$

3. En désignant par $\mathcal{I}_1$ la partition grossière de $\mathcal{I}$: $\mathcal{P}_1 = \{1, 2, \ldots, n\}$

$$val(\mathcal{P}_1) = \gamma[\mathcal{P}_1] = \sum_{(i,j)\in\mathcal{I}} s_{ij}$$

4. Soit X le classement agrégé associé à la partition $\mathcal{P}$. La relation (4.8) permet d'établir une expression commode pour le nombre de concordance entre X et les classements des variables, $C(X, \mathcal{V})$.

   En explicitant les s_{ij} la relation (4.8) devient :

$$\sum_{(i,j)\in\mathcal{I}} r_{ij} - \sum_{(i,j)\in\mathcal{I}} \bar{r}_{ij} = \gamma[\mathcal{P}] + \delta[\mathcal{P}]$$

   Par ailleurs :

$$\sum_{(i,j)\in\mathcal{I}} r_{ij} + \bar{r}_{ij} = N$$

   Il s'en suit :

$$\sum_{(i,j)\in\mathcal{I}} r_{ij} = \frac{\gamma[\mathcal{P}] + \delta[\mathcal{P}] + N}{2}$$

La relation (4.2) devient donc, en tenant compte du fait que $val(\mathcal{P}) = \gamma(\mathcal{P})$:

$$C(X, \mathcal{A}) = N - \frac{\gamma[\mathcal{P}] + \delta[\mathcal{P}] + N}{2} + \gamma[\mathcal{P}]$$

soit

$$C(X, \mathcal{A}) = \frac{\gamma[\mathcal{P}] - \delta[\mathcal{P}]}{2} + \frac{N}{2}$$

4.7 Structuration de l'ensemble des variables

Nous avons vu dans le paragraphe précédent, comment il était possible de structurer l'ensemble des individus.
Nous pouvons parfois considérer cette description comme très schématique, en particulier lorsque plusieurs tendances contradictoires s'opposent sur l'ensemble des variables ; dans ce cas là :

- Ou bien un groupe important définit à lui seul le classement collectif.
- Ou bien s'établit entre plusieurs groupes un compromis qui reflète mal le classement de chacune des variables.

Dans les deux cas, nous constatons que certaines variables ne sont pas d'accord avec le classement final et le nombre de concordances est assez faible.

Dans le but d'affiner le résultat final, nous pouvons essayer de déterminer plusieurs classements collectifs, chacun d'eux correspondant à un sous ensemble de juges.
Il s'agit donc d'effectuer une classification sur l'ensemble des variables.
Deux approches sont possibles :

- **Classique :** Où le nombre de groupes est fixé par l'utilisateur.
- **Relationnelle :** Consiste à adapter la méthode de recuit simulé en considérant l'ensemble des variables comme un ensemble à classifier.

4.7.1 Formulation classique du problème

Nous considérons deux ensembles $\mathcal{A}$ et $\mathcal{I}$ définis au paragraphe 4.1 avec les même notions et définitions.

Le problème étudié dans le paragraphe précédent consiste à chercher l'application $X^*(K)$ qui maximise

$$C(X, \mathcal{A}) = \sum_{k \in K} C(X, A_k)$$

avec le contrainte que X va engendrer une relation d'équivalence sur $\mathcal{I}$.
Nous désignons par $\mathbb{E}$ l'ensemble des variables qui engendre sur $\mathcal{I}$ une relation d'équivalence.

Pour simplifier, nous supposons que nous cherchons une partition de l'ensemble des variables en deux classes ; soit :

$$\mathcal{P} = \{K_1, K_2\}$$

Nous désignons par $\mathbb{P}_2$ l'ensemble des partitions en deux classes.
D'autre par, nous désignons par :

$$\mathcal{X} = \{X_1, X_2\}$$

un couple d'opinions telles que $X_1, X_2 \in \mathbb{E}$

Chaque variable de K_1 (resp. K_2) exprime son accord par rapport à l'opinion X_1 (resp. X_2).
Ainsi, on définit le nombre de concordances global par :

$$C(\mathcal{X}, \mathcal{P}) = \sum_{k \in K_1} C(X_1, A_k) + \sum_{k \in K_2} C(X_2, A_k)$$

Le problème est donc de déterminer simultanément la meilleure partition $\mathcal{P} = \{K_1, K_2\}$ de l'ensemble des variables, et le couple d'opinions $\mathcal{X} = \{X_1, X_2\}$ associé, de façon à maximiser le nombre de concordances global :

$$Max[C(\mathcal{X}, \mathcal{P}) \,|\, \mathcal{P} = \{K_1, K_2\} \in \mathbb{P}_2, \mathcal{X} = \{X_1, X_2\}, X_1, X_2 \in \mathbb{E}]$$

4.7.2 Une méthode heuristique pour la résolution du problème

Nous allons procéder à la résolution de ce problème en s'inspirant de la méthode des centre mobiles [For65] :

1. **Initialisation :**
   On part d'une partition quelconque de K tirée au hassard :

$$\mathcal{P} = \{K_1, K_2\}$$

2. On détermine les solutions respectives $X^*(K_1)$ et $X^*(K_2)$ de chacun des problèmes $(P^{(1)})$ et $(P^{(2)})$:

$$(P^{(i)}) \qquad max[\sum_{k \in K_i} C(X, A_k) \,|\, X \in \mathbb{E}]$$

   au moyen de la méthode utilisée précédemment.

3. On réaffecte chaque variable au groupe $K_i (i = 1$ ou $2)$ avec le classement agrégé duquel il est le plus en concordance ; plus précisément, pour toute variable $k \in K$

   – Si $k \in K_1$ et $C(X^*(K_2), A_k) > C(X^*(K_1), A_k)$ alors $K_1 = K_1 - \{k\}$ et $K_2 = K_2 \cup \{k\}$
   – Si $k \in K_2$ et $C(X^*(K_1), A_k) > C(X^*(K_2), A_k)$ alors $K_2 = K_2 - \{k\}$ et $K_1 = K_1 \cup \{k\}$
   – Dans tous les autres cas K_1 et K_2 sont inchangés.

4. Si au cours de l'étape (3) au moins un variable k a changé de groupe, on retourne à l'étape (2). Sinon l'algorithme est terminé.

Il faut noter que le résultat ne correspond, en général, qu'à un optimal local.

Nous pouvons assurer que l'algorithme s'arrête après un nombre fini d'étapes en prenant les précautions suivantes :

– Lors de l'étape (3), un juge ne sera changé de groupe que si ce changement induit une augmentation du nombre d'accord,
– Lors de la deuxième itération, (et des itérations ultérieures) on ne modifiera les opinion $X^*(K_1)$ ou $X^*(K_2)$ que si ces modifications induisent une augmentation du nombre d'accords.

Il est claire que ce type d'algorithme peut être généralisé pour la recherche de partitions en plus de deux groupes. La présence des données manquantes ne pose aucun problème particulier.

Conclusion

Nous devons noter d'abord que l'algorithme du recuit simulé comporte un aspect aléatoire, ce qui explique que deux exécutions successives sur le même exemple ne donnent pas forcément le même résultat. Pour cela nous appellerons "Résultat" de l'algorithme, le meilleur des résultats obtenus lors de plusieurs exécutions successives.
La principale avantage de cette méthode de classification, est que les données ne sont pas forcement quantitatives et sa particularité par rapport d'autres méthodes est que nous n'avons pas calculer une distance ou un indice de dissimilarité. En même temps nous avons vu dans l'extension de la méthode que cette approche est applicable dans la cas où nous voulons utiliser une mesure de proximité. Finalement, nous pouvons dire que cette méthode est basée sur l'opinion de l'expert du domaine, qui définit les fonctions de classement.

Chapitre 5

Approche prétopologique

Sommaire

5.1 Introduction **98**

5.2 Eléments de prétopologie **99**

5.2.1 Définitions et propriétés de base : 99

5.2.2 Concepts de préfiltre, filtre, filtre stable 101

5.2.3 Concepts de prévoisinages et caractérisations 102

5.2.4 Comparaison de prétopologies 104

5.2.5 Encadrement d'une structure s par deux structures de type V . 105

5.2.6 Structure de type VDS et relation binaire 107

5.2.7 Relation réflexive et relations d'équivalences associées 107

5.2.8 Encadrement de s par deux structures de type VDS (cas E fini) 110

5.2.9 Concept de continuité 112

5.2.10 Prétopologie induite (second type de problème) 114

5.3 Application à un problème de classification **115**

5.3.1 (Pré)topologie et partition 116

5.3.2 Familles de topologies-classes associées à s et éléments remarquables 117

5.3.3 Processus de construction de $\mathcal{T}_C^+(s)$ et de $\mathcal{T}_C^-(s)$ 117

5.3.4 Produit d'espaces prétopologique 119

5.3.5 Application à la classification 120

5.4 Algorithme de la méthode **127**

5.4.1 Conclusion 129

5.1 Introduction

De nombreux contextes issus de problématiques propres aux sciences sociales nécessitent que l'on soit en mesure d'établir des liens ou d'apprécier des proximités ou des ressemblances entre des individus. L'outil mathématique connu de formalisation du concept de proximité est la topologie, dont souvent on ne connaît que le cas particulier qu'est la métrique. Mais, dans les domaines évoqués, on s'aperçoit assez rapidement que l'axiomatique de la topologie est trop contraignante, parce que trop riche, et n'est pas en mesure d'apporter des solutions intéressantes aux problèmes posés. D'où l'idée de chercher à affaiblir l'axiomatique de la topologie, pour espérer une meilleure adaptation aux problèmes rencontrés : telle est l'idée qui conduit à ce qu'on appellera ici la "prétopologie".

Les premières évocations des travaux de ce type peuvent être attribués à **Frechet** [Fre28]. On trouve, dans la littérature, des propositions d'affaiblissement de l'axiomatique topologique chez **Appert et Fan** [AF51] et chez **P.C. Hammer** [Ham64].
Cech [Cec66] a apporté une importante contribution au développement de structures qui s'avèrent être celles de type **VD** évoquées dans ce travail.
De nombreux travaux de **Brissaud** [Bri75], ont porté sur le développement des structures prétopologiques et leurs applications,en particulier dans le domaine de la modélisation des préférences.
Des contributions ont été apportées par **Auray** [Aur82] dans le cadre d'une approche de la formation des coalitions en théorie des jeux, et Duru [Dur80].
Les structures prétopologiques ont été utilisées, notamment parce qu'elles ont permis une bonne formalisation des multiples concepts de "plus proches voisins", pour traiter des problèmes de classification par, d'imagerie et de reconnaissance : ce sont les travaux de Lamure[Lam87], Emptoz[Emp83], Zighed ou Nicoloyannis [Nic88].
Un ouvrage collectif de synthèse a été publié en 1993 sous le nom acronyme de Belmandt [Bel93].
Des travaux, menés conjointement par des chercheurs de l'équipe "Méthodes d'Aide à la décision" (*LIRIS-MA2D, UMR 5205 du CNRS*) et (*Laboratoire d'Informatique et des Systèmes Complexes*) de l'EPHE (CNRS) sont actuellement en cours en vue de réaliser des algorithmes permettant d'implanter des outils informatiques spécifiques.
Dans le domaine de la classification, la plupart des méthodes ont plus ou moins implicitement recours à un concept de mesure de "proximité". Dans le cas où le contexte est tel qu'une telle "proximité" n'a rien de naturel les outils habituels ne sont guère satisfaisants. Supposons par exemple, qu'on dispose d'une population Ω sur laquelle on a observé divers

caractères et que, pour chacun de ces caractères, l'observation se traduise par une variable X définie sur Ω et prenant ses valeurs dans l'ensemble E des modalités du caractère. Si le caractère est quantitatif alors E peut naturellement être muni d'une mesure de proximité qui pourra être utilisée pour apprécier la ressemblance ou non de deux objets de Ω. Mais si le caractère est purement qualitatif alors ceci n'est plus possible, sauf à prendre le risque de malmener sérieusement la réalité. Il faut alors construire une procédure spécifique au contexte qualitatif, en vue de proposer une formalisation adaptée du concept de "proximité". La prétopologie peut être utile à la réalisation de cet objectif.

5.2 Eléments de prétopologie

On se limitera ici à la présentation des principaux éléments de prétopologie. On trouvera un exposé complet de la théorie dans Belmandt [Bel93].

5.2.1 Définitions et propriétés de base :

Adhérence et Intérieur

Soit E un ensemble non vide. On note $\mathbb{P}(E)$ l'ensemble des parties de E.

On note $c(.)$ ou simplement c, l'application de $\mathbb{P}(E)$ dans lui-même qui, à tout élément A de $\mathbb{P}(E)$ associe A^c, le complémentaire de A dans E.

Si $a(.)$ est une application de $\mathbb{P}(E)$ dans lui-même, et si $i(.)$ est définie par :

$i = c \circ a \circ c$

simplement notée encore $i = cac$, c'est-à dire que, pour toute partie A de E, $i(A) = (a(A^c))^c$, alors on vérifie facilement que $a = cic$; on dit que i **et** a **sont duales**.

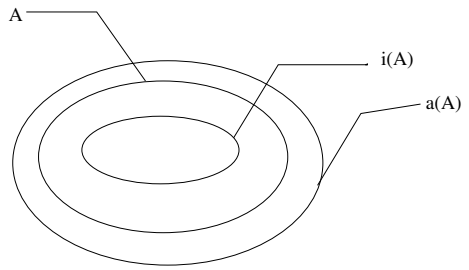

FIG. 5.1 – Adhérence et Intérieur

On vérifie facilement que l'on a les propriétés suivantes :

- P_1 : $i(E) = E \Leftrightarrow a(\emptyset) = \emptyset$,
- P_2 : $\forall A \in \mathbb{P}(E), i(A) \subset A \Leftrightarrow \quad \forall A \in \mathbb{P}(E), A \subset a(A)$

- P_3 : $[\forall A \in \mathbb{P}(E), \forall B \in \mathbb{P}(E), A \subset B \Rightarrow i(A) \subset i(B)]$
  $\Leftrightarrow [\forall A \in \mathbb{P}(E), \forall B \in \mathbb{P}(E), A \subset B \Rightarrow a(A) \subset a(B)]$
- P_4 : $[\forall A \in \mathbb{P}(E), \forall B \in \mathbb{P}(E), i(A \cap B) = i(A) \cap i(B)]$
  $\Leftrightarrow [\forall A \in \mathbb{P}(E), \forall B \in \mathbb{P}(E), a(A \cup B) = a(A) \cup a(B)]$
- P_5 : $[\forall A \in \mathbb{P}(E), i(A) = \bigcap_{x \notin A} i(E - \{x\})] \Leftrightarrow [\forall A \in \mathbb{P}(E), a(A) = \bigcup_{x \in A} a(\{x\})]$
- P_6 : $[\forall A \in \mathbb{P}(E), i(i(A)) = i(A)] \Leftrightarrow [\forall A \in \mathbb{P}(E), a(a(A)) = a(A)]$

La propriété P_4, valable pour deux éléments A et B s'étend sans problème à toute famille finie d'éléments de $\mathbb{P}(E)$. En revanche, l'extension à une famille infinie d'éléments de E n'est due qu'à P_5.

On montre que $P_5 \Rightarrow P_4$ et que $P_4 \Rightarrow P_3$.

Structure Prétopologique sur E

On appelle **structure prétopologique sur E**, ou simplement, **prétopologie sur E**, le couple $s = (i(.), a(.))$, ou simplement $s = (i, a)$

où $i(.)$ et $a(.)$ sont des applications duales possédant les propriétés P_1 et P_2.
Pour toute partie A, $i(A)$ **est l'intérieur de** A et $a(A)$ **l'adhérence de** A.
pour la prétopologie s ; dira encore que, pour s, i est l'intérieur et a l'adhérence.

prétopologie de type V
Si $s = (i, a)$ est une prétopologie sur E qui possède la propriété P_3, on dit que s est **“ de type V ”.**

prétopologie de type VD
Si $s = (i, a)$ est une prétopologie sur E qui possède la propriété P_4, on dit que s est **“ de type VD ”.**

prétopologie de type VDS
Si $s = (i, a)$ est une prétopologie sur E qui possède la propriété P_5, on dit que s est **“ de type VDS ”.**

De ce qui précède, il résulte que toute prétopologie de type **VDS** est de type **VD** et que toute prétopologie de type **VD** est de type **V**.

Par ailleurs, il est facile de voir que si E est fini toute prétopologie de type **VD** est aussi de type **VDS**.

Topologie

Une topologie n'est autre qu'une prétopologie de type **VD** qui possède la propriété P_6.

Désormais, on notera $\mathbf{P}(E)$ l'ensemble des prétopologies sur E.

5.2.2 Concepts de préfiltre, filtre, filtre stable

Soit $\mathcal{F}$ une famille de parties de E.

On dit que $\mathcal{F}$ est un **préfiltre** (de parties de E) si et seulement si elle possède la propriété P_F suivante :

(P_F) $\quad \forall V \in \mathcal{F}, W \supset V \Rightarrow W \in \mathcal{F}$.

On vérifie que toute intersection de préfiltre est un préfiltre, et que toute réunion de préfiltres est aussi un préfiltre.

Si $\mathcal{F}$ est un préfiltre et si $\emptyset \in \mathcal{F}$, alors $\mathcal{F} = \mathbb{P}(E)$: un tel préfiltre est dit **préfiltre dégénéré.**

On dit qu'une famille $\mathcal{B}$ de parties de E est une **base du préfiltre** $\mathcal{F}$ si et seulement si elle a la propriété P_{BF} :

(P_{BF}) $\quad \forall V \in \mathcal{F}, \exists B \in \mathcal{B}$ tel que $B \subset V$.

Un préfiltre peut posséder plusieurs bases et d'ailleurs il est lui-même une de ses bases. Si $\mathcal{B}$ est une base de $\mathcal{F}$, on a évidemment $\mathcal{B} \subset \mathcal{F}$. L'intérêt d'une base $\mathcal{B}$ est d'être une famille " plus petite " que $\mathcal{F}$ qui suffit à caractériser $\mathcal{F}$. En pratique, il arrivera souvent de se donner une famille $\mathcal{B}$ de parties de E et de construire à partir de cette famille un préfiltre dont elle est la base : il est facile de voir que ce préfiltre est unique : on dira que c'est le **préfiltre engendré** par $\mathcal{B}$.

Si $\mathcal{B}$ est un préfiltre tel que deux éléments quelconques A et B de $\mathcal{F}$ ont toujours une intersection non vide, alors dit que $\mathcal{F}$ est un **préfiltre propre** .

On dit qu'un préfiltre $\mathcal{F}$ est **un filtre** si et seulement il possède la propriété F suivante :

(F) $\quad \forall V \in \mathcal{F}, \forall W \in \mathcal{F}, V \cap W \in \mathcal{F}$.

Plus généralement un filtre est un préfiltre stable pour l'intersection finie.

Si un filtre est non propre, alors il est forcément dégénéré.

Un préfiltre (ou un filtre) $\mathcal{F}$ est appelé **préfiltre stable** si et seulement si il possède la propriété FS suivante :

(FS) toute intersection d'éléments de $\mathcal{F}$ est un élément de $\mathcal{F}$.

Si $\mathcal{F}$ est un préfiltre stable, c'est donc obligatoirement un filtre, éventuellement dégénéré.

Si on note $\mathbf{F}$ l'intersection de tous les éléments d'un filtre stable $\mathcal{F}$, $\mathbf{F}$ est élément de $\mathcal{F}$ et est donc le plus petit (au sens de l'inclusion ensembliste) élément de $\mathcal{F}$. Alors $\mathcal{F}$ n'est autre que la famille des parties E qui contiennent $\mathbf{F}$, famille que l'on notera encore $\mathbf{F}^s$. $\{F\}$ est alors la plus petite base de $\mathcal{F}$; par extension $\mathbf{F}$ sera aussi appelée " base du filtre $\mathcal{F}$ ".

Ainsi les propriétés (1), (2) et (3) suivantes sont équivalents

(1)$\mathcal{F}$ filtre stable

(2)$\mathbf{F}^s$, intersection de tous les éléments de $\mathcal{F}$ est un élément de $\mathcal{F}$

(3) $\mathcal{F} = \mathbf{F}^s = \{V \in \mathbb{P}(E)\,; F \subset V\}$.

Remarque : dans la propriété (3), la situation $\mathbf{F} = \emptyset$ n'est pas exclue : dans ce cas $\mathbf{F}^s = \mathbb{P}(E)$ qui se trouve donc être dégénéré.

5.2.3 Concepts de prévoisinages et caractérisations

Soit $s = (i, a)$ une structure prétopologique.

A tout x de E, on associe la famille $\mathcal{V}(x)$ suivante de parties de E :

$$\mathcal{V}(x) = \{V \subset E; x \in i(V)\}$$

et on dit que $\mathcal{V}(x)$ est la famille des prévoisinages (par analogie avec le langage topologique qui prévoisinages est un cas particulier, on dira parfois simplement " voisinages ") de x.

De cette définition, il résulte clairement que tout prévoisinages de x contient x et que E est prévoisinage de chacun de ses points.

On montre alors que la nature d'une prétopologie est caractérisable en termes de prévoisinages. Plus précisément on a les trois propriétés caractéristiques suivantes.

Soit $s = (i, a)$ une structure prétopologique et $\forall x \in E, \mathcal{V}(x) = \{V \subset E\,; x \in i(V)\}$, la famille des prévoisinages de x pour s.

- ***Structures de type* V *:***

  Les deux propositions suivantes sont équivalentes :

  $(PV1)$ s est de type V

  $(PV2)$ pour tout x de $E, \mathcal{V}(x)$ est un préfiltre.
- ***Structures de type* VD *:***

  Les deux propositions suivantes sont équivalentes :

  $(PVD1)$ s est de type VD

  $(PVD2)$ pour tout x de $E, \mathcal{V}(x)$ est un filtre.
- ***Structures de type* VDS *:***

  Les deux propositions suivantes sont équivalentes :

  $(PVDS1)$ s est de type VDS

  $(PVDS2)$ pour tout x de $E, \mathcal{V}(x)$ est un filtre stable.

Etant donnée une structure $s = (i, a)$, il résulte de la définition même que le concept de prévoisinage est équivalent à celui d'intérieur. Plus précisément : si on se donne un intérieur

i alors pour tout x la famille $\mathcal{V}(x)$ des prévoisinages de x est connue ; inversement si à tout x on associe une famille $\mathcal{V}(x)$ de parties de E qui, chacune contient x, et telle que E appartient à $\mathcal{V}(x)$, alors l'application $i(.)$ définie par :

$$\forall A \in \mathbb{P}(E), i(A) = \{x \in E; A \in \mathcal{V}(x)\}$$

est l'intérieur d'une structure prétopologique $s = (i, a)$ où $a = c \circ i \circ c$, et dont, pour tout x ; $\mathcal{V}(x)$ est la famille de prévoisinages.

Lorsque s n'est pas de type V, l'une au moins des familles $\mathcal{V}(x)$ n'est pas un préfiltre et inversement.

La définition du concept de prévoisinage établit un lien caractéristique entre intérieur et prévoisinages. On peut se demander si un tel lien caractéristique existe entre adhérence et prévoisinages. Un tel lien est donné ci-dessous.

Soit $s = (i, a)$ une structure prétopologique quelconque, avec $a = c \circ i \circ c$ et, $\forall x \in E, \mathcal{V}(x) = \{V \subset E\,; x \in i(V)\}$, et soit x élément quelconque de E. On considère les deux affirmations suivantes :

$(A1)$ $\forall V \in \mathcal{V}(x), V \cap A \neq \emptyset$

$(A2)$ $x \in a(A)$

alors $A1 \Rightarrow A2$.

L'exemple suivant construit avec $E = \{x, y, z\}$ et $i(.)$ définie dans le tableau ci-dessous, montre que la réciproque n'est pas vraie :

$\mathbb{P}(E)$	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\{x, y\}$	$\{x, z\}$	$\{y, z\}$	E
i(.)	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\emptyset$	$\emptyset$	$\{z\}$	E
$a = c \circ i \circ c$	$\emptyset$	$\{x, y\}$	E	E	$\{x, y\}$	$\{x, z\}$	$\{y, z\}$	E

On a alors :

$\mathcal{V}(x) = \{\{x\}, E\}$

$\mathcal{V}(y) = \{\{y\}, E\}$

$\mathcal{V}(z) = \{\{z\}, \{y, z\}, E\}$.

On a par exemple $z \in a(\{y\})$, donc $(A2)$ est vérifiée pour z en place de x et $A = \{y\}$, alors que $\{z\} \in \mathcal{V}(z)$ et $\{z\} \cap \{y\} = \emptyset$ et $(A1)$ n'est pas vraie. On remarquera, par ailleurs, que ni $\mathcal{V}(x)$ ni $\mathcal{V}(y)$ ne sont des préfiltres.

En revanche, il est facile de vérifier que si $s = (i, a)$ est de type V alors on a $(A1) \Leftrightarrow (A2)$. Dans ce cas, intérieur et adhérence ont une caractérisation en termes de voisinages, et ces caractérisations sont rappelées ci-dessous :

(Int) $x \in i(A) \Leftrightarrow \quad \forall V \in \mathcal{V}(x) = x \in i(V)$

$(Adh)\ x \in a(A) \Leftrightarrow \quad \forall V \in \mathcal{V}(x), V \cap A \neq \quad \emptyset$

Cette caractérisation qui ne vaut que pour les structures de type V est extrêmement utile et justifie à elle seule la facilité d'utilisation des structures de type V contrairement à celles qui ne le sont pas.

Remarque

Si $s = (i, a)$, avec $a = c \circ i \circ c$ est une structure qui n'est pas de type V, on peut construire l'application $\mathring{a}$ suivante, de $\mathbb{P}(E)$ dans lui-même, définie par :

$\forall A \in \mathbb{P}(E)$, $\mathring{a}(A) = \{x \in E; \forall V \in \mathcal{V}(x), V \cap A \neq \quad \emptyset\}$.

On montre sans problème que, pour toute partie A de E, on toujours $\mathring{a}(A) \subset a(A)$ et que la structure $\mathring{s} = (\mathring{i}, \mathring{a})$ dans laquelle $\mathring{i}$ est définie, pour toute partie A de E, par $\mathring{i}(A) = (\mathring{a}(A^c))^c$ est de type V et que, pour cette structure, la famille des prévoisinage de tout x de E n'est autre que le préfiltre de base $\mathcal{V}(x)$.

On vérifie ces affirmations sur l'exemple précédent complété par le tableau ci-dessous.

$\mathbb{P}(E)$	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\{x, y\}$	$\{x, z\}$	$\{y, z\}$	E
i(.)	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\emptyset$	$\emptyset$	$\{z\}$	E
$a = c \circ i \circ c$	$\emptyset$	$\{x, y\}$	E	E	$\{x, y\}$	$\{x, z\}$	$\{y, z\}$	E
$\mathring{a}$	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\{x, y\}$	$\{x, z\}$	$\{y, z\}$	E

5.2.4 Comparaison de prétopologies

Soient $s_1 = (i_1, a_1)$ et $s_2 = (i_2, a_2)$ deux prétopologies sur E.

Les deux propriétés suivantes sont équivalentes :

- (i) $\forall A \in \mathbb{P}(E),\ i_2(A) \subset i_1(A)$
- (ii) $\forall A \in \mathbb{P}(E),\ a_1(A) \subset a_2(A)$

Si de plus elles sont de type V, alors chacune de ces deux propriétés est équivalente à la suivante :

- (iii) $\forall x \in E, \mathcal{V}_2(x) \subset \mathcal{V}_1(x)$
  avec $\mathcal{V}_1(x)$ famille des prévoisinages de x pour s_1 et $\mathcal{V}_2(x)$ famille des prévoisinages de x pour s_2.

Etant données deux prétopologies s_1 et s_2 on dira que la prétopologie s_1 est plus fine que s_2 si et seulement si l'une des deux propriétés équivalentes (i), (ii) est vraie. On note alors $s_1 \succcurlyeq_{pf} s_2$.

Si s_1 et s_2 sont de type V, s_1 est plus fine que s_2 si et seulement si l'une ou l'autre des propriétés (i), (ii) et (iii) est vérifiée.

Si s_1 est plus fine que s_2 et s'il existe A tel que $i_2(A)$ est contenu strictement dans $i_1(A)$ ou, ce qui est équivalent, s'il existe B tel que $a_1(B)$ est strictement contenu dans $a_2(B)$ alors s_1 est strictement plus fine que s_2. Dans le cas où les structures s_1 et s_2 sont de type V, s_1 est strictement plus fine que s_2 si et seulement s'il existe $x \in E$ tel que $\mathcal{V}_2(x)$ est strictement inclus dans $\mathcal{V}_1(x)$.

Sur l'ensemble $\mathbf{P}$(E) de toutes les prétopologies possibles sur E, qui contient l'ensemble $\mathbf{T}$(E), de toutes les topologies sur E, la relation $\succcurlyeq_{pf}$ induit une structure de treillis.

Ce treillis possède un "plus grand élément" qui est la topologie discrète, notée $\mathbf{t}_d$ et un "plus petit élément" qui est la topologie grossière notée $\mathbf{t}_g$.

On remarquera que, sur ce treillis, la relation d'ordre $\succcurlyeq_{pf}$ n'est pas totale. Par suite, en général, un sous-ensemble de $\mathbf{P}$(E), peut ne pas posséder de plus grand ou de plus petit élément, mais seulement des éléments maximaux ou minimaux.

Il est en général très difficile d'établir des propriétés intéressantes pour les structures qui ne sont pas de type V, puisque, par définition, elles nécessitent une description "in extenso" excluant toute possibilité d'établir des résultats généraux. Pour cette raison, on cherchera souvent à travailler avec des structures ayant au minimum la propriété d'être de type V. De ce fait, il sera alors utile d'être capable, étant donnée une structure $s = (i, a)$ quelconque, de trouver une structure de type V qui soit suffisamment " proche " d'elle pour pouvoir lui être substituée. Trouver une solution à ce problème nécessite qu'au préalable on soit en mesure de donner un sens au mot "proche" : on conviendra que le sens qu'on attribue au mot "proche" correspond au sens de "proche" au sein du treillis $\mathbb{P}(E)$. Cette précision conduit alors à la propriété suivante.

5.2.5 Encadrement d'une structure s par deux structures de type V

Soit $s = (i, a)$ une structure quelconque. Pour tout x de E, on note $\mathcal{V}(x) = \{V \subset E; x \in i(V)\}$, la famille des prévoisinages de x pour cette structure. On note $\mathcal{P}_V^+(s, E)$ l'ensemble des structures sur E, de type V plus fines que s, et $\mathcal{P}_V^-(s, E)$ l'ensemble des structures sur E, de type V moins fines que s.

Alors $\mathcal{P}_V^+(s, E)$ possède, pour la relation $\succcurlyeq_{pf}$ un élément minimal noté s_V^+ et $\mathcal{P}_V^-(s, E)$ possède, pour cette même relation un élément maximal noté s_V^-.

Proposition 5.1. *Encadrement d'une structure s par deux structures de type V*

– *Pour tout x de E, la famille $\mathcal{V}^+(x)$ des prévoisinages de x pour s_V^+ est le préfiltre de base $\mathcal{V}(x)$. Pour cette structure $s_V^+ = (i_V^+, a_V^+)$, on a, pour toute partie A de E :*

$$i_v^+(A) = \bigcup_{B \subset A} i(B) \text{ et } a_v^+(A) = \bigcap_{B \supset A} a(B)$$

– *pour tout x de E, la famille $\mathcal{V}^-(x)$ des prévoisinages de x pour s_V^- est le plus grand (au sens de l'inclusion ensembliste) préfiltre contenu dans $\mathcal{V}(x)$. Pour cette structure $s_V^- = (i_V^-, a_V^-)$, on a, pour toute partie A de E :*

$$i_v^-(A) = \bigcap_{B \supset A} i(B) \text{ et } a_v^-(A) = \bigcup_{B \subset A} a(B)$$

Preuve :

Pour tout x de E, $\mathcal{V}^+(x)$ est le filtre le moins fin contenant $\mathcal{V}(x)$. On a alors pour tout x de E, x appartient à $i_V^+(A)$ si et seulement si A appartient à $\mathcal{V}^+(x)$; comme $\mathcal{V}^+(x)$ admet $\mathcal{V}(x)$ comme base, ceci est encore équivalent à dire qu'il existe B dans $\mathcal{V}(x)$ avec $B \subset A$; mais ceci équivaut à dire que x appartient à $i(B)$, avec B contenu dans A ; en tenant compte du fait que $a_V^+(A)$ est duale de $i_V^+(.)$, on en déduit la caractérisation de A pour tout A. Ce qui prouve la première affirmation.

Pour tout x de E, l'ensemble des préfiltres contenus dans $\mathcal{V}(x)$ est non vide, puisqu'il contient par exemple le préfiltre $\{E\}$. La réunion de tous ces préfiltres est aussi un préfiltre, noté $\mathcal{V}^-(x)$, qui est donc le plus grand préfiltre contenu dans $\mathcal{V}(x)$. La prétopologie qui, à tout x de E, associe la famille $\mathcal{V}^-(x)$ est donc de type V, est moins fine que s et est, de toutes les prétopologies de type V moins fines que s, la plus fines de toutes : c'est la structure s_V^-. On a alors, pour toute partie A de E, x appartient à $i_V^-(A)$ si et seulement si A appartient à $\mathcal{V}^-(x)$; par suite, pour toute partie B contenant A, on a aussi B appartient à $\mathcal{V}^-(x)$, puisque $\mathcal{V}^-(x)$ est un préfiltre, et donc à $\mathcal{V}(x)$, puisque $\mathcal{V}^-(x)$ est contenu dans $\mathcal{V}(x)$, et donc x appartient à $i(B)$. La caractérisation de $a_V^-(.)$ est obtenue par le fait que $i_V^-(.)$ et $a_V^-(.)$ sont duales. Ce qui achève la démonstration.

Remarques :

– On a donc : $s_V^+ \succcurlyeq_{pf} s \succcurlyeq_{pf} s_V^-$ et, dans aucun des deux intervalles $]s_V^-, s[$ et $]s, s_V^+[$ définis au sens de la relation $\succcurlyeq_{pf}$, il n'existe de structure de type V.
– Si $s = (i, a)$ qui n'est pas de type V, on sait (c.f. 5.2.3) qu'on peut lui associer $s^\circ = (i^\circ, a^\circ)$ de type V ; il est aisé de voir que s° n'est alors autre que s_V^+.
– Enfin, si s est de type V, alors évidemment $s = s_V^+ = s_V^-$.

5.2.6 Structure de type VDS et relation binaire

Il est facile de voir que toute structure de type **VDS** sur un ensemble E est équivalente à la donnée d'une relation binaire réflexive sur cet ensemble. En effet, si s est de type **VDS** sur E, alors pour tout x de E la famille $\mathcal{V}(x)$ des prévoisinages de x pour s est de la forme $\mathcal{V}(x) = V(x)^s$, où $V(x)$ est l'intersection de tous les prévoisinages de x pour s. On peut alors associer à s la relation binaire réflexive $\mathcal{R}$ sur E définie par : $x\mathcal{R}y$ si et seulement si $y \in V(x)$. On dira que $\mathcal{R}$ est la relation binaire " correspondante " à la structure de type **VDS**. Inversement si $\mathcal{R}$ est une relation réflexive sur E, et si on note $\mathcal{R}(x) = \{y \in E; x\mathcal{R}y\}$, alors la structure s telle que pour tout x de E, la famille des prévoisinages de x est le filtre de base $\{\mathcal{R}(x)\}$ est de type **VDS**. On dit que la structure s de type **VDS** ainsi construite est la structure "correspondante" à $\mathcal{R}$. Il est clair alors que les affirmations "$\mathcal{R}$ est correspondante à s" et "s est correspondante à $\mathcal{R}$" sont équivalentes. Par extension on dira que $\mathcal{R}$ et s sont équivalentes.

On sait qu'une relation $\mathcal{R}'$ est plus fine qu'une relation $\mathcal{R}''$ si et seulement si $\forall x \in E, \{y \in E; x\mathcal{R}'y\} \subset \{y \in E; x\mathcal{R}''y\}$. Il en résulte que si s' et s'' sont les structures de type **VDS** équivalentes à $\mathcal{R}'$ et $\mathcal{R}''$ respectivement, alors on a : $\mathcal{R}'$ est plus fine que $\mathcal{R}''$ si et seulement si s' est plus fine que s''.
Autrement dit, la relation de finesse pour les structures prétopologiques coïncide avec la relation de finesse pour les relations binaires. Si $\mathcal{R}$ est une relation réflexive et symétrique, alors on dira que la structure s qui lui est équivalente est de type **VDSS**. Une telle structure est caractérisée par le fait que , pour tout x et tout $y, x \in \mathcal{V}(y) = \{z \in E; y\mathcal{R}z\}$ si et seulement si $y \in \mathcal{V}(x) = \{z \in E; x\mathcal{R}z\}$.
Si $\mathcal{R}$ est une relation d'équivalence, alors la structure s qui lui est équivalente est une topologie dont tous les ouverts sont aussi fermés ; les ouverts ou fermés non vides de cette topologie sont les classes d'équivalences ou des réunions de classes. Une telle topologie est encore appelée ici **"topologie-classe" :** $\mathbf{t}_d$ et $\mathbf{t}_g$ sont des topologies-classes, $\mathbf{t}_d$ est la plus fine de toutes et $\mathbf{t}_g$ la moins fines de toutes.

5.2.7 Relation réflexive et relations d'équivalences associées

Etant donnée une relation binaire réflexive $\mathcal{R}$, on peut considérer les deux ensembles $\mathcal{E}^+(\mathcal{R})$ formé de l'ensemble des relations d'équivalences plus fines que $\mathcal{R}$ et $\mathcal{E}^-(\mathcal{R})$ formé de l'ensemble des relations d'équivalences moins fines que $\mathcal{R}$.

Etant donnée une relation binaire réflexive $\mathcal{R}$ sur E, rechercher une classification

"proche de $\mathcal{R}$" reviendra :

- Soit à chercher une relation d'équivalence $\mathcal{E}$ telle que si $\mathcal{E}$ est plus fine que $\mathcal{R}$, et si $\mathcal{E}'$ est une autre relation d'équivalence plus fine que $\mathcal{R}$ et moins fine que $\mathcal{E}$, alors $\mathcal{E} = \mathcal{E}'$: une telle relation $\mathcal{E}$ n'est autre qu'un élément minimal de $\mathcal{E}^+(\mathcal{R})$ pour la relation de finesse entre relations.
- Soit à chercher une relation d'équivalence $\mathcal{E}$ telle que si $\mathcal{E}$ est moins fine que $\mathcal{R}$, et si $\mathcal{E}'$ est une autre relation d'équivalence moins fine que $\mathcal{R}$ et plus fine que $\mathcal{E}$, alors $\mathcal{E} = \mathcal{E}'$: une telle relation $\mathcal{E}$ n'est autre qu'un élément maximal de $\mathcal{E}^-(\mathcal{R})$ pour la relation de finesse entre relations.

  La recherche de ces éléments est un problème bien connu. Nous rappelons ici la solution à chacun de ces deux problèmes

Proposition 5.2. *Eléments minimaux de $\mathcal{E}^+(\mathcal{R})$*

L'ensemble $\mathcal{E}^+(\mathcal{R})$ comporte un ou plusieurs éléments minimaux qui sont alors les relations d'équivalence les moins fines possibles parmi toutes celles qui sont plus fines que $\mathcal{R}$.

Tout élément minimal est obtenu en effectuant les deux étapes suivantes :

1. *étape 1 : Construction de $\mathcal{R}_S$, relation réflexive et symétrique déduite de $\mathcal{R}$ par :*

$$\forall x \in E, \forall y \in E, x\,\mathcal{R}_S\,y \Leftrightarrow x\mathcal{R}y \text{ et } y\mathcal{R}x$$

2. *étape 2 : Construction de $\mathcal{R}_{ST}$, relation d'équivalence plus fine que $\mathcal{R}_S$ et telle que si C est une classe d'équivalence pour cette relation alors*

$$x \in E, x \notin C \Leftrightarrow \exists y \in C; non(x\,\mathcal{R}_S\,y)$$

Preuve

Notons $\mathcal{E}_m^+(\mathcal{R})$ l'ensemble des éléments minimaux de $\mathcal{E}^+(\mathcal{R})$ et $\Phi(\mathcal{R})$ l'ensemble des relations d'équivalence sur E construites par l'exécution des deux étapes ci-dessus.
On a $\Phi(\mathcal{R}) \subset \mathcal{E}_m^+(\mathcal{R})$.
Il est clair que, par construction on a $\Phi(\mathcal{R}) \subset \mathcal{E}^+(\mathcal{R})$.
S'il existait $\mathcal{E} \in \Phi(\mathcal{R})$ et $\mathcal{E} \notin \mathcal{E}_m^+(\mathcal{R})$ alors, il existerait $\mathcal{E}'$ dans $\mathcal{E}_m^+(\mathcal{R})$ avec $\mathcal{E}$ strictement plus fine que $\mathcal{E}'$ et $\mathcal{E}'$ plus fine que $\mathcal{R}$. Il existerait donc C' sous ensemble de E qui serait classe d'équivalence pour $\mathcal{E}'$ qui contiendrait au moins deux classes A et B pour $\mathcal{E}$. Pour x dans A, il existerait donc y dans B tel que l'on ait $non(x\mathcal{R}_S y)$. Mais, comme x et y sont dans C', on aurait aussi $x\mathcal{E}'y$ et par suite on aurai aussi $x\mathcal{R}y$ et $y\mathcal{R}x$ soit encore $x\,\mathcal{R}_S\,y$,

ce qui est contradictoire.
On a $\mathcal{E}_m^+(\mathcal{R}) \subset \Phi(\mathcal{R})$.
Soit $\mathcal{E} \in \mathcal{E}_m^+(\mathcal{R})$ et soit C une classe d'équivalence pour E. on a forcément $C \neq E$, car $C = E$ alors c'est que $\mathcal{E} = \mathcal{R}$ est la relation "grossière" et donc que trivialement $\mathcal{E}_m^+(\mathcal{R}) = \Phi(\mathcal{R}) = \{\mathcal{E}\}$. Puisque $C \neq E$, alors il existe $x \notin C$; notons H la classe de x pour E. Si pour tout y de C on avait $x\,\mathcal{R}_S\,y$ alors on aurait aussi $z\,\mathcal{R}_S\,y$ pour tout z dans H et y dans C. Considérons alors la relation $\mathcal{E}'$ définie par :

$$\forall z \in E - (H \cup C), \forall t \in E - (H \cup C), z\mathcal{E}'t \text{ si et seulement si } z\mathcal{E}t$$

$$\forall z \in E - (H \cup C), \forall t \in (H \cup C), non(z\mathcal{E}'t)$$

$$\forall z \in (H \cup C), \forall t \in (H \cup C), z\mathcal{E}'t$$

Cette relation d'équivalence n'est autre que la relation ayant exactement les mêmes classes que $\mathcal{E}$, à l'exception de la classe $H \cup C$ obtenue par réunion de H et de C.
Elle est plus fine que $\mathcal{R}$ et strictement moins fine que $\mathcal{E}$: ceci contredirait le fait que $\mathcal{E} \in \mathcal{E}_m^+(\mathcal{R})$. C'est donc que pour tout $x \notin C$, il existe y dans C tel que $non(x\mathcal{R}_S y)$: par suite $\mathcal{E} \in \Phi(\mathcal{R})$.

Remarque :
On sait que si $\mathcal{R}$ est une relation réflexive sur E, alors elle est caractéristique de la structure de type **VDS** pour laquelle, pour tout x de E, la famille des prévoisinages de x est la famille $\mathcal{R}(x)^s$, avec $\mathcal{R}(x) = \{y \in E; x\mathcal{R}y\}$. Si, en plus d'être réflexive, $\mathcal{R}$ est symétrique, on a alors la propriété suivante :

$$\forall x \in E, \forall y \in E, y \in \mathcal{R}(x) \Leftrightarrow x \in \mathcal{R}(y)$$

La structure associée à une telle relation pourra être appelée structure de type **VDSS**.

Proposition 5.3. *Plus grand éléments de $\mathcal{E}^-(\mathcal{R})$*

L'ensemble $\mathcal{E}^-(\mathcal{R})$ comporte un plus grand éléments qui est la relation d'équivalence la plus fine possible parmi toutes celles qui sont moins fines que $\mathcal{R}$.
Ce plus grand élément est obtenu en effectuant les deux étapes suivantes :

1. *étape 1 : Construction de $\mathcal{R}_S$, relation réflexive et symétrique déduite de $\mathcal{R}$ par :*

$$\forall x \in E, \forall y \in E, x\,\mathcal{R}_S\,y \Leftrightarrow x\mathcal{R}y \text{ ou } y\mathcal{R}x$$

2. *étape 2 : Construction de $\mathcal{R}_{ST}$, relation d'équivalence moins fine que $\mathcal{R}_S$ définie*

par :

$$\forall x \in E, \forall y \in E, x\,\mathcal{R}_{ST}\,y \Leftrightarrow \exists z; x \in E, x\mathcal{R}z \text{ et } z\mathcal{R}y$$

Preuve :

Evidente.

Rappelons que $\mathcal{R}_S$ définie ci-dessus est la plus fine des relations réflexives symétriques moins fines que $\mathcal{R}$ et la relation $\mathcal{R}_{ST}$ construite ci-dessus est encore appelée la "fermeture transitive de $\mathcal{R}_S$".

5.2.8 Encadrement de s par deux structures de type VDS (cas E fini)

Soit $s = (i, a)$ une structure quelconque sur E que l'on supposera fini (on verra en remarque le problème posé par le cas E non fini).

Pour tout x de E, $\mathcal{V}(x) = \{V \subset E; x \in i(V)\}$, désigne la famille des prévoisinages de x pour cette structure.

On rappelle que :

- $\mathcal{P}_V^+(s, E)$ désigne l'ensemble des structures sur E, de type V plus fines que s,
- $\mathcal{P}_V^-(s, E)$ désigne l'ensemble des structures sur E, de type V moins fines que s.

On a vu que $\mathcal{P}_V^+(s, E)$ possède, pour la relation $\succcurlyeq_{pf}$, un élément minimal noté s_V^+ et que $\mathcal{P}_V^-(s, E)$ possède, pour cette même relation un élément maximal noté s_V^-.

Rappelons que, pour tout x de E, $\mathcal{V}^+(x)$ et $\mathcal{V}^-(x)$ désignent les familles de prévoisinages de x pour s_V^+ et s_V^- respectivement.

On note :

- $\mathcal{P}_{VDS}^+(s, E)$ l'ensemble des structures de type **VDS** sur E, plus fines que s.
- $\mathcal{P}_{VDS}^-(s, E)$ l'ensemble des structures de type **VDS** sur E, moins fines que s.

Proposition 5.4. *Encadrement de s par deux structures de type VDS*

- *$\mathcal{P}_{VDS}^+(s, E)$ possède, pour la relation $\succcurlyeq_{pf}$ un élément minimal noté s_{VDS}^+ et, pour tout x de E, la base de prévoisinages $\mathcal{V}_{VDS}^+(x)$ pour cette structure s_{VDS}^+ est le filtre stable $V^+(x)^s$ de base $\{V^+(x)\}$, $V^+(x)$ étant l'intersection de tous les V appartenant à $\mathcal{V}^+(x)$*
- *$\mathcal{P}_{VDS}^-(s, E)$ possède, pour la relation $\succcurlyeq_{pf}$ des éléments maximaux. Si on note s_{VDS}^- l'un de ces éléments maximaux, alors pour tout x de E, la base de prévoisinages $\mathcal{V}_{VDS}^-(x)$ de x pour cette structure s_{VDS}^- est un filtre du type $W(x)^s$ de base de base $W(x)$, où $W(x)$ est un élément minimal, pour l'inclusion, de l'ensemble $\mathcal{V}(x)$.*

Preuve :

Pour ce qui est de la première affirmation, il suffit de remarquer que, pour tout $x \in E$, le plus petit filtre stable contenant $\mathcal{V}(x)$ est le préfiltre $V^+(x)^s$ où $V^+(x)$ est l'intersection de tous les V appartenant à $V^+(x)$.

Pour ce qui est de la seconde affirmation, il suffit de chercher quels sont, pour tout $x \in E$, les plus grand préfiltres stables contenus dans $\mathcal{V}(x)$. Pour cela, il faut rechercher les éléments minimaux de $\mathcal{V}(x)$. De tels éléments existent.

Si V est élément de $\mathcal{V}(x)$ on détermine l'ensemble $\mathcal{V}_m(V) = \{W \in \mathcal{V}(x); W \subset V\}$. Deux cas de figure sont possibles :

- Cas 1 : $\mathcal{V}_m(V) = \{V\}$: dans ce cas V est élément minimal de $\mathcal{V}(x)$.
- Cas 2 : il existe $V' \in \mathcal{V}_m(V)$ tel que V' est strictement contenu dans V. Dans ce cas, on recommence le processus avec $\mathcal{V}_m(V')$. Comme E est fini, ce processus se termine obligatoirement par le cas 1.

Soit $\mathcal{V}_{min}(x) \subset \mathcal{V}(x)$ l'ensemble des éléments minimaux de $\mathcal{V}(x)$. Alors on vérifie sans difficulté que tout préfiltre de la forme W^s où W est élément de $\mathcal{V}_{min}(x)$ est, d'une part un préfiltre stable contenu dans $\mathcal{V}(x)$, et d'autre part un élément maximal dans l'ensemble des préfiltres stables contenus dans $\mathcal{V}(x)$.

On vérifie enfin que toute prétopologie de type **VDS** telle que, pour tout $x \in E$, la famille des prévoisinages de x soit de la forme $W(x)^s$ avec $W(x) \in \mathcal{V}_{min}(x)$ est élément maximal de $\mathcal{P}^-_{VDS}(s, E)$.

Remarque :

Dans le cas où E n'est pas fini, l'existence d'éléments maximaux de $\mathcal{P}^-_{VDS}(s, E)$ n'est pas assurée, comme le montre l'exemple suivant.

Soit $E = \mathbb{Q}$ l'ensemble des rationnels. On considère la structure s de type V telle que, pour tout $x \in E$ la base de prévoisinages de x est la famille

$$\mathcal{B}(x) = \{[x - 1/n, x], [x, x + 1/n]; n \in \mathbb{N} - \{0\}\}$$

Cette structure s est de type V. La famille s^-_{VDS} est formée de toutes les structures qui ont, pour prévoisinage de tout x de E, un filtre propre de la forme $[x - 1/n, x]^s$ ou de la forme $[x, x + 1/n]^s$. Il est clair que cette famille n'a pas d'éléments maximaux. On vérifie que la famille des majorants (au sens de la relation $\succcurlyeq_{pf}$) de cette famille est réduite à l'ensemble $\{\mathbf{t}_d\}$ qui est ainsi la borne supérieure de $\mathcal{P}^-_{VDS}(s, E)$, mais qui n'est pas dans $P^-_{VDS}(s, E)$ et qui, par ailleurs est strictement plus fine que s.

On trouvera, dans le schéma (Fig. 5.2), la présentation synthétique de l'ensemble des résultats.

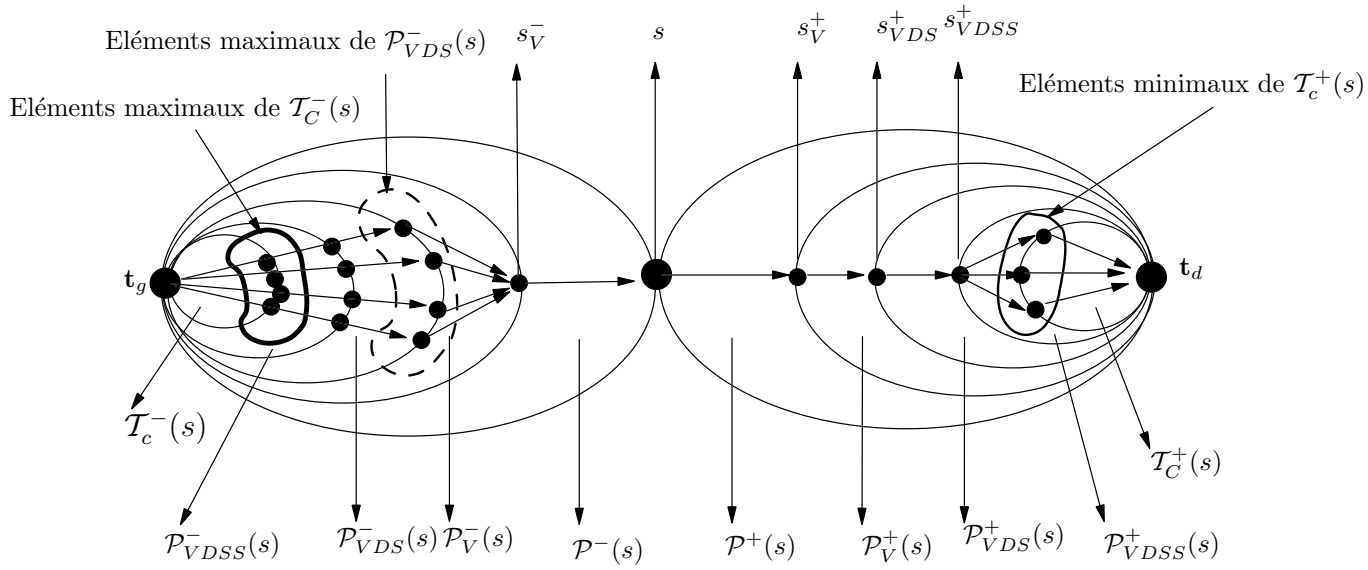

FIG. 5.2 – Présentation synthétique de l'ensemble des résultats

5.2.9 Concept de continuité

Soient E et E' deux ensembles, et $f(.)$ une application de E dans E'.
Si A est une partie de E, on désigne par $f(A)$ la partie de E' définie par :

$$f(A) = \{f(x)\ ; x \in A\}$$

et si A' est une partie de E', on désigne par $f^{-1}(A')$ la partie de E définie par :
$f^{-1}(A') = \{x \in A\ ; f(x) \in A'\}$.
On rappelle qu'on a les propriétés suivantes :

- (App01) $f(\emptyset) = \emptyset$
- (App02) $\forall A \subset E, \forall B \subset E, A \subset B \Rightarrow f(A) \subset f(B)$
- (App03) $\forall A \subset E, \forall B \subset E, f(A \cup B) = f(A) \cup f(B)$
- (App04) $\forall A \subset E, \forall B \subset E, f(A \cap B) \subset f(A) \cap f(B)$
- (App05) f surjective $\Leftrightarrow f(E) = E'$
- (App06) f surjective $\Leftrightarrow \quad \forall A \subset E, (f(A))^c \quad \subset f(A^c)$
- (App07) f injective $\Leftrightarrow \quad \forall A \subset E, \forall B \subset E, f(A \cap B) = f(A) \cap f(B)$
- (App08) f injective $\Leftrightarrow \quad \forall A \subset E, f(A^c) \subset (f(A))^c$
- (App09) f bijective $\Leftrightarrow \quad \forall A \subset E, (f(A))^c = f(A^c)$
- (App10) $f^{-1}(\emptyset) = \emptyset$
- (App11) $f^{-1}(E') = E$
- (App12) $\forall A' \subset E', f^{-1}(A'^c) = (f^{-1}(A'))^c$

– (App13) $\forall A' \subset E', \forall B' \subset E', A' \subset B' \Rightarrow f^{-1}(A') \subset f^{-1}(B')$
– (App14) $\forall A' \subset E', \forall B' \subset E', f^{-1}(A' \cup B') = f^{-1}(A') \cup f^{-1}(B')$
– (App15) $\forall A' \subset E', \forall B' \subset E', f^{-1}(A' \cap B') = f^{-1}(A') \cap f^{-1}(B')$
– (App16) $\forall A' \subset E', f(f^{-1}(A')) \subset A'$
– (App17) f surjective $\Leftrightarrow \quad \forall A' \subset E', f(f^{-1}(A')) = A'$
– (App18) $\forall A \subset E, f^{-1}(f(A)) \supset A$

Soient E muni d'une structure prétopologique $s = (i, a)$ et E' muni d'une structure prétopologique $s' = (i', a')$ et soit $f(.)$ une application de E dans E'.

Les relations entre s, s' et f sont régies par le concept de continuité.

Soit x un élément de E et $f(x)$ son image par f dans E'. La familles des prévoisinages de x est $\mathcal{V}(x)$ et celle des prévoisinages de $f(x)$ est $\mathcal{V}'(f(x))$.

On dit que $f(.)$ **est continue en** x si l'affirmation suivante est vraie :
– (C1) $\forall W' \in \mathcal{V}'(f(x)), f^{-1}(W') \in \mathcal{V}(x)$.
On montre que (C1) implique (C2) :
– (C2) $\forall W' \in \mathcal{V}'(f(x)), \exists V \in \mathcal{V}(x), f(V) \subset W'$.
et que si s est de type V (C2) implique (C1).

Soit E muni de s et E' muni de s', et soit $f(.)$ continue en x ; alors $f(.)$ est aussi continue en x pour toutes structures s^* et s'_* lorsque s^* est plus fine que s et s'_* moins fine que s'.

On dit que $f(.)$ **est continue sur** E si et seulement si elle est continue en tout x de E.

On considère les deux affirmations suivantes :
– (C3) $f(.)$ est continue sur E
– (C4) $\forall A \subset E, f(a(A)) \subset a'(f(A))$

On démontre que, si s est de type V, alors (C3) $\Rightarrow$ (C4), et que si s' est de type V alors (C4) $\Rightarrow C3$ Il en résulte que si s <u>et</u> s' sont de type V alors (C3) et (C4) sont équivalentes.

Ainsi la continuité est-elle le concept permettant de " transporter " d'un ensemble à un autre via une application une structure prétopologique : en d'autres termes, la continuité est pour la prétopologie (et donc la topologie) l'équivalent de ce qu'est un " morphisme " pour ce qui est du transport de structures algébriques.

Une des façons commodes d'utiliser ce concept est de chercher sur E ou sur E' des conditions " a minima " permettant de rendre $f(.)$ continue.

Il y a alors deux types de problèmes.

Premier type de problèmes : si $f(.)$ est une application de E dans E' et si E est muni d'une structure prétopologique s, on cherchera à munir E' d'une structure s' qui rende $f(.)$ continue sur E. De nombreuses structures s' conviennent et, en particulier

$s = \mathbf{t}_g$. Evidemment cette solution “ triviale ” et extrême n'est pas très intéressante car elle est bien trop grossière et traduit mal la proximité (“ tout objet de E' est proche de tout autre objet de E'”), et il y a d'autres structures s' plus fines qui pourraient convenir ; en revanche, en général, la solution extrême opposée $s' = \mathbf{t}_d$, ne convient pas non plus, car s' est bien trop fine pour être compatible avec s via $f(.)$. La solution est alors de chercher, si elle existe, une structure s' qui soit un élément maximal dans l'ensemble des structures sur E' rendant $f(.)$ continue.

Second type de problème : si $f(.)$ est une application de E dans E' et si E' est muni d'une structure prétopologique s', on cherchera à munir E d'une structure s qui rende $f(.)$ continue sur E. De nombreuses structures s conviennent, et en particulier $s = \mathbf{t}_d$. Mais cette solution triviale et extrême n'est pas intéressante car elle est bien trop fine et traduit mal la proximité (“ un objet quelconque de E n'est proche d'un autre objet de E”) et il y a d'autres structures s moins fines qui pourraient convenir. La solution est alors de chercher, si elle existe, une structure s qui soit minimale dans l'ensemble des structures sur E rendant $f(.)$ continue. Si une telle structure existe, on l'appelle ” prétopologie induite sur E via f”

Nous verrons plus loin comment la résolution de ce type de problème peut être d'un grand secours pour l'élaboration de processus de classification en cas de données discrètes. Nous aurons besoin, dans cette tentative de résolution, de faire éventuellement appel au concept d'espace prétopologique produit dont nous donnons la définition plus loin. Ce ne sont pas les problèmes du premier type qui nous intéresseront ici, mais ceux du second type qui pourront nous être utiles pour la recherche de classifications. On peut, dans le cas de ce second type de problèmes chercher quelles sont, en fonction de s', la caractérisation et les propriétés de la structure s. On établit alors la propriété suivante.

5.2.10 Prétopologie induite (second type de problème)

Soient E et E' deux ensembles non vides et $f(.)$ application de E dans E'.
On suppose que E' est muni d'une structure $s' = (i', a')$ avec $a' = c \circ i' \circ c$; pour cette structure, la famille des prévoisinages de $x' \in E'$ est $\mathcal{V}'(x') = \{V' \in E'; x' \in i'(V')\}$.
On appelle ”structure (prétopologique) induite de s' sur E via $f(.)$ la structure $s = (i, a)$ ainsi définie :

– pour tout x de E, la famille $\mathcal{V}(x)$ des prévoisinages de x est :

$$\mathcal{V}(x) = \{f^{-1}(V'); V' \in \mathcal{V}'(f(x))\}$$

– pour toute partie A de E, $i(A) = \{x \in E; A \in \mathcal{V}(x)\}$
– $a = c \circ i \circ c$

Si s' n'est pas de type V, s ne l'est généralement pas non plus. Même si s' est de type V, s peut fort bien ne pas être de type V, comme le montre l'exemple suivant :
$E = \{x, y, z, t\}$; $E' = \{x', y', z'\}$; $f(x) = x', f(y) = y', f(z) = f(t) = z'$; s' est telle que $\mathcal{V}'(x') = \{\{x', y'\}, E'\}$; par suite $\mathcal{V}(x) = \{\{x, y\}, E\}$ qui n'est pas un préfiltre. Ce n'est que dans le cas très particulier où $f(.)$ est une bijection que l'on peut être sûr que s est de type V lorsque s' l'est. On établit la propriété suivante qui résulte immédiatement de la définition de s.

Propriété de la prétopologie induite

Soient E et E' deux ensembles non vides.
Soit $s' = (i', a')$ une structure prétopologique sur E'.
Soit $f(.)$ une application de E dans E'.
Soit $s = (i, a)$ la structure prétopologique induite par s' sur E, via $f(.)$.
Soit $\mathcal{P}_c(E, f(.))$ l'ensemble de toutes les prétopologies sur E rendant $f(.)$ continue.
Alors s est le plus petit élément (au sens de la relation $\succcurlyeq_{pf}$)de $\mathcal{P}_c(E, f(.))$.
En d'autres termes, s est la moins fine des structures rendant $f(.)$ continue.

Remarques :

1. En général, comme on l'a dit plus haut, s n'est pas de type V, même si s' l'est. Pour des raisons de commodité, il sera souvent utile de chercher la moins fine des structures de type V rendant $f(.)$ continue. Or on sait qu'il existe sur E deux structures de type V, s_V^+ et s_V^- qui "encadrent" s au sens de la relation $\succcurlyeq_{pf}$, c'est-à-dire telles que $s_V^+ \succcurlyeq_{pf} s \succcurlyeq_{pf} s_V^-$: il en résulte que s_V^+ est alors la moins fine des structures de type V rendant $f(.)$ continue.
2. Bien entendu, si s est elle-même de type V on a $s = s_V^- = s_V^+$.

5.3 Application à un problème de classification

On considère ici un ensemble E et une structure $s = (i, a)$ définie sur E. On se propose alors de déterminer un moyen d'obtenir une ou plusieurs classifications qu'on dira " issues de s ".
Dans cette partie, on supposera que E est fini, en raison du contexte dans lequel se fera

l'usage des résultats. Toutefois, il est clair qu'un certain nombre des propos tenus dans cet paragraphe restent vrais en contexte non fini.
Dans un paragraphe préalable, nous avons défini les structures de type **VDS** et montré que de telles structures étaient caractéristiques d'une relation binaire. Par ailleurs, nous avons aussi évoqué les structures de type **VDSS**, comme caractéristiques des relations binaires symétriques. Nous devons compléter ce tour d'horizon entre structures et relation binaire en établissement un lien entre relation d'équivalence et une structure particulière : celle de topologie-calsse.

5.3.1 (Pré)topologie et partition

Une classification de E est caractérisées par la donnée d'une partition P de E :

$$P = \{P_1, \ldots, P_k, \ldots, P_m\}$$

où les P_k constituent une famille de parties non vides, deux à deux disjointes de E, et dont la réunion est égale à E. Un sous ensemble P_k est appelé "élément de la partition P" ou "classe pour la partition" ; m est le nombre de classes de la partition.
Si $x \in E$, il existe k unique entier entre 1 et m tel que $x \in P_k$: on dit que P_k est la "classe de x" et on la notera encore $Cl(x)$; on a donc alors $Cl(x) = P_k$. Tous les éléments de P_k ont donc la même classe qui est P_k.
On peut alors considérer la structure prétopologique $s_{\mathbf{P}}$ définie ainsi :

$$\forall x \in E, \mathcal{B}(x) = \{Cl(x)\}$$

Il en résulte que les voisinages de x sont tous les sur-ensembles de $Cl(x)$.
Il est facile de vérifier que $s_{\mathbf{P}}$ est une topologie tout à fait particulière. Pour cette topologie, en effet, l'adhérence d'une partie A est la réunion des $Cl(x)$ pour tous les x dans A ; l'intérieur de A est l'ensemble des x de A tels que $Cl(x)$ est contenu dans A ; les seuls ensembles fermés (non vides) sont des réunions de $Cl(x)$ et tout fermé est aussi ouvert et réciproquement.
Si on considère la relation binaire $\mathcal{E}$ définie par :

$$\forall x \in E, \forall y \in E, x\mathcal{E}y \Leftrightarrow y \in Cl(x)$$

$\mathcal{E}$ est une relation d'équivalence dont les $Cl(x)$ sont les classes.
Si on considère le graphe représentatif de cette relation d'équivalence, alors les classes sont les composantes connexes du graphe, appelées encore "cliques maximales".

Dans la suite, on appellera **"topologie-classe"** toute structure (pré)topologique définie telle que $s_{\mathbf{P}}$.
Le problème que nous cherchons à résoudre est donc le suivant :
Connaissant une structure s sur E comment "associer", par un moyen adéquat à définir, une (voire plusieurs) topologie-classe.
Pour cela, nous allons associer à s deux familles de topologies-classes.

5.3.2 Familles de topologies-classes associées à s et éléments remarquables

Soit s une structure prétopologique sur E.
On note $\mathcal{T}_C$ la famille de toutes les topologies-classes sur E.
On désigne par

$$\mathcal{T}_C^+(s) = \{t \in \mathcal{T}_C; t \succcurlyeq_{pf} s\}$$

et

$$\mathcal{T}_C^-(s) = \{t \in \mathcal{T}_C; s \succcurlyeq_{pf} t\}$$

Il est clair que $\mathbf{t}_d \in \mathcal{T}_C^+(s)$ et que $\mathbf{t}_g \in \mathcal{T}_C^-(s)$.
On s'intéressera alors :

- aux éléments minimaux de $\mathcal{T}_C^+(s)$ pour la relation $\succcurlyeq_{pf}$. On notera $Min(\mathcal{T}_C^+(s))$ l'ensemble de ces éléments minimaux.
- aux éléments maximaux de $\mathcal{T}_C^-(s)$ pour la relation $\succcurlyeq_{pf}$. On notera $Max(\mathcal{T}_C^-(s))$ l'ensemble de ces éléments maximaux.

Ces éléments présentent, comme intérêt, d'être "pas trop éloignés" de la structure s qui sert de référence, au sens de la relation de finesse.
De façon, plus précise, si s est une structure prétopologique quelconque sur E, si t^+ est un élément de $Min(\mathcal{T}_C^+(s))$ et t^- un élément de $Max(\mathcal{T}_C^-(s))$, alors on a :

- $t^+ \succcurlyeq_{pf} s \succcurlyeq_{pf} t^-$
- Si t est un élément de $\mathcal{T}_C$ tel que $t^+ \succcurlyeq_{pf} t \succcurlyeq_{pf} s$: alors $t = t^+$
- Si t est un élément de $\mathcal{T}_C$ tel que $s \succcurlyeq_{pf} t \succcurlyeq_{pf} t^-$: alors $t = t^-$.

5.3.3 Processus de construction de $\mathcal{T}_C^+(s)$ et de $\mathcal{T}_C^-(s)$

On se propose ici de décrire les différentes étapes de la construction des deux ensembles $Max(\mathcal{T}_C^-(s))$ et $Min(\mathcal{T}_C^+(s))$.

Soit s prétopologie sur E. On fera appel aux notations suivantes, dont certaines ont déjà été utilisées :

- $\mathcal{S}^-(s)$ est l'ensemble des structures prétopologiques moins fines que s ; s est le plus grand élément (au sens de $\succcurlyeq_{pf}$) de $\mathcal{S}^-(s)$.
- $\mathcal{S}^+(s)$ est l'ensemble des structures prétopologiques plus fines que s ; s est le plus petit élément de $\mathcal{S}^+(s)$.
- $\mathcal{S}_V^-(s)$ est l'ensemble des structures prétopologiques de type V, moins fines que s ; s_V^- est le plus grand élément de $\mathcal{S}_V^-(s)$.
- $\mathcal{S}_V^+(s)$ est l'ensemble des structures prétopologiques de type V, plus fines que s ; s_V^+ est le plus petit élément de $\mathcal{S}_V^+(s)$.
- $\mathcal{S}_{VDS}^-(s)$ est l'ensemble des structures prétopologiques de type VDS, moins fines que s. On sait que cet ensemble possède des éléments maximaux (5.2.8). Soit $Max(\mathcal{S}_{VDS}^-(s))$ cet ensemble d'éléments maximaux.
- $\mathcal{S}_{VDS}^+(s)$ est l'ensemble des structures prétopologiques de type VDS, plus fines que s ; s_{VDS}^+ est le plus petit élément de $\mathcal{S}_{VDS}^-(s)$.

On rappelle que si une structure prétopologique sur E est de type VDS, elle est équivalente à la donnée d'une relation réflexive $\mathcal{R}$ sur E. Pour une telle structure chaque élément $x \in E$ admet comme base de voisinage le préfiltre stable $V(x)^s$ où $V(x)$ désigne l'intersection de tous les éléments de $V(x)$. Inversement si on considère la relation $\mathcal{R}$ sur E définie par $x\mathcal{R}y$ si et seulement si $y \in V(x)$ alors $\mathcal{R}$ est réflexive.
Une structure de type VDS est donc caractéristique d'une relation réflexive.
Si une structure de type VDS est caractéristique d'une relation réflexive et symétrique, elle est de type VDSS. Une telle structure est caractérisée par la fait que, pour tous éléments x et y de E, alors $x \in V(y)$ si et seulement si $y \in V(x)$. On notera alors :

- $\mathcal{P}_{VDSS}^-(s)$ est l'ensemble des structures prétopologiques de type VDSS, moins fines que s
- $\mathcal{P}_{VDSS}^+(s)$ est l'ensemble des structures prétopologiques de type VDSS, plus fines que s.

On a alors les propositions suivantes.

Proposition 5.5. *Comparaison des divers ensembles de prétopologies*

On a les propriétés suivantes :

1. $\Phi \neq \mathcal{T}_C^+(s) \subset \mathcal{P}_{VDSS}^+(s) \subset \mathcal{P}_{VDS}^+(s) \subset \mathcal{P}_V^+(s) \subset \mathcal{P}^+(s)$
2. $\Phi \neq \mathcal{T}_C^-(s) \subset \mathcal{P}_{VDSS}^-(s) \subset \mathcal{P}_{VDS}^-(s) \subset \mathcal{P}_V^-(s) \subset \mathcal{P}^-(s)$
3. $\mathcal{P}^+(s) \cap \mathcal{P}^-(s) = \{s\}$
4. $\mathcal{P}_{VDS}^+(s)$ *possède un plus petit élément noté* s_{VDS}^+. *Cette structure est caractéristique*

d'une relation binaire réflexive notée $\mathcal{R}^+(s)$. Toute relation d'équivalence plus fine que $\mathcal{R}^+(s)$ est caractéristique d'un élément de $\mathcal{T}_C^+(s)$.
L'ensemble $\mathcal{T}_C^+(s)$ n'est autre que l'ensemble $\mathcal{E}^+(\mathcal{R}^+(s))$ tel qu'il a été défini en 5.2.7
Cet ensemble possède des éléments minimaux qui sont les éléments de l'ensemble $Min(\mathcal{T}_C^+(s))$.

5. *L'ensemble $\mathcal{P}_{VDS}^-(s)$ possède un nombre fini m d'éléments maximaux ; on note $Max(\mathcal{P}_{VDS}^-(s))$ cet ensemble. Tout élément de $Max(\mathcal{P}_{VDS}^-(s))$ est caractéristique d'une relation binaire réflexive. Soit $s_k, k = 1, \ldots, m$ un tel élément et $\mathcal{R}_k(s)$ la relation qui lui est équivalente et soit $\mathcal{E}^-(\mathcal{R}_k(s))$ l'ensemble des relations d'équivalence moins fines que $\mathcal{R}_k(s)$. Alors $\mathcal{E}^-(\mathcal{R}_k(s))$ possède un plus grand élément $t_k^-(s)$.*
L'ensemble $\{t_k^-(s); k = 1, \ldots, m\}$ est l'ensemble $Max(\mathcal{T}_C^-(s))$.

Preuve :

1 et 2 On a évidemment $\mathbf{t}_d \in \mathcal{T}_C^+(s)$ et $\mathbf{t}_g \in \mathcal{T}_C^-(s)$; toutes les inclusions résultent évidemment des définitions des ensembles considérés.
3 Evident.
4 Résulte de (5.2.7). et du fait que les topologies-classe plus fines que s sont exactement les topologies-classe plus fines que s_{VDS}^+.
5 Résulte de (5.2.7). et du fait que toute topologie-classe moins fine que s est aussi moins fine que s_V^-.

5.3.4 Produit d'espaces prétopologique

On se limitera ici au produit cartésien d'un nombre fini d'espaces prétopologiques. Mais on trouvera dans Belmandt [Bel93] la généralisation au cas infini.

Soient $E_1, E_2, \ldots, E_p$, p ensembles non vides.

On suppose que chacun de ces ensembles E_j $(j = 1, \ldots, p)$ est muni d'une structure prétopologique $s_j = (i_j, a_j)$. Pour tout élément x_{jk_j} de E_j, on note $\mathcal{V}_j(x_{jk_j})$ la famille de ses prévoisinages ; on a donc pour toute partie V de E_j, $V \in \mathcal{V}_j(x_{jk_j})$ si et seulement si $x_j \in i_j(V)$.

On pose $E = E_1 \times E_2 \times \ldots, E_j \times \ldots, \times E_p$

Il existe deux types de produits : le produit " ordinaire ", et le produit "en croix ".

Considérons $e = (e_{1k_1}, e_{2k_2}, e_{3k_3}, \ldots, e_{jk_j}, \ldots, e_{pk_p})$ un élément de E.

Dans le cas **du produit ordinaire**, la famille $\mathcal{V}(e)$ des voisinages de e a pour base $\mathcal{B}(e)$ la famille des ensembles de la forme B suivante :

B $\in \mathcal{B}(e)$ si et seulement si

$B = V_1 \times V_2 \times \ldots \times V_k \times \ldots \times V_p$ où, pour tout $k = 1, \ldots, p$, V_k est un élément de $\mathcal{V}(e_{jk_j})$.

Dans le cas du produit en croix, la famille $\mathcal{V}(e)$ des voisinages de e a pour base $\mathcal{B}(e)$ la famille des ensembles de la forme B suivante :

$B \in \mathcal{B}(e)$ si et seulement si

$B = (V_1 \times \{e_{2j_2}\} \times \ldots \times \{e_{kj_k}\} \times \ldots \times \{e_{pj_p}\}) \cup (\{e_{1j_1}\} \times V_2 \times \{e_{3j_3}\} \times \ldots \times \{e_{pj_p}\}) \cup \ldots \cup (\{e_{1j_1}\} \times \{e_{2j_2}\} \times \ldots \times V_{p-1} \times \{e_{pj_p}\}) \cup (\{e_{1j_1}\} \times \{e_{2j_2}\} \times \ldots \times \{e_{pj_p}\} \times V_p)$

où, pour tout $k = 1, \ldots p$, V_k est un élément de $\mathcal{V}(e_{jk_j})$.

Le produit ordinaire n'est autre que la généralisation, au cas de la prétopologie, du produit topologique classique. En revanche le produit en croix est une spécificité de la prétopologie.

Rappelons que, si on considère deux structures prétopologiques de type V sur un même ensemble E la première est plus fine que la seconde si et seulement si, pour tout x de E, la famille des prévoisinages de x pour la première contient la famille des prévoisinages de x pour la seconde.

Il en résulte que le produit en croix sur E induit sur E une structure plus fine que le produit ordinaire.

5.3.5 Application à la classification

On se propose ici de construire une méthode de classification sur l'ensemble $\Omega = \{\omega_1, \ldots, \omega_n\}$ de n objets décrits par leurs caractéristiques relativement à p critères qualitatifs. Le principe de base de la classification repose de façon fondamentale sur le concept de "proximité" : deux objets de Ω seront "mis dans une même classe" si et seulement si ils sont "suffisamment proches" l'un de l'autre, et cette notion de proximité relèvera tout naturellement du concept de prétopologie : on cherchera donc à construire sur Ω une structure prétopologique "ad hoc".

Pour construire cette structure, on utilisera logiquement chacune des applications $X_j(.)$ de Ω dans E_j, où $j \in \{1, \ldots, p\}$, pour laquelle $X_j(\omega_i) = e_{jk_i}$ est la modalité observée par ω_i pour le critère j.

La méthodologie que nous proposons est basée sur le fait que "l'expert" a une idée "a priori" des proximités qu'il considère comme naturelles entre les diverses modalités d'un même critère j. Cette proximité se traduira alors par une structure prétopologique sur E_j ; nous pouvons alors en déduire une structure sur $E = \prod_{j=1}^{p} E_j$ puis, en utilisant des propriétés de

continuité des applications $X_j(.)$, une structure prétopologique sur l'ensemble Ω, structure dont on déduira, in fine, une - ou plusieurs - classifications sur Ω.
On est alors amené à résoudre trois problèmes successifs :

- En premier lieu, de quelle prétopologie munir E, traduisant les sentiments de l'expert en matière de proximités des diverses modalités ?
- En second lieu, quelle(s) structure(s) en déduire sur Ω ?
- Enfin, comment en déduire une classification sur Ω ?

Nous nous proposons de donner ici des éléments de réponses à ces trois problèmes.

Construction d'une structure sur E

Cette construction passe par celle préalable d'une structure sur chacun des ensembles E_j de modalités des critères. Deux cas de figure sont possibles, suivant que E_j possède ou non une structure d'ordre nature, c'est-à-dire suivant que la variable $X_j(.)$ est ordonnée ou non. Mais dans ces deux cas, on cherchera à associer à chaque modalité $e_{jk} \in E_j$ une famille $\mathcal{V}(e_{jk})$ de prévoisinages.

Cas où E_j est muni d'une relation d'ordre $<_j$

Supposons que l'ensemble $E_j = \{e_{j1}, e_{j2}, \ldots, e_{jk}, \ldots, e_{jn_j}\}$ soit tel que :

$$e_{j1} <_j e_{j2} <_j \ldots <_j e_{jk_{-1}} <_j e_{jk} <_j e_{jk_{+1}} <_j \ldots <_j e_{jn_j}$$

alors on peut imaginer que, pour l'expert, e_{j1} "est proche de e_{j2}", "e_{j2} est proche de e_{j1} et de e_{j3}" etc. ou toute autre proximité de ce genre : sachant que, en prétopologie, un objet x est "d'autant plus proche de l'objet y que x appartient à un plus grand nombre de prévoisinages de y" et que "x est plus proche de y que z ne l'est de y, si le nombre de voisinages de y auxquels x appartient est plus grand que le nombre de voisinages auxquels z appartient". C'est en tenant compte de ces principes que l'expert construira sur chaque E_j une structure prétopologique de type V traduisant sa proximité sur les modalités.

Cas où E_j n'est pas muni d'une structure d'ordre

Dans le cas où E_j n'a pas de structure particulière, l'expert pourra là encore considérer comme "proches" telles ou telles modalités et tenir compte de cet avis pour la construction de ses prévoisinages. Au pire si, pour un caractère particulier - et ce sera probablement le cas pour des ensembles E_j de la forme $E_j = \{Oui, Non\}$ - il considérait qu'aucune modalité n'était "proche" d'une autre, il pourrait traduire cela en mettant sur cet ensemble la structure (pré)topologique discrète.

On trouvera ci-dessous un exemple montrant comment la prétopologie peut permettre de traduire les idées "a priori" de l'expert sur les proximités entre modalités.

Exemple1 :

On suppose que $E_j = \{e_{j1}, e_{j2}\}$: situation où les modalités sont de type ("oui","non"). Dans cette situation ; ni e_{j1} proche de e_{j2} ni e_{j2} proche de e_{j1}. Alors on prend $B_{j1} = \{e_{j1}\}$ et $B_{j2} = \{e_{j2}\}$.

Exemple2 :

On suppose que $E_j = \{e_{j1}, e_{j2}, e_{j3}\}$:

1. On pense que e_{j1}, e_{j2}, e_{j3} sont "équivalents", alors on prend :
   $B_{j1} = \{\{e_{j1}, e_{j2}\}, \{e_{j1}, e_{j3}\}\}$.
   $B_{j2} = \{\{e_{j2}, e_{j1}\}, \{e_{j2}, e_{j3}\}\}$.
   $B_{j3} = \{\{e_{j3}, e_{j1}\}, \{e_{j3}, e_{j2}\}\}$.
   Comme dans le figure 5.3

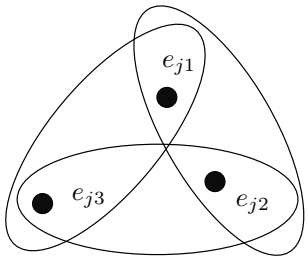

FIG. 5.3 – Exemple 1 de bases des voisinages

2. On pense que e_{j1} est plus proche de e_{j2} que de e_{j3} et
   e_{j2} est aussi proche de e_{j1} que e_{j1}
   e_{j3} est plus proche de e_{j2} que de e_{j1}
   alors on prend :
   $B_{j1} = \{e_{j1}, e_{j2}\}$.
   $B_{j2} = \{e_{j1}, e_{j2}, e_{j3}\}$.
   $B_{j3} = \{e_{j3}, e_{j2}\}$.
   Comme dans le figure 5.4

Désormais, on supposera que chaque E_j est muni d'une structure prétopologique qui est parfaitement déterminée par la donnée, pour tout élément e_{jk} de E_j, de la famille $\mathcal{V}(e_{jk})$ des prévoisinages de e_j. Et, dans la suite, on supposera que pour tout j et tout k, $\mathcal{V}(e_{jk})$ est au minimum un préfiltre de parties contenant e_{jk}. Chaque E_j étant muni d'une

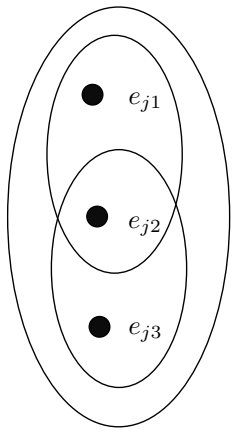

FIG. 5.4 – Exemple 2 de bases des voisinages

structure prétopologique, il sera alors possible d'en construire une sur le produit E des E_j : $E = E_1 \times E_2 \times \ldots \times E_p$, qu'il s'agisse du produit ordinaire ou du produit en croix. Dans tous les cas de figure, cette structure de type V traduit les "proximités" que l'expert perçoit sur les proximités. Dans la suite, on notera s_P la structure retenue sur E.

Construction d'une structure sur Ω

On dispose alors d'une application $X(.)$ de l'ensemble Ω des objets à classifier dans l'ensemble E, application qui à chaque élément ω_i de Ω associe

$$X(\omega_i) = (X_1(\omega_i), X_2(\omega_i), \ldots, X_j(\omega_i), \ldots, X_p(\omega_i)) \in E = \prod_{j=1}^{p} E_j$$

Il semble alors naturel de construire sur une structure prétopologique qui tiendra compte à la fois de $X(.)$ et de la structure prétopologique adoptée sur E, elle-même directement construite des diverses structures sur E_j, telle que définie plus haut.
Sachant que E est muni de la structure s_P et que la continuité est l'outil de transport des structures de proximité, on cherchera à construire sur Ω une structure qui rende $X(.)$ continue : cette construction relève du second type de problèmes évoqués en 5.2. On cherchera donc à munir de la structure prétopologique la moins fine rendant $X(.)$ continue.
Cette structure existe et n'est autre que la structure induite de s_P sur Ω via $X(.)$.
Cette prétopologie induite n'est peut-être pas celle qui servira de structure à partir de laquelle sera construite une classification sur Ω, mais elle nous servira de base pour la construction de cette structure de référence.
En effet, ayant construit s_P on en déduit les deux familles $\mathcal{T}_C^-(s_P)$ et $\mathcal{T}_C^+(s_P)$ de topologies-

classes qui "encadrent" s_P. Ces deux familles pourront servir de référence, en ce sens que toute classification qui serait plus fine qu'un élément de $\mathcal{T}_C^+(s_P)$ ou moins fine qu'un élément de $\mathcal{T}_C^-(s_P)$ devrait être rejetée.
Plus généralement, toute proposition de classification devrait être comparée avec celles de $\mathcal{T}_C^+(s_P)$ ou de $\mathcal{T}_C^-(s_P)$.
On peut en effet imaginer que la structure à partir de laquelle on proposera la construction d'une classification ne soit pas s_P elle-même, mais une autre structure construite "directement" sans référence directe à un quelconque problème de continuité : le "classificateur" pourra laisser libre cours à son imagination, il pourra toujours "contrôler" son résultat en le comparant aux éléments de $\mathcal{T}_C^+(s_P)$ ou de $\mathcal{T}_C^-(s_P)$.

Exemple
On considère un ensemble $\Omega = \{\omega_1, \ldots, \omega_n\}$ et p applications $X_1(.), \ldots, X_p(.)$ de Ω dans les ensembles respectifs $E_1, E_2, \ldots, E_p$. Ces p applications peuvent être considérées comme p variables statistiques permettant de donner les caractéristiques des objets de Ω relativement à p critères ; pour tout caractère $k = 1, \ldots, p$ on note $\{e_{k1}, e_{k2}, \ldots, e_{km_k}\}$ l'ensemble des modalités du caractère qui possède donc au plus m_k modalités.

Pour tout critère $k, k = 1, \ldots, p$, la famille :

$$\{X_k^{-1}(e_{kj})\,;\, j = 1, \ldots, m_k\}$$

constitue une partition de Ω.
A tout ω_i , on peut associer la famille de sous ensembles de Ω :

$$\{X_k^{-1}(X_k(\omega_i)\,;\, k = 1, \ldots, m\}$$

Cette famille peut alors être utilisée par construire une base de prévoisinages associée à ω_i.

On peut, par exemple associer à ω_i l'ensemble

$$\bigcap_{k=1}^{m} X_k^{-1}(X_k(\omega_i))$$

Mais on peut aussi imaginer d'autres façons de procéder. Par exemple, on peut fixer un entier q, avec $1 \leq q \leq p$ et associer à tout ω_i de Ω, l'ensemble $B(\omega_i, q)$ contenant l'ensemble de tous les éléments de Ω qui figurent dans au moins q des p sous-ensembles $X_k^{-1}(X_k(\omega_i))$.

Cette façon de procéder peut aussi être utilisable lorsqu'un système de pondération est associé aux critères ; notons α_k(α_k, réel strictement positif) le poids du critère k, (On suppose que $\alpha_1 + \alpha_2 + \ldots + \alpha_m = 1$) et enfin β un réel compris entre 0 et 1. On pourra associer à tout ω_i de Ω, l'ensemble $B(\omega_i, \beta)$ contenant tout élément ω_t de Ω tels que la somme des poids des critères tels que $X_k^{-1}(X_k(\omega_i))$ contient ω_t est supérieure ou égale à β.
On trouvera ci-après un exemple illustrant cette méthodologie et une application dans le chapitre (7).

Exemple 1

On considère $\Omega = \{\omega_1, \omega_2, \ldots, \omega_8\}$ et trois variables X_1, X_2 et X_3 définies par le tableau suivant :

	X_1	X_2	X_3
ω_1	o	a	1
ω_2	n	b	2
ω_3	o	c	3
ω_4	n	c	2
ω_5	o	b	1
ω_6	o	a	3
ω_7	o	a	1
ω_8	o	a	3
	E_1	E_2	E_3

Les trois ensembles de modalités E_1, E_2 et E_3 sont :
$E_1 = \{o, n\}$
$E_2 = \{a, b, c\}$
$E_3 = \{1, 2, 3\}$

Nous en déduisons :
$X_1^{-1}(o) = \{\omega_1, \omega_3, \omega_5, \omega_6, \omega_7, \omega_8\}$
$X_1^{-1}(n) = \{\omega_2, \omega_4\}$

$X_2^{-1}(a) = \{\omega_1, \omega_6, \omega_7, \omega_8\}$
$X_2^{-1}(b) = \{\omega_2, \omega_5\}$
$X_2^{-1}(c) = \{\omega_3, \omega_4\}$

$X_3^{-1}(1) = \{\omega_1, \omega_5, \omega_7\}$
$X_3^{-1}(2) = \{\omega_2, \omega_4\}$
$X_3^{-1}(3) = \{\omega_3, \omega_6, \omega_8\}$

Pour chaque $\omega_i \in \Omega$ on se propose de construire une base de voisinage ; $B(\omega_i, q)$ en choisissant dans un premier temps $q = 3$.

Première étape :
Nous trouvons :
$\omega_1 = \{o, a, 1\} \Rightarrow B(\omega_1, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(a) \cap X_3^{-1}(1)\} = \{\omega_1, \omega_7\}$
$\omega_2 = \{n, b, 2\} \Rightarrow B(\omega_2, 3) = \{X_1^{-1}(n) \cap X_2^{-1}(b) \cap X_3^{-1}(2)\} = \{\omega_2\}$
$\omega_3 = \{o, c, 3\} \Rightarrow B(\omega_3, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(c) \cap X_3^{-1}(3)\} = \{\omega_3\}$
$\omega_4 = \{n, c, 2\} \Rightarrow B(\omega_4, 3) = \{X_1^{-1}(n) \cap X_2^{-1}(c) \cap X_3^{-1}(2)\} = \{\omega_4\}$
$\omega_5 = \{o, b, 1\} \Rightarrow B(\omega_5, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(b) \cap X_3^{-1}(1)\} = \{\omega_5\}$
$\omega_6 = \{o, a, 3\} \Rightarrow B(\omega_6, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(a) \cap X_3^{-1}(3)\} = \{\omega_6, \omega_8\}$
$\omega_7 = \{o, a, 1\} \Rightarrow B(\omega_7, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(a) \cap X_3^{-1}(1)\} = \{\omega_1, \omega_7\}$
$\omega_8 = \{o, a, 3\} \Rightarrow B(\omega_8, 3) = \{X_1^{-1}(o) \cap X_2^{-1}(a) \cap X_3^{-1}(3)\} = \{\omega_6, \omega_8\}$

Nous proposons alors de construire sur Ω la relation R définie par :

$$\forall \omega_i \in \Omega\,; \omega_i R \omega_j \Leftrightarrow \omega_j \in B(\omega_i, q)$$

Un telle relation est donnée dans le (Fig. 5.7)

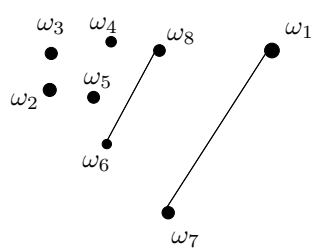

FIG. 5.5 – Graphe des relations entre les individus - cas q=3

On en déduit la partition suivante sur Ω en classes d'équivalence comme le figure 5.6

Deuxième étape
Pour chaque $\omega_i \in \Omega$ on se propose de construire une base de voisinage ; $B(\omega_i, q)$ en choisissant $q = 2$. On trouve :
$\omega_1 = \{o, a, 1\} \Rightarrow B(\omega_1, 2) = \{\omega_1, \omega_5, \omega_6, \omega_7, \omega_8\}$
$\omega_2 = \{n, b, 2\} \Rightarrow B(\omega_2, 2) = \{\omega_2, \omega_4\}$

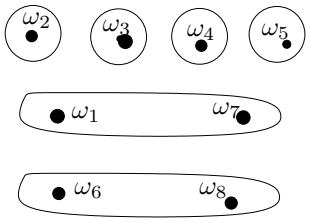

FIG. 5.6 – Partition donnée dans le cas q=3

$\omega_3 = \{o, c, 3\} \Rightarrow B(\omega_3, 2) = \{\omega_3, \omega_6, \omega_8\}$
$\omega_4 = \{n, c, 2\} \Rightarrow B(\omega_4, 2) = \{\omega_2, \omega_4\}$
$\omega_5 = \{o, b, 1\} \Rightarrow B(\omega_5, 2) = \{\omega_1, \omega_5, \omega_7\}$
$\omega_6 = \{o, a, 3\} \Rightarrow B(\omega_6, 2) = \{\omega_1, \omega_3, \omega_6, \omega_7, \omega_8\}$
$\omega_7 = \{o, a, 1\} \Rightarrow B(\omega_7, 2) = \{\omega_1, \omega_5, \omega_6, \omega_7, \omega_8\}$
$\omega_8 = \{o, a, 3\} \Rightarrow B(\omega_8, 2) = \{\omega_1, \omega_3, \omega_6, \omega_7, \omega_8\}$

Il en résulte comme précédemment une relation dont le graphe est donné par 5.7

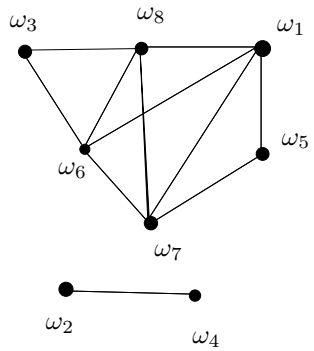

FIG. 5.7 – Graphe des relations entre les individus - cas q=2

Nous en déduisons, par la recherche de cliques maximales les deux solutions suivantes données par (Fig. 7.1)

5.4 Algorithme de la méthode

Nous donnons ci-après le schéma de la procédure :
Algorithme Approche prétopologique

`Données :` $W[n, p]$ `un tableau des données , q où (`$\frac{p}{2} + 1 < q < p$`)`
`Résultat :` $\mathcal{P}$ `une partition du tableau` W

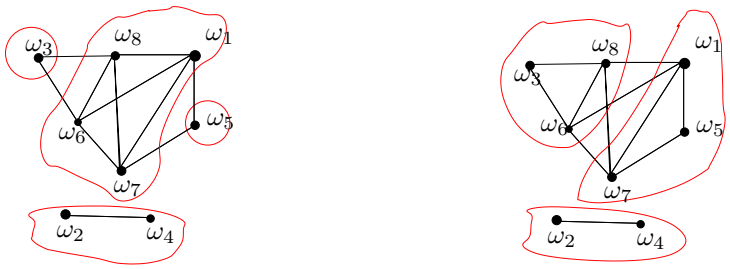

FIG. 5.8 – Les Cliques Maximales trouvés dans ce graphe

```
Begin
    Pour j :=1 à p faire
            Trouver l'ensemble des modalités E_j pour chaque variable X_j
            E_j = {e_ji | e_ji  est une modalité pour la variable X_j}
    Fin Pour

    Pour j :=1 à p faire
        Pour e_ji ∈ E_j faire
            Trouver les individus qui ont la modalité e_ji
            X_j^{-1}(e_ij) = {ω_i | X_j(ω_i) = e_ij}
        Fin Pour
    Fin Pour

    Pour i :=1 à n faire
        Construire les Bases des Voisinages B(w_i, q) pour les individus w_ij
        B(w_i, q) = {∩X_k^{-1}(X_k(w_i))} k = 1...,p
    Fin Pour

    Pour i :=1 à n faire
        Pour j :=1 à n faire
        Construire le graphe G des relations entre les individus :
            Si  w_i ∈ B(w_j, q)  donc :
                Faire un ligne entre les deux noeuds w_i et w_j
        Fin Pour (j)
    Fin Pour (i)

    i=1
```

```
    Répéter
        Trouver dans G un Clique Maximale
        Une classe Cl_i = {Le clique maximale trouver dans G}
            Supprimer les individus formant Cl_i du graphe G
    i=i+1
    Jusqu'à plus de Clique Maximale dans G

    P_i = {Cl_i | Cl_i Un Clique maximale en G }

end
```

Remarques :

- Dans cette méthode, nous ne pouvons pas fixer le nombre de classes à l'avance.
- Le nombre des classes liée au nombre des modalité par variable et de nombre de variables.
- Nous pouvons prendre dans le deuxième cas le nombre des éléments les plus répétés dans les bases des voisinages égale à $(p-1), (p-2), \ldots,$ jusqu'au le temps que nous trouvons un nombre satisfait des classes.

5.4.1 Conclusion

Nous avons présenté dans ce document les éléments de base pour une première approche du concept de prétopologie. Nous avons ensuite montré de quelle manière la prétopologie pouvait être utilisée pour élaborer une méthodologie de classification faisant appel aux concepts de prévoisinages, et applicable dans un contexte très général où les éléments devant faire l'objet de la classification sont décrits par des caractéristiques purement qualitatives ; ainsi donc, il n'a nullement été besoin de recourir à une métrique plus ou moins artificielle pour apprécier la "proximité" entre deux objets. Il ne s'agit là que d'une application parmi de nombreuses possibles. On trouvera dans [Bel93] la présentation d'autres notions telles que celles de compacité ou de connexité et d'autres applications, en particulier dans le domaine de l'analyse d'images ou de la formation des coalitions en théorie des jeux, problème qui est, formellement, très proche de celui de la classification que nous avons présenté ici.

Chapitre 6

Approche de la classification par l'agrégation des préférences

Sommaire

6.1 Introduction : 132
6.2 Position du problème 132
6.3 Etude de la structure des familles de variables décisives . . . 136
6.4 Etude spécifique de la famille $\mathcal{D}_{(2,Ag)}$ 142
6.5 Hypothèse de finesse 146
6.6 Conclusion 151

6.1 Introduction :

Dans ce chapitre, nous proposons d'analyser le problème de la classification d'une population à partir de la mesure de k variables quantitatives X_i sur ses individus à la manière des modèles d'agrégation des préférences. L'objectif n'est donc pas de développer une méthode originale de classification mais plutôt d'interpréter le problème de fond de la classification par les concepts de l'agrégation des préférences et de produire des résultats qui permettent de qualifier la qualité d'une procédure de classification et, peut être d'introduire dans cette procédure une part d'intervention humaine permettant d'en contrôler le déroulement.

L'idée qui sous-tend cette approche s'appuie sur la constatation qu'il est en général aisé de classifier une population à partir de la donnée des observations d'une seule variable. En effet, les diverses méthodes habituellement utilisées de classification fournissent, la plupart du temps, des résultats quasiment identiques lorsqu'on les applique dans des situations où n'intervient qu'une seule variable. Il est alors naturel de se dire que lorsque k variables $X_1, X_2, \ldots, X_k$ sont en jeu, on obtient k classifications associées à chacune de ces variables et qu'obtenir une classification finale revient à agréger ces k classifications en une seule. On se ramène ainsi à un problème d'agrégation. Les travaux dédiés aux problèmes d'aide à la décision apportent de nombreux résultats à ce sujet, notamment les travaux situés dans la lignée de ceux menés par **K. J. Arrow** [Arr63]. Nous nous situerons dans cette lignée tout en notant que le principal résultat est qu'il n'existe pas de méthode d'agrégation, mais des méthodes d'agrégation. C'est la justification même de ce paragraphe : partant du même constat qu'il n'existe pas une méthode de classification mais des méthodes de classification, nous souhaitons produire une réflexion s'appuyant sur le parallélisme qui existe entre les deux problèmes : classification et agrégation pour obtenir des résultats qui peuvent éclairer, sur le fond, l'utilisateur des méthodes de classification dans sa démarche et l'aider à se poser des questions.

6.2 Position du problème

Nous désignons par Ω la population que l'on cherche à classifier. $\mathcal{X}, \mathcal{X} = \{X_1, X_2, \ldots, X_k\}$ désigne l'ensemble des variables définies sur Ω. Pour des raisons de simplicité dans l'exposé, nous supposons que toutes les variables sont quantitatives, cela ne changerait cependant rien au discours si certaines de ces variables étaient des variables qualitatives. Chaque variable X_i est associée à une fonction P_i dite fonction d'agrégation et définie de la manière

suivante :

Définition 6.1. *Fonction de classement élémentaire P_i*
$\forall i, i = 1, 2, \ldots, k$. $P_i : \Omega \times \Omega \to E = \{1, 2\}$ *telle que* $P_i(x, y) = 1$ *si et seulement si* x *et* y *sont dans la même classe définie par* X_i *et* $P_i(x, y) = 2$ *sinon.*

Exemple 6.1. $\mathcal{X} = \{X_1, X_2, X_3\}$, $\Omega = \{a, b, c, d, e\}$.
La partition engendrée par X_1 *est* $\{\{a, b\}, \{c, d, e\}\}$, *la partition engendrée par* X_2 *est* $\{\{a, b\}, \{c, d\}, \{e\}\}$ *et la partition engendrée par* X_3 *est* $\{\{a, b, c\}, \{d, e\}\}$. *Les fonctions de classification associées sont :*

P_1	a	b	c	d	e
a	1	1	2	2	2
b	1	1	2	2	2
c	2	2	1	1	1
d	2	2	1	1	1
e	2	2	1	1	1

P_2	a	b	c	d	e
a	1	1	2	2	2
b	1	1	2	2	2
c	2	2	1	1	2
d	2	2	1	1	2
e	2	2	2	2	1

P_3	a	b	c	d	e
a	1	1	1	2	2
b	1	1	1	2	2
c	1	1	1	2	2
d	2	2	2	1	1
e	2	2	2	1	1

De manière générale, nous désignerons par $\mathfrak{P}$ la famille de toutes les applications de $\Omega \times \Omega$ dans E, et E est appelé l'ensemble des "codes".
Si nous connaissons les fonctions d'agrégation P_i associées à chacune des variables X_i, nous avons une information sur la fonction de classification de la population des variables, prises individuellement. Bien évidemment, ces fonctions P_i sont entièrement déterminées par la méthode de classification qui est utilisée avec la variable X_i : changer de méthode de classification amène à changer de fonction P_i.
Cela nous amène à poser, de manière générale, la définition suivante.

Définition 6.2. *Fonction de classification*

Nous appelons "fonction de classification" toute application $P : \Omega \times \Omega \times \mathfrak{P}^0(\mathcal{X}) \to E = \{1, 2\}$ *qui traduit cette information,* $\mathfrak{P}^0(\mathcal{X})$ *désignant la famille des parties non vides de* $\mathcal{X}$*.*

Une fonction de classification correspond donc au procédé qui permet de classifier une population à partir d'un ensemble non vide de variables mesurées sur cette population. Les applications P_i définies précédemment sont des cas particuliers de fonctions de classification correspondant à la situation où on n'utilise qu'une seule variable. Pour cette raison, on pourra les appeler "fonctions élémentaires de classification".
Si nous revenons au problème initial, il est clair que proposer une classification de la population à partir de k variables X_i revient à agréger les k fonctions élémentaires de classification P_i en une fonction de classification P. Ceci suppose l'existence d'un procédé d'agrégation, d'où la définition suivante.

Définition 6.3. *Fonction d'agrégation :*
On appelle fonction d'agrégation toute fonction Ag *qui associe aux fonctions de classification élémentaires* P_i *une fonction de classification que nous noterons* AgP*.*

Ainsi, la fonction d'agrégation Ag est une application qui, à toute famille $\{P_i\}_{i=1,\ldots,k}$ de fonctions élémentaires de classification, associe une fonction de classification notée AgP.
Le propos dans la suite de ce chapitre ne sera pas de proposer un procédé de construction de cette fonction Ag, mais plutôt d'examiner les propriétés qu'un tel procédé d'agrégation devrait ou pourrait posséder, à partir desquelles il serait possible soit de contrôler le procédé, soit d'obtenir des résultats à priori sur la classification obtenue.
Nous appellerons "agrégation" la procédure qui prend en charge les fonctions élémentaires de classification et fournit une fonction de classification associée à la population des variables. Nous désignons par $\mathfrak{G}$ la famille de toutes les fonctions d'agrégation, c'est-à-dire l'ensemble des procédés qui permettent de définir une classification globale de la population à partir des classifications obtenues variable par variable.
Ceci étant posé, il est évident que les fonctions P_i correspondent en fait à des relations d'équivalence. Elles possèdent donc certaines propriétés que nous allons traduire en propriété de symétrie et propriété de transitivité.

Définition 6.4. *Propriété de symétrie*
$\forall i, i = 1, 2, \ldots, k$, *la fonction* P_i *possède la table de symétrie* S_i *suivante :*

S_i	a	$S_i(a)$
	1	*1*
	2	*2*

Ceci traduit la propriété de symétrie de la relation d'équivalence.

Définition 6.5. *Propriété de transitivité*
$\forall i, i = 1, 2, \ldots, k$, *la fonction* P_i *possède la table de de transitivité* T_i *suivante :*

	T_i	$P_i(y,z)$	
		1	*2*
$P_i(x,y)$	*1*	$\{1\}$	$\{2\}$
	2	$\{2\}$	$\{1,2\}$

Cette table de transitivité T_i est une multiapplication de E^2 dans E (application de E dans $\mathfrak{P}(E)$) qui traduit le fait que :

- Si x et y sont dans la même classe, si y et z sont la même classe, nécessairement x et z sont dans la même classe,
- Si x et y sont dans la même classe et y et z dans deux classes différentes, nécessairement x et z sont dans deux classes différentes, ou l'inverse,
- Si x et y sont dans deux classes différentes, si y et z sont dans deux classes différentes, à priori on ne peut rien dire quant à x et z : ils peuvent être dans deux classes différentes comme appartenir à la même classe.

Bien évidemment, il existe des conditions de compatibilité entre les tables de symétrie et les tables de transitivité. Pour des raisons de simplicité dans l'écriture, l'indice i est omis dans les lignes qui suivent.
En particulier, si on note pour α et β dans $E = \{1, 2\}$, $S^*(\alpha, \beta) = (S(\alpha), S(\beta))$, on peut déterminer TS^*. Plusieurs cas sont à considérer :
$TS^*(1,1) = T(S(1), S(1)) = T(1,1) = 1$
$TS^*(1,2) = T(S(1), S(2)) = T(1,2) = 2$
$TS^*(2,1) = T(S(2), S(1)) = T(2,1) = 2$
$TS^*(2,2) = T(S(2), S(2)) = T(2,2) = \{1,2\} = E$
Parallèlement, déterminons ST :
$ST(1,1) = S(1) = 1$
$ST(1,2) = S(2) = 2$
$ST(2,1) = S(2) = 2$
$ST(2,2) = S(\{1,2\}) = \{1,2\} = E$
Il vient donc une première propriété.

Lemme 6.1.
$TS^* = ST$

Posons $i(\alpha,\beta) = \alpha$ et $j(\alpha,\beta) = \beta$ pour tout α et tout β dans E et calculons $T(T(\alpha,\beta), S(i(\alpha,\beta))$.
$T(T(1,1), S(i(1,1)) = T(1,1) = \{1\} \ni j(1,1)$
$T(T(1,2), S(i(1,2)) = T(2,1) = \{2\} \ni j(1,2)$
$T(T(2,1), S(i(2,1)) = T(2,2) = \{1,2\} \ni j(1,2)$
$T(T(2,2), S(i(2,2)) = T(2,2) = \{1,2\} \ni j(1,2)$
On obtient donc :

Lemme 6.2.
$\forall \alpha, \alpha \in E, \forall \beta, \beta \in E,\ j(\alpha,\beta) \in T(T(\alpha,\beta), S(i(\alpha,\beta))$
On en déduit en particulier le résultat suivant.

Lemme 6.3.
- $\forall \alpha, \alpha \in E,\ ST(\alpha, S(\alpha)) = T(S(\alpha), \alpha)$
- $\forall \alpha, \alpha \in E, \alpha \in T(T(\alpha, S(\alpha), \alpha)$

Remarque : dans le cas plus général où on ne travaille pas avec une relation d'équivalence, ce dernier résultat n'implique pas nécessairement que $T(\alpha,\beta) = T(\beta,\alpha)$.

6.3 Etude de la structure des familles de variables décisives

Jusqu'à présent nous avons traduit différemment le processus de classification en nous ramenant à l'agrégation de classifications élémentaires. C'est précisément l'analyse des propriétés que peut permettre cette approche qui a de l'intérêt. En particulier, contrairement à ce qui se fait dans les méthodes classiques, l'utilisateur peut intervenir au cours du processus. Il peut notamment décider que certaines variables ont plus d'importance que d'autres, ce qui nous amène à poser le concept de variables décisives.

Définition 6.6. *Ensemble de variables décisives*
Un sous ensemble A de l'ensemble $\{X_1, X_2, \ldots, X_k\}$ des variables est dit sous ensemble de variables décisives en α, $\alpha \in E$, sur le couple (x,y) de Ω^2 pour une agrégation Ag si et seulement si :

$$\forall X_j, X_j \in A, P_j(x,y) = \alpha \Rightarrow AgP(x,y) = \alpha$$

Un sous ensemble de variables décisives est donc :
- Lié à un couple d'objets à classifier,
- Lié à une procédure d'agrégation,

– Défini pour une réponse classificatoire.

Il est constitué de variables qui "emportent la décision" sur le couple en question.
Par la suite, nous noterons $\mathcal{D}_{(\alpha,(x,y),J,Ag)}$ un tel ensemble, où J désigne le sous ensemble de $\{1, 2, \ldots, k\}$ des indices des variables X_j décisives.

Il est aisé d'imaginer que la fonction d'agrégation Ag ne doit pas être quelconque, mais doit vérifier des propriétés en liaison avec le problème à résoudre. En particulier, considérons la propriété (Hi) suivante, dite propriété d'indépendance :

Propriété 6.1. *Propriété (Hi)*
$\forall (x, y), (x, y) \in \Omega^2, \forall i, i \in I, P_i(x, y) = Q_i(x, y) = \alpha \Rightarrow AgP(x, y) = AgQ(x, y) = \alpha$

Cette propriété traduit le fait que le résultat de l'agrégation sur le couple (x, y) ne dépend pas de ce qui se passe sur les autres couples. Les fonctions P_i et Q_i fonctions élémentaires de classification sont les mêmes sur (x, y) mais peuvent être différentes sur les autres couples.
Cette propriété est importante puisqu'elle nous dit que, quel que soit le processus de classification (P_i ou Q_i) associé à la variable X_i qui est utilisé, dès que sur un coule (x, y) il fournit un résultat identique variable par variable, alors la classification finale agrégeant ces classifications par variable fournit le même résultat. Cela traduit bien une propriété d'indépendance à la manière de classifier variable par variable, ce qui nous assure que si (Hi) est vérifiée, nous n'avons pas à nous préoccuper de la méthode de classification qui est mise en œuvre variable par variable.

Lemme 6.4.

Si la fonction agrégation Ag vérifie la propriété (Hi) (propriété 6.1), $\mathcal{D}_{(\alpha,(x,y),P,Ag)} = \mathcal{D}_{(\alpha,(x,y),Q,Ag)}$, dès que P_i et Q_i sont égales pour les couples (x, y), $\forall i \in I$.

Preuve :
Soit $A, A \in \mathcal{D}_{(\alpha,(x,y),P,Ag)}$ et $P_i(x, y) = \alpha$, $\forall i \in I$. Alors $AgP(x, y) = \alpha$. Or $P_i(x, y) = Q_i(x, y), \forall i, i \in I$, donc pour la propriété (Hi), $AgP(x, y) = AgQ(x, y) = \alpha$. Donc A est tel que $Q_i(x, y) = \alpha$, $\forall i, i \in I$ avec $AgQ(x, y) = \alpha$. Ainsi $A \in \mathcal{D}_{(\alpha,(x,y),Q,Ag)}$.
■
Ce résultat nous précise donc que l'ensemble des variables décisives ne dépend pas du processus de classification utilisé, associé à chaque variable X_i, dès que la propriété (Hi) est supposée vérifiée. Nous pourrons donc, dès à présent, simplifier la notation $\mathcal{D}_{(\alpha,(x,y),P,Ag)}$ pour la remplacer par la notation $\mathcal{D}_{(\alpha,(x,y),Ag)}$.

Lemme 6.5.

Si la fonction d'agrégation Ag vérifie la propriété (Hi), la famille $\mathcal{D}_{(\alpha,(x,y),Ag)}$ a une structure de préfiltre.

Preuve :
Si $B \supset A$, si $A \in \mathcal{D}_{(\alpha,(x,y),Ag)}$, il est immédiat que si $P_i(x,y) = \alpha$ pour tout i tel que $X_i \in B$, alors $P_i(x,y) = \alpha$ pour tout i tel que $X_i \in A$ et ainsi $B \in \mathcal{D}_{(\alpha,(x,y),Ag)}$. Ce qui assure que $\mathcal{D}_{(\alpha,(x,y),Ag)}$ a une structure de préfiltre.
■

Ce résultat nous assure donc que, dès que la propriété (Hi) est assurée, tout sur ensemble de variables décisives sur (x,y) en est lui-même décisif sur (x,y) en α .

La deuxième propriété que l'on peut souhaiter imposer au processus Ag est la propriété dite d'unanimité qui s'énonce comme suit.

Propriété 6.2. *Propriété (Hu)*
$\forall \alpha \in E, \forall (x,y), (x,y) \in \Omega^2, i, i \in I, P_i(x,y) = \alpha \Rightarrow AgP(x,y) = \alpha.$

En d'autres termes, si toutes les fonctions élémentaires de classification P_i donnent le même résultat sur un couple (x,y), la classification finale, résultat du processus d'agrégation Ag donnera ce même résultat pour le couple (x,y).

Lemme 6.6.

Si la fonction agrégation Ag vérifie les hypothèses (Hi) et (Hu), $\{X_1, X_2, \ldots, X_k\} \in \mathcal{D}_{(\alpha,(x,y),Ag)}$ et ceci $\forall \alpha, \alpha \in E, \forall (x,y) \in \Omega^2$, En conséquence $\mathcal{D}_{(\alpha,(x,y),Ag)}$ n'est jamais vide et $\phi \notin \mathcal{D}_{(\alpha,(x,y),Ag)}$.

Preuve :
Immédiate.
■

Lemme 6.7.

La propriété (Hi) étant vérifiée par Ag, soient $\alpha, \alpha \in E$ et $\beta, \beta \in E, \alpha \neq \beta$, alors $\forall A \in \mathcal{D}_{(\alpha,(x,y),Ag)}, A \neq \phi$ et $\forall B \in \mathcal{D}_{(\beta,(x,y),Ag)}, B \neq \phi$, nous avons : $A \cap B \neq \phi$

Preuve :
En effet, si $A \cap B = \phi$, nous pourrions prendre P_i telle que $P_i(x,y) = \alpha$ pour tout i tel que $X_i \in A$ et $P_j(x,y) = \beta$ pour tout j tel que $X_j \in B$. Cela conduirait à une impossibilité, à savoir que $AgP(x,y) = \alpha$ et $AgP(x,y) = \beta$.
■

De manière évidente, les fonctions élémentaires de classification P_i ont toutes la même table de symétrie S, la table de symétrie d'une relation d'équivalence, et de la même manière, elles possèdent toutes la même table de transitivité T. On peut alors énoncer :

Lemme 6.8.

Si la fonction agrégation Ag vérifie les propriétés (Hi) et (Hu), la table de symétrie de la fonction de classification AgP est la même que celle de chacune des fonctions élémentaires de classification P_i.

Preuve :

Soit $P_i(x, y) = \alpha, \forall i, i \in I$. D'après la propriété (Hu), nous obtenons $AgP(x, y) = \alpha$. Il est immédiat que $P_i(y, x) = S(\alpha) \forall i, i \in I$. Cela donne donc $AgP(y, x) = S(\alpha)$. Ceci est vrai pour toute valeur α de E, pour tout couple (x, y) de Ω^2 et toute fonction d'agrégation Ag. D'où le résultat.

■

Lemme 6.9.

Si la fonction agrégation Ag vérifie les propriétés (Hi) et (Hu), nous avons :
$\mathcal{D}_{(\alpha,(x,y),Ag)} = \mathcal{D}_{(S(\alpha),(x,y),P,Ag)}$ *, pour tout $\alpha \in E$, tout $(x, y) \in \Omega^2$, tout Ag.*

Preuve :

Soit $A \in \mathcal{D}_{(\alpha,(x,y),Ag)}$ et $P_i(y, x) = S(\alpha)$ pour tout i tel que $X_i \in A$. Cela donne $P_i(x, y) = \alpha$ pour tout i tel que $X_i \in A$, donc $AgP(x, y) = \alpha$ et $AgP(x, y) = S(\alpha)$ ce qui entraîne $A \in \mathcal{D}_{(S(\alpha),(y,x),Ag)}$.
Ceci tient pour tout i et tout $\forall A \in \mathcal{D}_{(\alpha,(x,y),Ag)}$, donc $A \in \mathcal{D}_{(S(\alpha),(y,x),Ag)}$. Un résultat identique est obtenu en renversant les rôles. D'où le résultat.

■

Lemme 6.10.

Si la fonction agrégation Ag vérifie les hypothèses (Hi) et (Hu), la table de transitivité T_{Ag} de la procédure de classification AgP contient celle de chacune de des fonctions de classification élémentaires P_i.

Preuve :

Soit un triplet quelconque (x, y, z) un triplet d'éléments de Ω.
Soit P_i fonctions de classification telle que $P_i(x, y) = \alpha \forall i, i \in I$.
D'après (Hu) nous obtenons $AgP(x, y) = \alpha$. Supposons aussi que $P_i(y, z) = \beta \forall i, i \in I$.

Il est immédiat que pour un certaine variable $X_i, P_i(y,z) \in T_i(\alpha,\beta) = T(\alpha,\beta)$.
Supposons donc, c'est possible, que $P_i(x,z) = \gamma$, avec $\gamma \in T(\alpha,\beta)$, $\forall i, i \in I$.
D'après la propriété (Hu), nous avons $AgP(x,z) = \gamma$. Cela entraîne que $\gamma \in T_{AgP}(\alpha,\beta)$.
Donc $T(\alpha,\beta) \subset T_{AgP}(\alpha,\beta), \forall \alpha, \alpha \in E, \forall \beta, \beta \in E$ et $\forall (x,y,z),(x,y,z) \in \Omega^3$. ■

Nous allons à présent revenir sur la propriété (Hi). Cette hypothèse consistait à affirmer que si deux fonctions élémentaires de classification P_i et Q_i étaient égales sur un couple (x,y) de Ω^2 pour tout $i, i \in I$, alors par le procédé d'agrégation Ag, elles conduisaient à une même valeur sur (x,y) : $AgP(x,y) = AgQ(x,y)$.
Supposons maintenant que deux fonctions élémentaires de classification soient égales l'une sur un couple (x,y) de Ω^2 et l'autre sur un couple (u,v) de Ω^2 : $\forall i, i \in I, P_i(x,y) = Q_i(u,v)$.
La nouvelle propriété d'indépendance consiste à supposer que les réponses données sur les autres couples n'interviennent pas dans le résultat de classification obtenus par une même fonction Ag sur ces couples (x,y) et (u,v). On écrit donc la propriété (Hi_2) :

Propriété 6.3. *Propriété* (Hi_2)
$\forall (x,y),(x,y) \in \Omega^2, \forall (u,v),(u,v) \in \Omega^2, \forall i, i \in I, P_i(x,y) = Q_i(u,v) \Rightarrow AgP(x,y) = AgQ(u,v)$.

Nous pouvons remarquer que cette hypothèse (Hi_2) implique l'hypothèse (Hi), en faisant $(u,v) = (x,y)$, et qu'elle implique aussi le cas particulier où pour une même fonction élémentaire de classification P_i, nous avons deux couples (x,y) et (u,v) tels que $P_i(x,y) = P_i(u,v) \, \forall i, i \in I$.

Lemme 6.11.
Si les propriétés (Hi) et (Hu) sont vérifiées, alors la propriété (Hi_2) *est également vérifiée.*

Preuve :
Soit (x,y) et (u,v) deux éléments de X^2 et soit (P_i) une famille de fonctions de classifications élémentaires telle que $AgP(x,y) = \alpha$, avec $\alpha = 1$ ou $\alpha = 2$. Et soit (P'_i) une famille de fonctions de classifications élémentaires déterminée par le tableau ci-dessous.

P'_i	$i = 1, \ldots, k$	valeur de AgP'
(x,y)	$P_i = P'_i$	α, d'après (Hi)
(y,u)	1	1, d'après (Hu)
(x,u)	$P'_i(x,u) = P'_i(x,y)$	α en appliquant T
(x,v)	1	1, d'après (Hu)
(u,v)	$P'_i(u,v) = P'_i(x,y)$	α en appliquant T

Considérons (Q_i) une famille de fonctions de classifications élémentaires telle que $\forall i, i \in I, Q_i(u,v) = P'_i(u,v)$. Comme $AgP'(u,v) = \alpha$, on obtient $AgQ(u,v) = \alpha = AgP(x,y)$ pour (P_i) et (Q_i) telles que $\forall i, i \in I, P_i(x,y) = Q_i(u,v)$.
■

Lemme 6.12.

Si la fonction agrégation Ag vérifie les propriétés (Hi_2) et (Hu), $\forall (x,y) \in \Omega^2, \forall (u,v) \in \Omega^2$, nous avons $\mathcal{D}_{(\alpha,(x,y),Ag)} = \mathcal{D}_{(\alpha,(u,v),Ag)}$
Nous en concluons $\forall (x,y) \in \Omega^2, \mathcal{D}_{(\alpha,(x,y),Ag)} = \mathcal{D}_{(\alpha,Ag)}$.

Preuve :
Soit $A, A \in \mathcal{D}_(\alpha,(x,y),Ag)$. P_i et Q_i telles que $\forall i, i$ tel que $X_i \in A, P_i(x,y) = \alpha$ et $Q_i(u,v) = P_i(x,y), \forall i, i \in I$. Il est immédiat que $AgP(x,y) = \alpha$ et, d'après la propriété (Hi_2), $AgQ(u,v) = \alpha$. Donc $A \in \mathcal{D}_{(\alpha,(u,v),Ag)}$ et nous obtenons le résultat annoncé.
■

Lemme 6.13.

Si la fonction agrégation Ag vérifie les propriétés (Hi_2) et (Hu), nous avons $\mathcal{D}_{(\alpha,Ag)} = \mathcal{D}_{(S(\alpha),Ag)}$ pour tout α de E

Preuve :
Immédiate puisque $\alpha = S(\alpha)$.
■

On peut remarquer d'ores et déjà que la table de transitivité T, commune à toutes les fonctions de classification élémentaires vérifie que $\forall \beta, \beta \in E, \exists b_\beta \in E$ et $\exists a_\beta \in E$ tels que $\beta \in T(\alpha, b_\beta)$ et $\beta \in T(a_\beta, \alpha)$. En effet, $1 \in T(1,1), 1 \in T(2,2), 2 \in T(1,2)$ et $2 \in T(2,1)$, il suffit de prendre $a_\beta = b_\beta = 2$.

Nous avons démontré avec cette série de lemmes que finalement la famille $\mathcal{D}_{(\alpha,Ag)}$ a une structure de préfiltre pour peu que les propriétés introduites soient vérifiées. Peut-on démontrer qu'il s'agit en fait d'un filtre ? La réponse est positive dans le cas où $\alpha = 1$. On peut même annoncer :

Théorème 1.

Si les propriétés (Hu) et (Hi_2) sont vérifiées par la fonction d'agrégation $Ag, \mathcal{D}_{(1,Ag)}$ est un filtre propre.

Preuve :
Le résultat est obtenu en notant que $AgP(x,y) = 1$ et $AgP(y,z) = 1 \Rightarrow AgP(x,z) = 1$ d'après la table de transitivité T et en utilisant la remarque précédente.
■
En revanche, ce résultat ne tient pas pour $\mathcal{D}_{(2,Ag)}$. On le constate à partir de l'exemple ci-dessous.
$X = \{x, y, z\}, \mathcal{X} = \{X_1, X_2, X_3\}$ avec P_1, P_2 et P_3 définies par :

P_i	(x,y)	(y,z)	(x,z)
P_1	1	2	2
P_2	1	1	1
P_3	2	1	2

Ce qui correspond aux trois classifications élémentaires $\{\{x,y\},\{z\}\}, \{\{x,y,z\}\}$ et $\{\{x\},\{y,z\}\}$.
Si Ag est définie comme étant la règle majoritaire (deux sur trois), on obtient $AgP(x,y) = 1, AgP(y,z) = 1$ et $AgP(x,z) = 2$. On n'a donc pas $AgP(x,y) = \alpha$ et $AgP(y,z) = \alpha \Rightarrow AgP(x,z) = \alpha$

Puisque $\mathcal{D}_{(1,Ag)}$ a une structure de filtre propre, il est intéressant de considérer $M^0 = \bigcap_{A \in \mathcal{D}_{(1,Ag)}} A$. Cet élément $M^0 \in \mathcal{D}_{(1,Ag)}$ et $M^0 \neq \phi$. Il constitue le "germe" de $\mathcal{D}_{(1,Ag)}$ et s'interprète comme le noyau minimum de variables décisives en 1, pour le procédé d'agrégation Ag.

En revanche, que peut-on dire de la famille $\mathcal{D}_{(2,Ag)}$?

6.4 Etude spécifique de la famille $\mathcal{D}_{(2,Ag)}$

Il est évident que pour cette famille, nous ne pouvons pas adopter la démarche suivie pour $\mathcal{D}_{(1,Ag)}$. Cependant, la réponse peut être donnée de manière complète.

Lemme 6.14.
Si les propriétés (Hi), (Hu) sont vérifiées, toute variable appartenant à M^0 vérifie la propriété :

$$X_i \in M^0 \quad et \quad P_i(x,y) = 2 \Rightarrow AgP(x,y) = 2$$

On dit que la variable X_i est dictatoriale en 2.

Preuve :
Soit $P_i(x,y) = 1$ pour toute variable X_i de M^0 et $P_i(x,y)$ quelconque sur le complémentaire $(M^0)^c$ de M^0. On a alors $AgP(x,y) = 1$. Si $Q_j(x,y) = 1$ pour j tel que $X_j \in M^0 - \{X_i\}$, $Q_j(x,y) = Pj(x,y)$ pour $j \in (M^0)^c$. Si $Q_i(x,y) = 1$, alors $AgQ(x,y) = 1$; si $Q_i(x,y) = 2$ et $AgQ(x,y) = 1$, cela mènerait à $(M^0 - \{X_i\}) \in \mathcal{D}_{(1,Ag)}$ et on obtient une contradiction avec la définition de M^0. En conséquence, dès que $X_i \in M^0$ et $P_i(x,y) = 2$, on a $AgP(x,y) = 2$.
■

Ce dernier résultat ne nous donne pas pour autant la preuve que $X_i \in \mathcal{D}_{(2,Ag)}$. Pour étudier ce problème, nous introduisons le concept de famille de variables **dichotomique** en 1 sur (x,y), pour la fonction d'agrégation Ag.

Définition 6.7.
*U, sous ensemble de $\mathcal{X}$, est dit famille de variables **dichotomique** en 1 sur (x,y) pour Ag si et seulement si :*
$\forall i$, tel que $X_i \in U, P_i(x,y) = 1$ et $\forall i$, tel que $X_i \in U^c, P_i(x,y) = 2 \Rightarrow AgP(x,y) = 1$.
Nous noterons $\mathcal{U}_{(1,(x,y),Ag)}$ l'ensemble de ces familles de variables dichotomiques.

Il est facile de démontrer que :
- $\mathcal{X} \in \mathcal{U}_{(1,(x,y),Ag)}$ par application de la propriété (Hu)
- Si $V \in \mathcal{U}_{(1,(x,y),Ag)}$ et si $V' \in \mathcal{U}_{(1,(x,y),Ag)}$, alors $V \cap V' = \phi$
- Si $V \in \mathcal{U}_{(1,(x,y),Ag)}$ et si $V' \in \mathcal{U}_{(1,(x,y),Ag)}$, alors $V \cap V' \in \mathcal{U}_{(1,(x,z),Ag)} \cap \mathcal{U}_{(1,(x,y),Ag)} \cap \mathcal{U}_{(1,(y,z),Ag)}$
- La famille $\mathcal{U}_{(1,(x,z),Ag)} \cap \mathcal{U}_{(1,(x,y),Ag)} \cap \mathcal{U}_{(1,(y,z),Ag)}$ est un préfiltre propre
- $\mathcal{U}_{(1,(x,z),Ag)} = \mathcal{U}_{(1,(u,v),Ag)}$ pour tout couple, donc $\mathcal{U}_{(1,(x,z),Ag)} = \mathcal{U}_{(1,Ag)}$
- $\mathcal{U}_{(1,Ag)}$ est un filtre propre
- $\mathcal{X}$ étant de cardinal fini, $U^0 = \bigcap_{U \in \mathcal{U}_{(1,Ag)}} U$ et $U^0 \in \mathcal{U}_{(1,Ag)}$
- $\mathcal{D}_{(1,Ag)} = \mathcal{U}_{(1,Ag)}$

Lemme 6.15.
Si $X_i \in M^0$ alors $\{X_i\} \in \mathcal{D}_{(2,Ag)}$.

Preuve :
Soit P telle que $P_i(x,y) = 2$ pour $X_i \in M^0$ et $P_j(x,y) = 1$ ou $P_j(x,y) = 2$ si $j \neq i$. Supposons que $AgP(x,y) = 1$ et posons $W = \{X_k / P_k(x,y) = 1\}$ (non vide) ; Alors $W \in \mathcal{U}_{(1,Ag)} = \mathcal{D}_{(1,Ag)}$, ce qui est impossible puisque pour tout V de $\mathcal{D}_{(1,Ag)}$, nous devons avoir $M^0 \subseteq V$. En conséquence $AgP(x,y) = 2$. Considérons à présent $V = \{X_k / P_k(x,y) =$

2}, donc $P_j(x,y) = 1$ pour j tel que $X_j \in V^c$. $M^0 \cap V \neq \phi$, donc $V^c \notin \mathcal{D}_{(1,Ag)}$. Si $AgP(x,y) = 1$, nous aurions $V^c \in \mathcal{U}_{(1,Ag)} = \mathcal{D}_{(1,Ag)}$. Ce qui traduit une contradiction, en conséquence, $AgP(x,y) = 2$ et $\{X_i\} \in \mathcal{D}_{(2,Ag)}$.

Lemme 6.16.

$\mathcal{D}_{(2,Ag)}$ a une structure de préfiltre.

Preuve :

Soit $V \in \mathcal{D}_{(2,Ag)}$ et $V' \supset V$. Si $P_j(x,y) = 2$ sur V', nous avons $P_j(x,y) = 2$ sur V et $AgP(x,y) = 2$.

■

Lemme 6.17.

Si $V \in \mathcal{D}_{(2,Ag)}$, alors $M^0 \cap V \neq \phi$

Preuve :

Immédiate.

■

Posons $\mathcal{S}(A)$ la famille des sur-ensembles d'une partie A. Alors :

Lemme 6.18.

$$\mathcal{D}_{(2,Ag)} = \bigcap_{X_i \in M^0} \mathcal{S}(\{X_i\})$$

Preuve :

Si $X_i \in M^0, \{X_i\} \in \mathcal{D}_{(2,Ag)}$ d'après un lemme précédent, et comme $\mathcal{D}_{(2,Ag)}$ est un préfiltre, $\mathcal{S}(\{X_i\}) \subset \mathcal{D}_{(2,Ag)}$. Par ailleurs, d'après un lemme précédent, si $X \in \mathcal{D}_{(2,Ag)}$, il existe $X_i \in M^0$ tel que $X \in \mathcal{S}(\{X_i\})$. Ce qui assure le résultat.

■

On peut alors énoncer le résultat suivant :

Théorème 2.

Soit $M^0 = \bigcap_{A \in \mathcal{D}_{(1,Ag)}} A$

- *Si $M^0 = \{X^*\}$, alors $\mathcal{D}_{(1,Ag)} = \mathcal{D}_{(2,Ag)}$,ultrafiltre propre formé de tous les sur-ensembles de $\{X^*\}$*

- *Si M^0 comprend plusieurs variables, $\mathcal{D}_{(2,Ag)}$ est la réunion des ultrafiltres propres, chacun formé de tous les sur-ensembles d'un singleton $\{X\}$, ceci pour chacun des éléments X de M^0. Ce qui s'écrit $\mathcal{D}_{(2,Ag)} = \bigcup_{X_i \in M^0} \mathcal{S}(\{X_i\})$ et $\mathcal{D}_{(2,Ag)} = \bigcap_{X_i \in M^0} \mathcal{S}(\{X_i\})$*
- *La base du filtre propre $\mathcal{D}_{(1,Ag)}$ est M^0, la base du préfiltre $\mathcal{D}_{(2,Ag)}$ est $\{\{X\}\}_{X \in M^0}$*
- $\mathcal{D}_{(1,Ag)} \subset \mathcal{D}_{(2,Ag)}$

On a ainsi caractérisé algébriquement les deux familles $\mathcal{D}_{(1,Ag)}$ et $\mathcal{D}_{(2,Ag)}$.
On peut également démontrer le résultat suivant, lié à la comparaison de partitions. On rappelle qu'une partition $\mathcal{P}$ (donc une classification) est plus fine qu'une partition $\mathcal{Q}$ si $\mathcal{P}$ est une subdivision des classes d'équivalence définies par $\mathcal{Q}$. On peut alors définir une notion de borne supérieure de partitions et de borne inférieure de partitions, par exemple $\mathcal{P} = \{\{a\}, \{b\}, \{c, d, e\}\}$ et $\mathcal{Q} = \{\{a, b\}, \{c\}, \{d, e\}\}$ sont deux partitions de $E = \{a, b, c, d, e\}$ telles que la borne supérieure de $\mathcal{P}$ et $\mathcal{Q}$ est la partition $\{\{a\}, \{b\}, \{c\}, \{d, e\}\}$ et la borne inférieure est la partition $\{\{a, b\}, \{c, d, e\}\}$.

Théorème 3.

Supposons que les propriétés (Hi) et (Hu) soient vérifiées, alors il existe un sous-ensemble non vide M des variables tel que la partition obtenue par l'agrégation est égale à la borne supérieure de l'ensemble des partitions émises par les variables de M, borne supérieure dans le sens où la classe d'équivalence d'un élément x de donnée par la partition engendrée par l'agrégation est égale à l'intersection des classes d'équivalence de x données par les partitions engendrées par les variables de M.

Preuve :

En effet, prenons $M = \bigcap_{a \in \mathcal{D}_{(1,Ag)}} A \in \mathcal{D}_{(1,Ag)}$. Nous savons que $M \neq \phi$, sans exclure le cas où $M = \mathcal{X}$. Soit $x \in \Omega$. Posons $Cl_i(x)$ la classe d'équivalence de x pour la fonction de classification élémentaire P_i engendrée par le variable X_i. Posons $Cl(x)$ la classe d'équivalence de x pour la classification déterminée par AgP. $Cl_i(x) = \{y \in \Omega / P_i(x, y) = 1\}$ et $Cl(x) = \{y \in \Omega / AgP(x, y) = 1\}$. Soit $y \in \bigcap_{X_i \in M} Cl_i(x)$, cela entraîne que $P_i(x, y) = 1$ pour tout i tel que $X_i \in M$, donc $AgP(x, y) = 1$ et ainsi $\bigcap_{X_i \in M} Cl_i(x) \subset Cl(x)$, pour tout x de Ω.
Inversement, pour x fixé, prenons $y \in Cl(x)$, donc $AgP(x, y) = 1$. Il existe obligatoirement $X_j, X_j \in \mathcal{X}$ tel que $P_j(x, y) = 1$. Pour y fixé, posons $B_{(x,y)} = \{X_i \in \mathcal{X} / P_i(x, y) = 1\}$. Nous obtenons $B(x, y) \in \mathcal{U}_{(1,Ag)} = \mathcal{D}_{(1,Ag)}$ et ainsi $M \subset B_{(x,y)}$. Prenons $z, z \in Cl(x)$ et $z \neq y$, nous obtenons à nouveau $M \subseteq B_{(x,z)}$. Ainsi, pour tout $X_j \in M$, tout $z \in Cl(x)$ est tel que $z \in Cl_j(x)$, ce qui conduit à $Cl(x) \subseteq \bigcap_{X_i \in M} Cl_i(x)$. D'où le résultat.

■

On constate donc que la partition finale obtenue n'est évidemment pas celle obtenue à partir d'une quelconque variable de M. cependant :

- Deux objets classés dans la même classe par le résultat de l'agrégation ont été classés dans la même classe par chacune des variables de M,
- Deux objets classés différemment par une variable quelconque de M sont classés différemment par l'agrégation. M est donc bien un ensemble de variables décisives en 2.

6.5 Hypothèse de finesse

Nous allons, dans ce paragraphe, nous intéresser à la relation de finesse entre partitions pour en examiner les conséquences lorsque la fonction Ag possède certaines propriétés par rapport à cette relation. Par convention, pour désigner qu'une partition $\mathcal{P}$ est plus fine qu'une partition $\mathcal{Q}$, au sens rappelé précédemment, nous écrirons $\mathcal{Q} \preccurlyeq \mathcal{P}$.
Rappelons ci-dessous de manière précise la notion de finesse.

Définition 6.8.

$\mathcal{P}$ et $\mathcal{Q}$ étant deux partitions de Ω. $\mathcal{P}$ est dite plus fine que $\mathcal{Q}$ (on écrit $\mathcal{Q} \preccurlyeq \mathcal{P}$) si et seulement si $\mathcal{P}$ est, au sens large une subdivision de $\mathcal{Q}$, c'est à dire que si A est une classe d'équivalence d'éléments de Ω provenant de la partition $\mathcal{Q}$, A peut s'écrire :

$$A = \bigcup_{x \in A} Cl^p(x)$$

où $Cl^p(x)$ désigne la classe d'équivalence de x pour la partition $\mathcal{P}$.

Il est alors possible d'étendre cette notion de finesse aux familles de fonctions de classification élémentaires.

Définition 6.9.

Soit deux famille de fonctions de classification élémentaires $\{P_i\}$ et $\{Q_i\}$ associées à un même ensemble de variables $\mathcal{X} = \{X_i\}$. $\{P_i\}$ est dite plus fine que $\{Q_i\}$ et on écrit $\{Q_i\} \preccurlyeq \{P_i\}$, si et seulement si pour les partitions associées P_i et Q_i, nous avons $Q_i \preccurlyeq P_i$, pour tout $i, i = 1, \ldots, k$.

Il est alors clair que $P_i(x, y) = 1 \Rightarrow Q_i(x, y) = 1$ et $P_i(x, y) = 1 \Rightarrow Q_i(x, y) = 1$

Définition 6.10.

$\mathcal{P}$ et $\mathcal{Q}$ étant deux partitions de Ω, on appelle borne inférieure de $\mathcal{P}$ et $\mathcal{Q}$, et on note $\mathcal{P} \wedge \mathcal{Q}$, la partition la plus fine des moins fines que $\mathcal{P}$ et $\mathcal{Q}$.
De la même manière, on appelle borne supérieure de $\mathcal{P}$ et $\mathcal{Q}$, et on note $\mathcal{P} \vee \mathcal{Q}$, la partition la moins fine des plus fines que $\mathcal{P}$ et $\mathcal{Q}$

Ainsi, par définition, deux éléments appartenant à la même classe d'équivalence pour $\mathcal{P}$ ou pour $\mathcal{Q}$ appartiennent à la même classe d'équivalence pour $\mathcal{P} \wedge \mathcal{Q}$.
Et toutes les coupures de $\mathcal{P}$ et $\mathcal{Q}$ se retrouvent dans la partition $\mathcal{P} \vee \mathcal{Q}$.
L'ensemble des partitions sur Ω possède un élément le plus grossier , la partition $\mathcal{P}_g = \{\Omega\}$, et un élément le plus fin, la partition $\mathcal{P}_d = \{\{x\}, x \in \Omega\}$.
Pour tout sous-ensemble A de W, nous noterons $\{A, \mathcal{P}_d\}$ la partition telle que le complémentaire de A est partitionné en ses singletons. Pour désigner une partition quelconque de X nous écrirons $\{qcq\}$.

Propriété 6.4. *Propriété (Hf)*
Nous dirons que la fonction Ag vérifie la propriété (Hf) si et seulement si $\{Q_i\} \preccurlyeq \{P_i\}$ entraîne $AgQ \preccurlyeq AgP$.

On peut alors démontrer le premier résultat suivant.

Lemme 6.19.

Si Ag vérifie les propriétés (Hu) et (Hf), nous avons :

$$\bigwedge_{X_i} P_i \preccurlyeq Ag \preccurlyeq \bigvee_{X_i} P_i$$

Preuve :
Posons $P^* = \bigwedge_{X_i} P_i$ et $P^{**} = \bigvee_{X_i} P_i$ et considérons la famille de fonctions de classification élémentaire $\{R_i\}$ telle que $\forall i, P_i = P^*$ et considérons la famille de fonctions de classification élémentaire $\{Q_i\}$ telle que $\forall i, Q_i = P^*$. D'après la propriété (Hu), nous avons $AgP = P^*$ et $AgQ = P^{**}$. Par ailleurs, il est immédiat que $\forall i, P^* \preccurlyeq P_i \preccurlyeq P^{**}$. Avec la propriété (Hf), on obtient le résultat.
■

Soit Ag une fonction agrégation, on considère les deux familles suivantes :
$\mathcal{U}_{Ag} = \{\{X_i\} \subset \mathcal{X} / P_1(\{X_i\})\}$ où $P_1(\{X_i\})$ désigne la propriété ci-dessous :

$\forall A, A \subset \Omega, \forall i, P_i = (A/A^c)$ pour i tel que $\{X_i\} \in \mathcal{U}_{Ag}$ et $P_i = (qcq)$ pour i tel que $\{X_i\} \notin \mathcal{U}_{Ag} \Rightarrow AgP = (A/A^c), \forall A, A \subset \Omega$
et
$\mathcal{V}_{Ag} = \{\{Xi\} \subset \mathcal{X}/P_2(\{X_i\})\}$ où $P_2(\{X_i\})$ désigne la propriété ci-dessous :
$\forall A, A \subset \Omega, \forall i, P_i = \{A, \mathcal{P}_d\}$ pour i tel que $\{X_i\} \in \mathcal{U}_{Ag}$ et $P_i = (qcq)$ pour i tel que $\{X_i\} \notin \mathcal{U}_{Ag} \Rightarrow AgP = \{A, \mathcal{P}_d\}, \forall A, A \subset \Omega$.
Il est immédiat que si Ag vérifie la propriété (Hu), ni $\mathcal{U}_{Ag}$, ni $\mathcal{V}_{Ag}$ ne sont vides car $\Omega \in \mathcal{U}_{Ag} \cap \mathcal{V}_{Ag}$

Lemme 6.20.

Si Ag vérifie les propriétés d'unanimité et de finesse, on a :
Soit $U \in \mathcal{U}_{Ag}$ et $\{R_i\}$ une famille de fonctions de classification élémentaires telle que :
$\mathcal{P}_i = \{A, B, \ldots, H\}$ pour i tel que $X_i \in U$ et $\mathcal{P}_i = \{\Omega\}$ sinon, alors $AgR = \{A, B, \ldots, H\}$
De même :
Soit $V \in \mathcal{V}_{Ag}$ et $\{Q_i\}$ une famille de fonctions de classification élémentaires telle que :
$\mathcal{P}_i = \{A, B, \ldots, H\}$ pour i tel que $X_i \in V$ et $\mathcal{P}_i = (\mathcal{P}_d)$ sinon, alors $AgQ = \{A, B, \ldots, H\}$

Preuve :

Pour R :
Prenons Q_Y telle que $q_{Y_i} = \{Y, Y^c\}$ pour $Y_i \in U$ et $q_{Y_i} = \{\Omega\}$ sinon, avec Y valant successivement $A, B, \ldots$, puis H.
Il est immédiat que $Q_Y \preccurlyeq R$, donc $\{Y, Y^c\} \preccurlyeq AgR$ avec $Y = A, Y = B, \ldots$, jusqu'à $Y = H$. Cela donne $\bigvee_{Y \in \{A,B,\ldots,H\}} \{Y, Y^c\} = \{A, B, \ldots, H\} \preccurlyeq AgR$.
Prenons ensuite Q' telle que $q'_{Y_i} = \{A, B, \ldots, H\}$ pour tout $Y_i \in \mathcal{X}$. Donc $AgQ' = \{A, B, \ldots, H\}$ et comme $R \preccurlyeq Q'$, on obtient $\{A, B, \ldots, H\} \preccurlyeq AgR \preccurlyeq \{A, B, \ldots, H\}$. D'où l'égalité annoncée.
Pour Q :
Prenons Q^*_Y telle que $q^*_{Y_i} = \{Y, \mathcal{P}_d\}$ pour $Y_i \in V$ et $q_{Y_i} = \mathcal{P}_d$ sinon., avec Y valant successivement $A, B, \ldots$, puis H.
Nous obtenons $R^* \preccurlyeq Q^*_Y$ et $AgQ \preccurlyeq \{Y, \mathcal{P}_d\}$, d'où
$AgQ \preccurlyeq \bigwedge_{Y \in \{A,B,\ldots,H\}} \{Y, \mathcal{P}_d\} = \{A, B, \ldots, H\}$ et l'égalité en considérant Q' comme ci-dessus.
■

Définition 6.11.

Nous appelons ensemble de variables décisives essentielles tout élément, sous-ensemble de $\mathcal{X}$, appartenant à la famille :

$\mathcal{W}_{Ag} = \{W \subset \mathcal{X}/[\mathcal{P}_i = \{A, B, \ldots, H\}$ *sur* W *et* $\mathcal{P}_i = (qcq)$ *ailleurs* $\Rightarrow [AgP = \{A, B, \ldots, H\}$ *pour toute partition* $\{A, B, \ldots, H\}$ *de* $X]\}$

On obtient alors le résultat suivant.

Théorème 4.

Si Ag vérifie les hypothèses (Hu) (unanimité) et (Hf) (finesse), nous avons :

$$\mathcal{W}_{Ag} = \mathcal{U}_{Ag} \cap \mathcal{V}_{Ag}$$

Preuve :

Si $W \in \mathcal{U}_{Ag} \cap \mathcal{V}_{Ag}$ il est immédiat que $R \preccurlyeq P \preccurlyeq Q$ (R et Q définis telles que dans le résultat précédent) et nous avons démontré que $AgR = AgQ = \{A, B, \ldots, H\}$, donc $AgP = \{A, B, \ldots, H\}$.

Cela donne $\mathcal{U}_{Ag} \cap \mathcal{V}_{Ag} \subset \mathcal{W}_{Ag}$.

Si $W \in \mathcal{W}_{Ag}$, il est immédiat que $W \in \mathcal{U}_{Ag} \cap \mathcal{V}_{Ag}$ car la définition de $\mathcal{W}_{Ag}$ est valable quelle que soit la partition de X, donc pour $\{A, A^c\}$ et $\{A, \mathcal{P}_d\}$.

■

Les résultats suivants déterminent la structure des ensembles $\mathcal{U}_{Ag}, \mathcal{V}_{Ag}$ et $\mathcal{W}_{Ag}$

Lemme 6.21.

Sous les propriétés (Hu) et (Hf), $\mathcal{U}_{Ag}$ *et* $\mathcal{V}_{Ag}$ *sont des préfiltres.*

Preuve :

Soit $U \in \mathcal{U}_{Ag}$ et $U' \supset U$. Considérons la fonction élémentaire de classification Q définie par $q_{Y_i} = \{A, A^c\}$ sur U' et $q_{Y_i} = (qcq)$ sinon. Il est immédiat qu'elle fait partie de la famille des fonctions élémentaires de classification P telles que $P_{Y_i} = \{A, A^c\}$ sur U et (qcq) ailleurs, donc $AgQ = \{A, A^c\}$.

Soit $V \in \mathcal{V}_{Ag}$ et $V' \supset V$. Considérons la fonction élémentaire de classification Q^* définie par $q^*_{Y_i} = \{A, \mathcal{P}_d\}$ sur U' et $q_{Y_i} = (qcq)$ sinon. Il est immédiat qu'elle fait partie de la famille des fonctions élémentaires de classification P telles que $P_{Y_i} = \{A, \mathcal{P}_d\}$ sur V et (qcq) ailleurs, donc $AgQ^* = \{A, \mathcal{P}_d\}$.

■

Lemme 6.22.

$\mathcal{U}_{Ag}$ *et* $\mathcal{V}_{Ag}$ *sont des préfiltres propres.*

Preuve :

Soient U et U' appartenant à $\mathcal{U}_{Ag}$ avec $U \cap U' = \phi$.

On en déduit que $U^c \in \mathcal{U}_{Ag}$ car $\mathcal{U}_{Ag}$ est un préfiltre. Soit $\{A, A^c\} \neq \{B, B^c\}$. Considérons

	U	U^c	Ag
P	$\{A, A^c\}$	(qcq)	$\{A, A^c\}$
Donc P'	$\{A, A^c\}$	$\{B, B^c\}$	$\{A, A^c\}$
Q	(qcq)	$\{B, B^c\}$	$\{B, B^c\}$
Donc Q'	$\{A, A^c\}$	$\{B, B^c\}$	$\{B, B^c\}$

D'où une contradiction puisque $P' = Q'$.
Soient V et V' appartenant à $\mathcal{V}_{Ag}$ avec $V \cap V' = \phi$.
On en déduit que $V^c \in \mathcal{V}_{Ag}$ car $\mathcal{V}_{Ag}$ est un préfiltre. Soit $\{A, \mathcal{P}_d\} \neq \{B, \mathcal{P}_d\}$. Considérons

	U	U^c	Ag
P	$\{A, \mathcal{P}_d\}$	(qcq)	$\{A, \mathcal{P}_d\}$
Donc P'	$\{A, \mathcal{P}_d\}$	$\{B, \mathcal{P}_d\}$	$\{A, \mathcal{P}_d\}$
Q	(qcq)	$\{B, \mathcal{P}_d\}$	$\{B, \mathcal{P}_d\}$
Donc Q'	$\{A, \mathcal{P}_d\}$	$\{B, \mathcal{P}_d\}$	$\{B, \mathcal{P}_d\}$

D'où une contradiction puisque $P' = Q'$.

■

On en déduit alors le théorème suivant.

Théorème 5.

Si la fonction agrégation Ag vérifie les propriétés (Hu) et (Hf), il est immédiat que

$$\mathcal{W}_{Ag} = \mathcal{U}_{Ag} \cap \mathcal{V}_{Ag}$$

est aussi un préfiltre propre et l'on a :

$$\mathcal{W}_{Ag} = \{U \cup V\}$$

avec $U \in \mathcal{U}_{Ag}$ *et* $V \in \mathcal{V}_{Ag}$.

Voyons quelles sont les conséquences de ce résultat. Ayant supposé dès le début de ce chapitre que l'ensemble $\mathcal{X}$ des variables et de cardinal fini, il est clair que l'on peut préciser la base de tout préfiltre propre. Sans oublier que tout filtre, tout ultrapréfiltre,

tout ultrafiltre est un préfiltre, dont les éléments de base peuvent se décrire facilement. Soient $\mathcal{V}$ et $\mathcal{V}'$ deux sous ensembles de $\mathcal{X}$ formant la base d'un préfiltre propre, $\mathcal{V} \cap \mathcal{V}' = \phi$. Les variables placées dans cette intersection jouent un rôle clé. Si elles produisent la même classification soit avec les variables de $\mathcal{V}$, soit avec les variables de $\mathcal{V}'$, elles forment un ensemble de variables décisives.

6.6 Conclusion

Dans ce chapitre, nous avons essayé d'aborder les défintions classiques d'une procédure d'agrégation des préférences en termes de la classification. Ensuite, nous avons étudié quelques lemmes et propriétés.

L'agrégation des préférence a été rendu célèbre par le très fameux Théorème d'impossibilité de ***ARROW*** dont la première version date de 1951.
"...Il n'existe aucune procédure d'agrégation des préférences individuelles qui soit à la fois rationnelle, efficiente et démocratique".

Grâce à ce théorème, certains chercheurs ont mis l'accent sur l'aptitude à sélectionner un ensemble non vide de résultat parmi l'ensemble de tous les résultats possibles.

Nous avons montré dans ce chapitre que le problème de la classification revient à définir un type de pouvoir détenu par une coalition "décisive". Il faut ensuite déterminer l'ensemble des telles coalitions décisives.

Nous avons donné quelques propriétés d'un famille de coalition décisive. Ensuite nous avons étudié l'hypothèse de finesse entre les partitions.

Finalement, nous pouvons dire que nous avons essayé d'ouvrir la porte vers un autre approche (l'agrégation des préférences) pour introduire une théorie mathématique assez riche et peu utilisé dans la classification des données.

Chapitre 7

Application

Nous reprenons ici le même exemple utilisé dans la méthode d'agrégation des opinions 4.4.1 pour qu'on puisse comparer le résultat avec celle de *Grötschel* [GW89]. Le tableau des données est constitué de 36 cétacés (Tab. 7.1) et l'ensemble des variables est constitué de 15 variables nominales pouvant prendre de 2 à 5 valeurs. (Les données manquantes sont représentées par "*").

Les 15 ensembles de modalités $E_1, E_2, \ldots, E_3$ sont :
$E_1 = \{0, 1\}$
$E_2 = \{0, 1, 2, 3, 4, 5\}$
$E_3 = \{0, 1\}$
$E_4 = \{0, 1, , 2, 3\}$
$E_5 = \{0, 1, 2, 3\}$
$E_6 = \{0, 1, 2, 3\}$
$E_7 = \{0, 1, 2, 3, 4\}$
$E_8 = \{0, 1, 2, 3\}$
$E_9 = \{0, 1, 2, 3\}$
$E_{10} = \{0, 1, 2, 3\}$
$E_{11} = \{0, 1\}$
$E_{12} = \{0, 1\}$
$E_{13} = \{0, 1, 2, 3, 4\}$
$E_{14} = \{0, 1, 2\}$
$E_{15} = \{0, 1, , 2, 3\}$

Individu	Variables														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Balaena	0	5	0	0	0	1	2	3	3	4	1	*	2	0	0
Balaenoptea	0	4	0	0	3	0	4	3	3	0	0	*	4	2	0
Balaenoptea Mus	0	4	0	0	3	3	4	3	3	3	0	*	4	2	0
Berardius	1	2	1	2	2	0	0	0	2	0	1	1	2	1	2
Cephalorhynehus	0	1	1	1	2	2	1	0	1	4	1	1	2	0	1
Delphinapterus	1	3	1	0	0	1	1	1	1	2	0	1	2	0	2
Delphinus	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1
Eschrichtius	0	1	0	0	0	3	3	3	3	3	0	*	4	1	0
Eubalaena	0	5	0	0	0	1	2	3	3	1	1	*	2	0	0
Globicephala	0	3	1	0	2	3	1	0	1	3	1	1	1	0	1
Grampus	0	3	1	0	2	3	0	0	1	0	1	1	4	0	1
Hyperoodon	1	2	1	2	2	0	0	0	2	0	1	1	4	1	2
Inia	1	2	1	3	1	1	1	1	2	0	0	0	0	0	*
Kogia	0	0	0	0	1	0	0	0	0	0	1	*	4	0	3
lagenorhynchus	0	2	1	2	2	2	1	1	1	0	1	1	2	0	1
Lipotes	1	2	1	3	1	1	1	1	0	0	0	0	0	0	*
Lissodelphis	0	1	1	1	0	2	1	1	1	0	1	1	1	0	1
Megaptera	0	4	0	0	3	3	4	3	3	3	0	*	4	2	0
Mesoplodon	1	1	1	2	2	0	0	0	2	0	1	1	4	1	2
Monodon	1	3	1	0	0	1	0	0	1	3	0	1	2	0	2
Neophalaena	0	1	0	0	3	0	2	3	3	0	1	*	2	0	0
Neophocaena	0	1	1	0	0	0	1	1	1	3	1	1	4	0	1
Orcaella	1	3	1	0	2	0	1	1	1	1	1	1	3	0	1
Orcinus	0	3	1	0	2	1	1	2	1	3	1	1	2	0	1
Phocaena	0	1	1	0	1	0	1	1	1	0	1	1	4	0	1
Physeter	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3
Platanista	1	2	1	3	1	1	1	1	2	1	0	0	0	0	1
Pseudorea	0	3	1	0	2	3	1	0	1	1	1	1	4	0	1
Sotalia	0	2	1	2	2	2	1	1	1	2	1	1	3	0	1
Sousa	0	2	1	2	2	2	1	1	1	0	1	1	3	0	1
Stenella	0	2	1	2	2	2	1	1	1	3	1	1	4	0	1
Steno	0	1	1	2	2	2	1	1	1	0	1	1	1	0	1
Stenodelphis	1	2	1	2	1	1	1	1	2	1	0	0	0	0	*
Tasmacetus	0	3	1	2	2	0	0	0	2	0	1	1	2	1	2
Tussio	0	2	1	2	2	2	1	1	1	0	1	1	4	0	1
Ziphus	0	1	1	2	2	0	0	0	2	3	1	1	4	1	2

Tab. 7.1 – Données cétacés

Nous en déduisons :

$X_1^{-1}(0) = \{\omega_1, \omega_2, \omega_3, \omega_5, \omega_7, \omega_8, \omega_9, \omega_{10}, \omega_{11}, \omega_{14}, \omega_{15}, \omega_{17}, \omega_{18}, \omega_{21}, \omega_{22}, \omega_{24}, \omega_{25}, \omega_{26}, \omega_{28},$
$\omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{34}, \omega_{35}, \omega_{36}\}$
$X_1^{-1}(1) = \{\omega_4, \omega_6, \omega_{12}, \omega_{13}, \omega_{16}, \omega_{19}, \omega_{20}, \omega_{23}, \omega_{27}, \omega_{33}\}$

$X_2^{-1}(0) = \{\omega_{14}, \omega_{26}\}$
$X_2^{-1}(1) = \{\omega_5, \omega_8, \omega_{17}, \omega_{19}, \omega_{20}, \omega_{25}, \omega_{32}, \omega_{36}\}$
$X_2^{-1}(2) = \{\omega_4, \omega_7, \omega_{12}, \omega_{13}, \omega_{15}, \omega_{16}, \omega_{27}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{33}, \omega_{35}\}$
$X_2^{-1}(3) = \{\omega_6, \omega_{10}, \omega_{11}, \omega_{20}, \omega_{23}, \omega_{24}, \omega_{28}, \omega_{34}\}$
$X_2^{-1}(4) = \{\omega_2, \omega_3, \omega_{18}\}$
$X_2^{-1}(5) = \{\omega_1, \omega_9\}$

$X_3^{-1}(0) = \{\omega_1, \omega_2, \omega_3, \omega_8, \omega_9, \omega_{14}, \omega_{18}, \omega_{21}, \omega_{26}\}$
$X_3^{-1}(1) = \{\omega_4, \omega_5, \omega_6, \omega_7, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{13}, \omega_{15}, \omega_{16}, \omega_{17}, \omega_{19}, \omega_{20}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{27}\omega_{28}, \omega_{29},$
$\omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}\omega_{34}, \omega_{35}, \omega_{36}\}$

$X_4^{-1}(0) = \{\omega_1, \omega_2, \omega_3, \omega_6, \omega_8, \omega_9, \omega_{10}, \omega_{11}, \omega_{14}, \omega_{18}, \omega_{20}, \omega_{21}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{26}, \omega_{28}\}$
$X_4^{-1}(1) = \{\omega_5, \omega_{17}\}$
$X_4^{-1}(2) = \{\omega_4, \omega_7, \omega_{12}, \omega_{15}, \omega_{19}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}\omega_{34}, \omega_{35}, \omega_{36}\}$
$X_4^{-1}(3) = \{\omega_{13}, \omega_{27}\}$

$X_5^{-1}(0) = \{\omega_1, \omega_6, \omega_8, \omega_9, \omega_{17}, \omega_{20}, , \omega_{22}, \omega_{26}\}$
$X_5^{-1}(1) = \{\omega_{13}, \omega_{14}, \omega_{16}, \omega_{25}, \omega_{27}\}$
$X_5^{-1}(2) = \{\omega_4, \omega_5, \omega_7, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{15}, \omega_{19}, \omega_{23}, \omega_{24}, \omega_{28}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}, \omega_{35}, \omega_{36}\}$
$X_5^{-1}(3) = \{\omega_2, \omega_3, \omega_{18}, \omega_{21}\}$

$X_6^{-1}(0) = \{\omega_2, \omega_{12}, \omega_{14}, \omega_{19}, \omega_{21}, \omega_{22}, \omega_{23}, \omega_{25}, \omega_{26}, \omega_{34}, \omega_{36}\}$
$X_6^{-1}(1) = \{\omega_1, \omega_6, \omega_9, \omega_{13}, \omega_{16}, \omega_{20}, \omega_{24}, \omega_{27}, \omega_{33}\}$
$X_6^{-1}(2) = \{\omega_5, \omega_7, \omega_{15}, \omega_{17}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}$
$X_6^{-1}(3) = \{\omega_3, \omega_8, \omega_{10}, \omega_{11}, \omega_{18}, \omega_{28}\}$

$X_7^{-1}(0) = \{\omega_4, \omega_{11}, \omega_{12}, \omega_{14}, \omega_{19}, \omega_{20}, \omega_{34}, \omega_{36}\}$
$X_7^{-1}(1) = \{\omega_5, \omega_6, \omega_7, \omega_{10}, \omega_{13}, \omega_{15}, \omega_{16}, \omega_{17}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{27}, \omega_{28}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}, \omega_{35}\}$
$X_7^{-1}(2) = \{\omega_1, \omega_9, \omega_{21}\}$
$X_7^{-1}(3) = \{\omega_8\}$
$X_7^{-1}(4) = \{\omega_2, \omega_3, \omega_{18}\}$

$X_8^{-1}(0) = \{\omega_4, \omega_5, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{14}, \omega_{19}, \omega_{20}, \omega_{26}, \omega_{28}, \omega_{34}, \omega_{36}\}$
$X_8^{-1}(1) = \{\omega_6, \omega_7, \omega_{13}, \omega_{15}, \omega_{16}, \omega_{17}, \omega_{22}, \omega_{23}, \omega_{25}, \omega_{27}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}, \omega_{35}, \}$
$X_8^{-1}(2) = \{\omega_{24}\}$
$X_8^{-1}(3) = \{\omega_1, \omega_2, \omega_3, \omega_8, \omega_9, \omega_{18}, \omega_{21}\}$

$X_9^{-1}(0) = \{\omega_{14}, \omega_{16}, \omega_{26}\}$
$X_9^{-1}(1) = \{\omega_5, \omega_6, \omega_7, \omega_{10}, \omega_{11}, \omega_{15}, \omega_{17}, \omega_{20}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{27}, \omega_{28}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}$
$X_9^{-1}(2) = \{\omega_4, \omega_{12}, \omega_{13}, \omega_{19}, \omega_{27}, \omega_{33}, \omega_{34}, \omega_{36}\}$
$X_9^{-1}(3) = \{\omega_1, \omega_2, \omega_3, \omega_8, \omega_9, \omega_{18}, \omega_{21}\}$

$X_{10}^{-1}(0) = \{\omega_2, \omega_4, \omega_7, \omega_{11}, \omega_{12}, \omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}, \omega_{17}, \omega_{19}, \omega_{21}, \omega_{25}, \omega_{26}, \omega_{30}, \omega_{32}, \omega_{34}, \omega_{35}\}$
$X_{10}^{-1}(1) = \{\omega_9, \omega_{23}, \omega_{27}, \omega_{28}, \omega_{33}\}$
$X_{10}^{-1}(2) = \{\omega_6, \omega_{29}\}$
$X_{10}^{-1}(3) = \{\omega_3, \omega_8, \omega_{10}, \omega_{18}, \omega_{20}, \omega_{22}, \omega_{24}, \omega_{31}, \omega_{36}\}$
$X_{10}^{-1}(4) = \{\omega_1, \omega_5\}$

$X_{11}^{-1}(0) = \{\omega_2, \omega_3, \omega_6, \omega_{13}, \omega_{16}, \omega_{18}, \omega_{20}, \omega_{27}, \omega_{33}\}$
$X_{11}^{-1}(1) = \{\omega_1, \omega_4, \omega_5, \omega_7, \omega_8, \omega_9, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{14}, \omega_{15}, \omega_{17}, \omega_{19}, \omega_{21}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{26}, \omega_{28},$
$\omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{34}, \omega_{35}, \omega_{36}\}$

$X_{12}^{-1}(0) = \{\omega_{13}, \omega_{16}, \omega_{26}, \omega_{27}, \omega_{33}\}$
$X_{12}^{-1}(1) = \{\omega_4, \omega_5, \omega_6, \omega_7, \omega_{10}, \omega_{11}, \omega_{12}, \omega_{15}, \omega_{17}, \omega_{19}, \omega_{20}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{28}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32},$
$\omega_{34}, \omega_{35}, \omega_{36}\}$
$X_{12}^{-1}(*) = \{\omega_1, \omega_2, \omega_3, \omega_8, \omega_9, \omega_{14}, \omega_{18}, \omega_{21}\}$

$X_{13}^{-1}(0) = \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\}$
$X_{13}^{-1}(1) = \{\omega_7, \omega_{10}, \omega_{17}, \omega_{26}, \omega_{32}\}$
$X_{13}^{-1}(2) = \{\omega_1, \omega_4, \omega_5, \omega_6, \omega_9, \omega_{15}, \omega_{20}, \omega_{21}, \omega_{24}, \omega_{34}\}$
$X_{13}^{-1}(3) = \{\omega_{23}, \omega_{29}, \omega_{30}\}$
$X_{13}^{-1}(4) = \{\omega_2, \omega_3, \omega_8, \omega_{11}, \omega_{12}, \omega_{14}, \omega_{18}, \omega_{19}, \omega_{22}, \omega_{25}, \omega_{28}, \omega_{31}, \omega_{35}, \omega_{36}\}$

$X_{14}^{-1}(0) = \{\omega_1, \omega_5, \omega_6, \omega_7, \omega_9, \omega_{10}, \omega_{11}, \omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}, \omega_{17}, \omega_{20}, \omega_{21}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{26}, \omega_{27}, \omega_{28},$
$\omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{33}, \omega_{35}\}$
$X_{14}^{-1}(1) = \{\omega_1, \omega_8, \omega_{12}, \omega_{19}, \omega_{34}, \omega_{36}\}$
$X_{14}^{-1}(2) = \{\omega_2, \omega_3, \omega_{18}\}$

$X_{15}^{-1}(0) = \{\omega_1, \omega_2, \omega_3, \omega_8, \omega_9, \omega_{18}, \omega_{21}\}$
$X_{15}^{-1}(1) = \{\omega_5, \omega_7, \omega_{10}, \omega_{11}, \omega_{15}, \omega_{17}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{27}, \omega_{28}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}$
$X_{15}^{-1}(2) = \{\omega_4, \omega_6, \omega_{12}, \omega_{19}, \omega_{20}, \omega_{34}, \omega_{36}\}$
$X_{15}^{-1}(3) = \{\omega_{14}, \omega_{26}\}$
$X_{15}^{-1}(*) = \{\omega_{13}, \omega_{16}, \omega_{33}\}$

Pour chaque $\omega_i \in \Omega$ on se propose de construire une base de voisinage ; $B(\omega_i, q)$ en choisissant dans un premier temps $q = 15$.

Première étape :
Nous trouvons :

$\omega_1 = \{0, 5, 0, 0, 0, 1, 2, 3, 3, 4, 1, *, 2, 0, 0\} \Rightarrow$
$B(\omega_1, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(5) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(1) \cap X_7^{-1}(2) \cap X_8^{-1}(3) \cap X_9^{-1}(3) \cap X_{10}^{-1}(4) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(0)\} = \{\omega_1\}$

$\omega_2 = \{0, 4, 0, 0, 3, 0, 4, 3, 3, 0, 0, *, 4, 2, 0\} \Rightarrow$
$B(\omega_2, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(4) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(3) \cap X_6^{-1}(0) \cap X_7^{-1}(4) \cap X_8^{-1}(3) \cap X_9^{-1}(3) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(2) \cap X_{15}^{-1}(0)\} = \{\omega_2\}$

$\omega_3 = \{0, 4, 0, 0, 3, 3, 4, 3, 3, 3, 0, *, 4, 2, 0\} \Rightarrow$
$B(\omega_3, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(4) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(3) \cap X_6^{-1}(3) \cap X_7^{-1}(4) \cap X_8^{-1}(3) \cap X_9^{-1}(3) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(2) \cap X_{15}^{-1}(0)\} = \{\omega_3, \omega_{18}\}$

$\omega_4 = \{1, 2, 1, 2, 2, 0, 0, 0, 2, 0, 1, 1, 2, 1, 2\} \Rightarrow$
$B(\omega_4, 15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap X_9^{-1}(2) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(2)\} = \{\omega_4\}$

$\omega_5 = \{0, 1, 1, 1, 2, 2, 1, 0, 1, 4, 1, 1, 2, 0, 1\} \Rightarrow$
$B(\omega_5, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(1) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(0) \cap X_9^{-1}(1) \cap X_{10}^{-1}(4) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_5\}$

$\omega_6 = \{1, 3, 1, 0, 0, 1, 1, 1, 1, 2, 0, 1, 2, 0, 2\} \Rightarrow$
$B(\omega_6, 15) = \{X_1^{-1}(1) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap X_9^{-1}(1) \cap X_{10}^{-1}(2) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(2)\} = \{\omega_6\}$

$\omega_7 = \{0,2,1,2,2,2,1,1,1,0,1,1,1,0,1\} \Rightarrow$
$B(\omega_7,15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(1) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_7\}$

$\omega_8 = \{0,1,0,0,0,3,3,3,3,3,0,*,4,1,0\} \Rightarrow$
$B(\omega_8,15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(3) \cap X_7^{-1}(3) \cap X_8^{-1}(3) \cap$
$X_9^{-1}(3) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(0)\} = \{\omega_8\}$

$\omega_9 = \{0,5,0,0,0,1,2,3,3,1,1,*,2,0,0\} \Rightarrow$
$B(\omega_9,15) = \{X_1^{-1}(0) \cap X_2^{-1}(5) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(1) \cap X_7^{-1}(2) \cap X_8^{-1}(3) \cap$
$X_9^{-1}(3) \cap X_{10}^{-1}(1) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(0)\} = \{\omega_9\}$

$\omega_{10} = \{0,3,1,0,2,3,1,0,1,3,1,1,1,0,1\} \Rightarrow$
$B(\omega_{10},15) = \{X_1^{-1}(0) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(2) \cap X_6^{-1}(3) \cap X_7^{-1}(1) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(1) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{10}\}$

$\omega_{11} = \{0,3,1,0,2,3,0,0,1,0,1,1,4,0,1\} \Rightarrow$
$B(\omega_{11},15) = \{X_1^{-1}(0) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(2) \cap X_6^{-1}(3) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{11}\}$

$\omega_{12} = \{1,2,1,2,2,0,0,0,2,0,1,1,4,1,2\} \Rightarrow$
$B(\omega_{12},15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(2)\} = \{\omega_{12}\}$

$\omega_{13} = \{1,2,1,3,1,1,1,1,2,0,0,0,0,0,*\} \Rightarrow$
$B(\omega_{13},15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(3) \cap X_5^{-1}(1) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(0) \cap X_{13}^{-1}(0) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(*)\} = \{\omega_{13}\}$

$\omega_{14} = \{0,0,0,0,1,0,0,0,0,0,1,*,4,0,3\} \Rightarrow$
$B(\omega_{14},15) = \{X_1^{-1}(0) \cap X_2^{-1}(0) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(1) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(0) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(3)\} = \{\omega_{14}\}$

$\omega_{15} = \{0,2,1,2,2,2,1,1,1,0,1,1,2,0,1\} \Rightarrow$
$B(\omega_{15},15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{15}\}$

$\omega_{16} = \{1,2,1,3,1,1,1,1,0,0,0,0,0,0,*\} \Rightarrow$
$B(\omega_{16},15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_1^{-1}(1) \cap X_4^{-1}(3) \cap X_5^{-1}(1) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(0) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(0) \cap X_{13}^{-1}(0) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(*)\} = \{\omega_{16}\}$

$\omega_{17} = \{0,1,1,1,0,2,1,1,1,0,1,1,1,0,1\} \Rightarrow$
$B(\omega_{17},15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(1) \cap X_5^{-1}(0) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(1) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{17}\}$

$\omega_{18} = \{0,4,0,0,3,3,4,3,3,3,0,*,4,2,0\} \Rightarrow$
$B(\omega_{18},15) = \{X_1^{-1}(0) \cap X_2^{-1}(4) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(3) \cap X_6^{-1}(3) \cap X_7^{-1}(4) \cap X_8^{-1}(3) \cap$
$X_9^{-1}(3) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(2) \cap X_{15}^{-1}(0)\} = \{\omega_3, \omega_{18}\}$

$\omega_{19} = \{1,1,1,2,2,0,0,0,2,0,1,1,4,1,2\} \Rightarrow$
$B(\omega_{19},15) = \{X_1^{-1}(1) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(2)\} = \{\omega_{19}\}$

$\omega_{20} = \{1,3,1,0,0,1,0,0,1,3,0,1,2,0,2\} \Rightarrow$
$B(\omega_{20},15) = \{X_1^{-1}(1) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(1) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(2)\} = \{\omega_{20}\}$

$\omega_{21} = \{0,1,0,0,3,0,2,3,3,0,1,*,2,0,0\} \Rightarrow$
$B(\omega_{21},15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(3) \cap X_6^{-1}(0) \cap X_7^{-1}(2) \cap X_8^{-1}(3) \cap$
$X_9^{-1}(3) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(*) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(0)\} = \{\omega_{21}\}$

$\omega_{22} = \{0,1,1,0,0,0,1,1,1,3,1,1,4,0,1\} \Rightarrow$
$B(\omega_{22},15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(0) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{22}\}$

$\omega_{23} = \{1,3,1,0,2,0,1,1,1,1,1,1,3,0,1\} \Rightarrow$
$B(\omega_{23},15) = \{X_1^{-1}(1) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(1) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(3) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{23}\}$

$\omega_{24} = \{0,3,1,0,2,1,1,2,1,3,1,1,2,0,1\} \Rightarrow$
$B(\omega_{24},15) = \{X_1^{-1}(0) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(2) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(2) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{24}\}$

$\omega_{25} = \{0,1,1,0,1,0,1,1,1,0,1,1,4,0,1\} \Rightarrow$
$B(\omega_{25},15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(1) \cap X_6^{-1}(0) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{25}\}$

$\omega_{26} = \{0,0,0,0,0,0,0,0,0,0,1,0,1,0,3\} \Rightarrow$
$B(\omega_{26},15) = \{X_1^{-1}(0) \cap X_2^{-1}(0) \cap X_3^{-1}(0) \cap X_4^{-1}(0) \cap X_5^{-1}(0) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(0) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(0) \cap X_{13}^{-1}(1) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(3)\} = \{\omega_{26}\}$

$\omega_{27} = \{1,2,1,3,1,1,1,1,2,1,0,0,0,0,1\} \Rightarrow$
$B(\omega_{27},15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(3) \cap X_5^{-1}(1) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(1) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(0) \cap X_{13}^{-1}(0) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{27}\}$

$\omega_{28} = \{0,3,1,0,2,3,1,0,1,1,1,1,4,0,1\} \Rightarrow$
$B(\omega_{28},15) = \{X_1^{-1}(0) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(0) \cap X_5^{-1}(2) \cap X_6^{-1}(3) \cap X_7^{-1}(1) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(1) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{28}\}$

$\omega_{29} = \{0,2,1,2,2,2,1,1,1,2,1,1,3,0,1\} \Rightarrow$
$B(\omega_{29},15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(2) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(3) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{29}\}$

$\omega_{30} = \{0,2,1,2,2,2,1,1,1,0,1,1,3,0,1\} \Rightarrow$
$B(\omega_{30},15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(3) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{30}\}$

$\omega_{31} = \{0,2,1,2,2,2,1,1,1,3,1,1,4,0,1\} \Rightarrow$
$B(\omega_{31},15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{31}\}$

$\omega_{32} = \{0,1,1,2,2,2,1,1,1,0,1,1,1,0,1\} \Rightarrow$
$B(\omega_{32},15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(1) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{32}\}$

$\omega_{33} = \{1,2,1,2,1,1,1,1,2,1,0,0,0,0,*\} \Rightarrow$
$B(\omega_{33},15) = \{X_1^{-1}(1) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(1) \cap X_6^{-1}(1) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(1) \cap X_{11}^{-1}(0) \cap X_{12}^{-1}(0) \cap X_{13}^{-1}(0) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(*)\} = \{\omega_{33}\}$

$\omega_{34} = \{0, 3, 1, 2, 2, 0, 0, 0, 2, 0, 1, 1, 2, 1, 2\} \Rightarrow$
$B(\omega_{34}, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(3) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(2) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(2)\} = \{\omega_{34}\}$

$\omega_{35} = \{0, 2, 1, 2, 2, 2, 1, 1, 1, 0, 1, 1, 4, 0, 1\} \Rightarrow$
$B(\omega_{35}, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(2) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(2) \cap X_7^{-1}(1) \cap X_8^{-1}(1) \cap$
$X_9^{-1}(1) \cap X_{10}^{-1}(0) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(0) \cap X_{15}^{-1}(1)\} = \{\omega_{35}\}$

$\omega_{36} = \{0, 1, 1, 2, 2, 0, 0, 0, 2, 3, 1, 1, 4, 1, 2\} \Rightarrow$
$B(\omega_{36}, 15) = \{X_1^{-1}(0) \cap X_2^{-1}(1) \cap X_3^{-1}(1) \cap X_4^{-1}(2) \cap X_5^{-1}(2) \cap X_6^{-1}(0) \cap X_7^{-1}(0) \cap X_8^{-1}(0) \cap$
$X_9^{-1}(2) \cap X_{10}^{-1}(3) \cap X_{11}^{-1}(1) \cap X_{12}^{-1}(1) \cap X_{13}^{-1}(4) \cap X_{14}^{-1}(1) \cap X_{15}^{-1}(2)\} = \{\omega_{36}\}$

Nous proposons alors de construire sur Ω la relation R définie par :

$$\forall \omega_i \in \Omega\,; \omega_i R \omega_j \Leftrightarrow \omega_j \in B(\omega_i, q)$$

Il claire que cette relation est une relation d'équivalence qui nous donne une partition des l'ensemble Ω. Cette partition est très proche d'une partition discrete où dans chaque classe nous avons un seul individu sauf une classe où nous avons deux individus, à savoir $\{\omega_3, \omega_{18}\}$, ce qui est tout à fait normal car les deux individus sont les mêmes valeurs. Ce qui peut être intéresent si nous voulons trouver dans une base des données les doublons :

$\mathcal{P}_{15} = \{\{\omega_1\}, \{\omega_2\}, \{\omega_3, \omega_{18}\}, \{\omega_4\}, \{\omega_5\}, \{\omega_6\}, \{\omega_7\}, \{\omega_8\}, \{\omega_9\}, \{\omega_{10}\}, \{\omega_{11}\}, \{\omega_{12}\}, \{\omega_{13}\}, \{\omega_{14}\},$
$\{\omega_{15}\}, \{\omega_{16}\}, \{\omega_{17}\}, \{\omega_{19}\}, \{\omega_{20}\}, \{\omega_{21}\}, \{\omega_{22}\}, \{\omega_{23}\}, \{\omega_{24}\}, \{\omega_{25}\}, \{\omega_{26}\}, \{\omega_{27}\}, \{\omega_{28}\}, \{\omega_{29}\},$
$\{\omega_{30}\}, \{\omega_{31}\}, \{\omega_{32}\}, \{\omega_{33}\}, \{\omega_{34}\}, \{\omega_{35}\}, \{\omega_{36}\}\}$

Deuxième étape

Pour chaque $\omega_i \in \Omega$ on se propose de construire une base de voisinage ; $B(\omega_i, q)$ en choisissant $q = 14$. Nous trouvons les bases des voisinages suivantes :

$B(\omega_1, 14) = \{\omega_1, \omega_9\}$
$B(\omega_2, 14) = \{\omega_2\}$
$B(\omega_3, 14) = \{\omega_3, \omega_{18}\}$
$B(\omega_4, 14) = \{\omega_4, \omega_{12}\}$
$B(\omega_5, 14) = \{\omega_5\}$
$B(\omega_6, 14) = \{\omega_6\}$

$B(\omega_7, 14) = \{\omega_7, \omega_{15}, \omega_{30}, \omega_{32}, \omega_{35}\}$
$B(\omega_8, 14) = \{\omega_8\}$
$B(\omega_9, 14) = \{\omega_1, \omega_9\}$
$B(\omega_{10}, 14) = \{\omega_{10}\}$
$B(\omega_{11}, 14) = \{\omega_{11}\}$
$B(\omega_{12}, 14) = \{\omega_4, \omega_{12}, \omega_{19}\}$
$B(\omega_{13}, 14) = \{\omega_{13}, \omega_{16}\}$
$B(\omega_{14}, 14) = \{\omega_{14}\}$
$B(\omega_{15}, 14) = \{\omega_7, \omega_{15}, \omega_{30}, \omega_{35}\}$
$B(\omega_{16}, 14) = \{\omega_{13}, \omega_{16}\}$
$B(\omega_{17}, 14) = \{\omega_{17}\}$
$B(\omega_{18}, 14) = \{\omega_3, \omega_{18}\}$
$B(\omega_{19}, 14) = \{\omega_{12}, \omega_{19}\}$
$B(\omega_{20}, 14) = \{\omega_{20}\}$
$B(\omega_{21}, 14) = \{\omega_{21}\}$
$B(\omega_{22}, 14) = \{\omega_{22}\}$
$B(\omega_{23}, 14) = \{\omega_{23}\}$
$B(\omega_{24}, 14) = \{\omega_{24}\}$
$B(\omega_{25}, 14) = \{\omega_{25}\}$
$B(\omega_{26}, 14) = \{\omega_{26}\}$
$B(\omega_{27}, 14) = \{\omega_{27}\}$
$B(\omega_{28}, 14) = \{\omega_{28}\}$
$B(\omega_{29}, 14) = \{\omega_{29}, \omega_{30}\}$
$B(\omega_{30}, 14) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{35}\}$
$B(\omega_{31}, 14) = \{\omega_{31}, \omega_{35}\}$
$B(\omega_{32}, 14) = \{\omega_7, \omega_{32}\}$
$B(\omega_{33}, 14) = \{\omega_{33}\}$
$B(\omega_{34}, 14) = \{\omega_{34}\}$
$B(\omega_{35}, 14) = \{\omega_7, \omega_{15}, \omega_{30}, \omega_{31}, \omega_{35}\}$
$B(\omega_{36}, 14) = \{\omega_{36}\}$

En utilisant la même relation binaire R définie précédemment, nous construisons le graphe des relations entre les individus comme dans le figure (7.1)

Dans ce graphe des relations, nous pouvons grouper les individus qui forment un clique maximal dans une même classes. Dès qu'une classe se forme, on coupe les relations qui relient les individus qu'ils sont à l'intérieur de classe avec les individus qu'ils sont à l'ex-

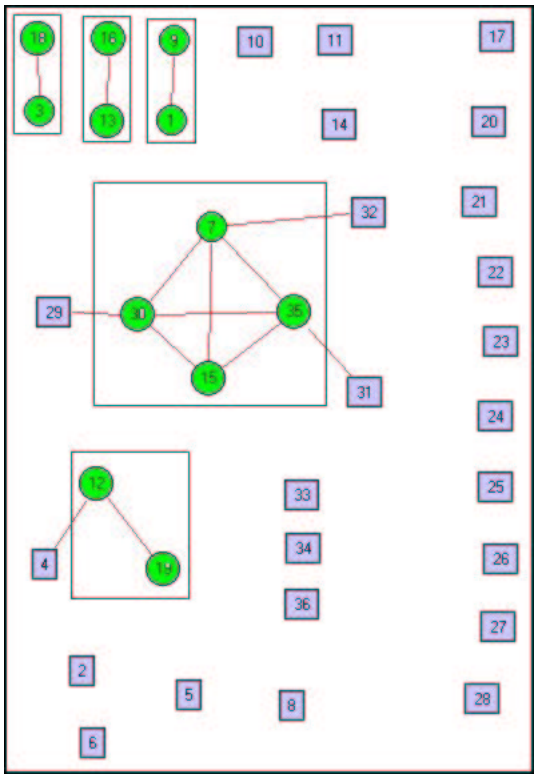

FIG. 7.1 – Les Cliques Maximales trouvés dans le graphe avec q=14

térieur de cette classe.

Finalement, la partition obtenue dans cette deuxième étape est :
$\mathcal{P}_{14} = \{\{\omega_1, \omega_9\}, \{\omega_2\}, \{\omega_3, \omega_{18}\}, \{\omega_4\}, \{\omega_5\}, \{\omega_6\}, \{\omega_7, \omega_{15}, \omega_{30}, \omega_{35}\}, \{\omega_8\}, \{\omega_{10}\}, \{\omega_{11}\}, \{\omega_{12,\omega_{19}}\}, \{\omega_{13}, \omega_{16}\}, \{\omega_{14}\}, \{\omega_{17}\}, \{\omega_{20}\}, \{\omega_{21}\}, \{\omega_{22}\}, \{\omega_{23}\}, \{\omega_{24}\}, \{\omega_{25}\}, \{\omega_{26}\}, \{\omega_{27}\}, \{\omega_{28}\}, \{\omega_{29}\}, \{\omega_{31}\}, \{\omega_{32}\}, \{\omega_{33}\}, \{\omega_{34}\}, \{\omega_{36}\}\}$

Nous pouvons continuer de la même manière jusqu'au moment qu'on trouve un nombre des classes qui nous convient ou on s'arrête à une valeur de $q > \frac{n}{2} + 1$.

Nous allons continuer cette exemple avec d'autre valeurs de q :

Troisième étape
Dans le cas où $q = 13$, nous trouvons les bases des voisinages suivantes :

$B(\omega_1, 13) = \{\omega_1, \omega_9\}$

$B(\omega_2, 13) = \{\omega_2, \omega_3, \omega_{18}\}$
$B(\omega_3, 13) = \{\omega_2, \omega_3, \omega_{18}\}$
$B(\omega_4, 13) = \{\omega_4, \omega_{12}, \omega_{19}, \omega_{34}\}$
$B(\omega_5, 13) = \{\omega_5\}$
$B(\omega_6, 13) = \{\omega_6\}$
$B(\omega_7, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}$
$B(\omega_8, 13) = \{\omega_8\}$
$B(\omega_9, 13) = \{\omega_1, \omega_9\}$
$B(\omega_{10}, 13) = \{\omega_{10}, \omega_{28}\}$
$B(\omega_{11}, 13) = \{\omega_{11}, \omega_{28}\}$
$B(\omega_{12}, 13) = \{\omega_4, \omega_{12}, \omega_{19}\}$
$B(\omega_{13}, 13) = \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\}$
$B(\omega_{14}, 13) = \{\omega_{14}\}$
$B(\omega_{15}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{35}\}$
$B(\omega_{16}, 13) = \{\omega_{13}, \omega_{16}\}$
$B(\omega_{17}, 13) = \{\omega_{17}, \omega_{32}\}$
$B(\omega_{18}, 13) = \{\omega_2, \omega_3, \omega_{18}\}$
$B(\omega_{19}, 13) = \{\omega_4, \omega_{12}, \omega_{19}, \omega_{36}\}$
$B(\omega_{20}, 13) = \{\omega_{20}\}$
$B(\omega_{21}, 13) = \{\omega_{21}\}$
$B(\omega_{22}, 13) = \{\omega_{22}, \omega_{25}\}$
$B(\omega_{23}, 13) = \{\omega_{23}\}$
$B(\omega_{24}, 13) = \{\omega_{24}\}$
$B(\omega_{25}, 13) = \{\omega_{22}, \omega_{25}\}$
$B(\omega_{26}, 13) = \{\omega_{26}\}$
$B(\omega_{27}, 13) = \{\omega_{13}, \omega_{27}, \omega_{33}\}$
$B(\omega_{28}, 13) = \{\omega_{10}, \omega_{11}, \omega_{28}\}$
$B(\omega_{29}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{35}\}$
$B(\omega_{30}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}$
$B(\omega_{31}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{35}\}$
$B(\omega_{32}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{32}, \omega_{35}\}$
$B(\omega_{33}, 13) = \{\omega_{13}, \omega_{27}, \omega_{33}\}$
$B(\omega_{34}, 13) = \{\omega_4, \omega_{34}\}$
$B(\omega_{35}, 13) = \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{32}, \omega_{35}\}$
$B(\omega_{36}, 13) = \{\omega_{19}, \omega_{36}\}$

Le graphe des relations entre les individus est donné dans le figure (7.2)

La partition obtenue dans cette étape est :

$\mathcal{P}_{13} = \{\{\omega_1, \omega_9\}, \{\omega_2\}, \{\omega_3, \omega_{18}\}, \{\omega_4, \omega_{12}, \omega_{19}\}, \{\omega_5\}, \{\omega_6\}, \{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{35}\}, \{\omega_8\}, \{\omega_{10}, \omega_{28}\}, \{\omega_{11}\}, \{\omega_{13}, \omega_{27}, \omega_{33}\}, \{\omega_{16}\}, \{\omega_{14}\}, \{\omega_{17}, \omega_{32}\}, \{\omega_{20}\}, \{\omega_{21}\}, \{\omega_{22}, \omega_{25}\}, \{\omega_{23}\}\{\omega_{24}\}, \{\omega_{26}\}, \{\omega_{34}\}, \{\omega_{36}\}\}$

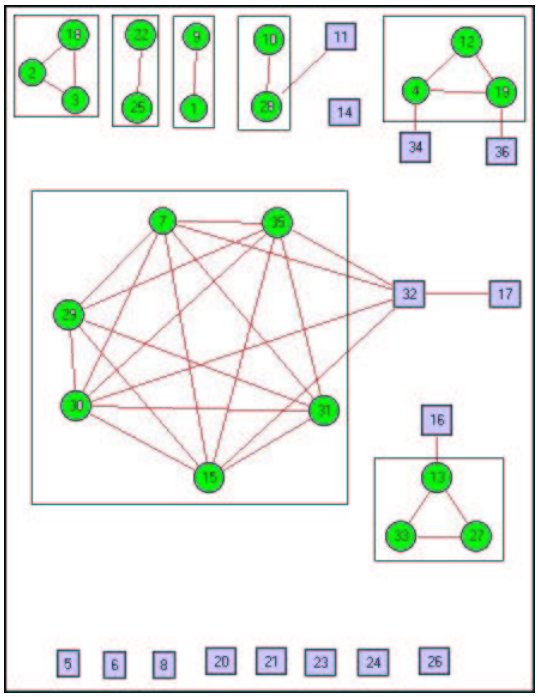

FIG. 7.2 – Les Cliques Maximales trouvés dans le graphe avec q=13

Dans le cas où q=12

Le graphe des relations entre les individus est donné dans le figure (7.3)

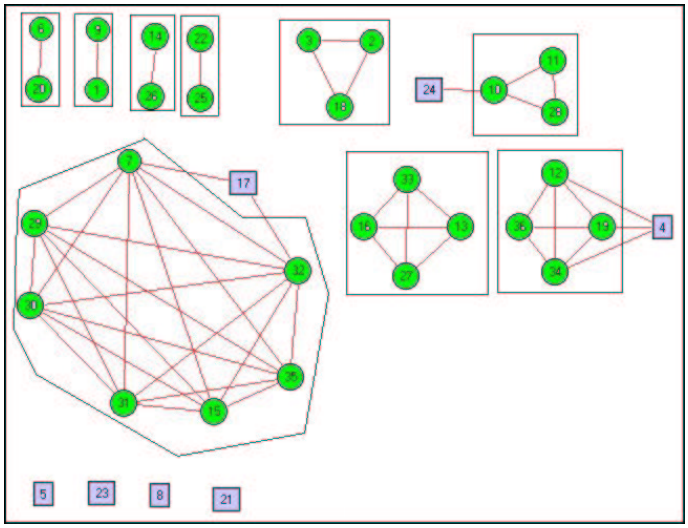

FIG. 7.3 – Les Cliques Maximales trouvés dans le graphe avec q=12

La partition obtenue dans cette étape est :

$\mathcal{P}_{12} = \{\{\omega_1, \omega_9\}, \{\omega_2, \omega_3, \omega_{18}\}, \{\omega_4\}, \{\omega_5\}\{\omega_{12}, \omega_{19}, \omega_{34}, \omega_{36}\}, \{\omega_6, \omega_{20}\},$
$\{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}, \{\omega_8\}, \{\omega_{10}, \omega_{11}, \omega_{28}\}, \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\},$
$\{\omega_{14}, \omega_{26}\}, \{\omega_{17}\}, \{\omega_{21}\}, \{\omega_{22}, \omega_{25}\}, \{\omega_{23}\}, \{\omega_{24}\}\}$

Dans le cas où q=11

Le graphe des relations entre les individus est donné dans le figure (7.4)

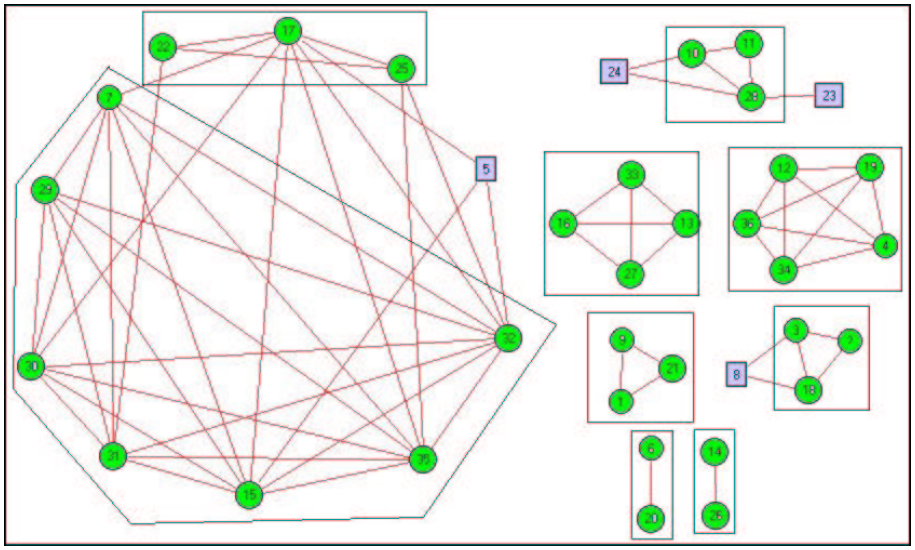

FIG. 7.4 – Les Cliques Maximales trouvés dans le graphe avec q=11

La partition obtenue dans cette étape est :

$\mathcal{P}_{11} = \{\{\omega_1, \omega_9, \omega_{21}\}, \{\omega_2, \omega_3, \omega_{18}\}, \{\omega_4, \omega_{12}, \omega_{19}, \omega_{34}, \omega_{36}\}, \{\omega_5\}, \{\omega_6, \omega_{20}\},$
$\{\omega_7, \omega_{15}, \omega_{29}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}, \{\omega_8\}, \{\omega_{10}, \omega_{11}, \omega_{28}\}, \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\},$
$\{\omega_{14}, \omega_{26}\}, \{\omega_{17}, \omega_{22}, \omega_{25}\}, \{\omega_{23}\}, \{\omega_{24}\}\}$

Dans le cas où q=10

Le graphe des relations entre les individus est donné dans le figure (7.5)

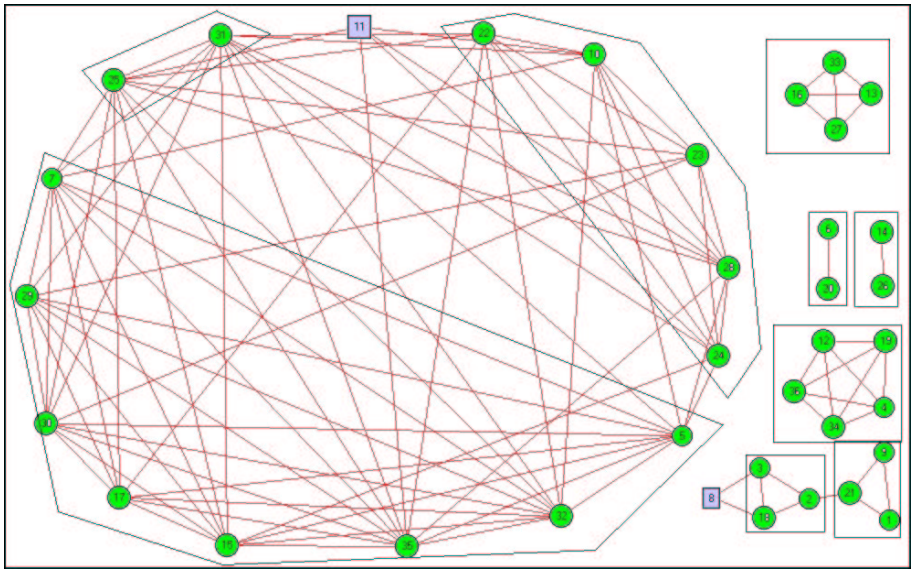

FIG. 7.5 – Les Cliques Maximales trouvés dans le graphe avec q=10

La partition obtenue dans cette étape est :

$\mathcal{P}_{10} = \{\{\omega_1, \omega_9, \omega_{21}\}, \{\omega_2, \omega_3, \omega_{18}\}, \{\omega_4, \omega_{12}, \omega_{19}, \omega_{34}, \omega_{36}\}, \{\omega_6, \omega_{20}\},$
$\{\omega_5, \omega_7, \omega_{15}, \omega_{17}, \omega_{29}, \omega_{30}, \omega_{32}, \omega_{35}\}, \{\omega_8\}, \{\omega_{10}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{28}\}, \{\omega_{11}\}, \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\},$
$\{\omega_{14}, \omega_{26}\}, \{\omega_{25}, \omega_{31}\}\}$

Dans le cas où q=9

La partition obtenue dans cette étape est :

$\mathcal{P}_9 = \{\{\omega_1, \omega_9, \omega_{21}\}, \{\omega_2, \omega_3, \omega_8, \omega_{18}\}, \{\omega_4, \omega_{12}, \omega_{19}, \omega_{34}, \omega_{36}\}, \{\omega_6, \omega_{20}\},$
$\{\omega_5, \omega_7, \omega_{10}, \omega_{11}, \omega_{15}, \omega_{22}, \omega_{24}, \omega_{25}, \omega_{28}, \omega_{30}, \omega_{31}, \omega_{32}, \omega_{35}\}, \{\omega_{23}\}, \{\omega_{13}, \omega_{16}, \omega_{27}, \omega_{33}\},$
$\{\omega_{14}, \omega_{26}\}, \{\omega_{17}, \omega_{29}\}\}$

Nous Allons arrêter l'algorithme à cette étape.

Nous pouvons constater que le résultat obtenu coïncide avec celui de Grötschel dans **Grötschel** dans [GW89].

Remarques :

- Nous pouvons remarquer que l'algorithme révise ces classe après chaque étape, d'une manière d'augmenter la qualité des classe.
- Le choix de $q = p$ donne une partition proche de la partition discrète en mettant un individu par classe ou plusieurs individus gémeaux (ont les même valeurs). Ce

choix peut être intéressent si nous voulons supprimer les doublons d'un ensemble des données.

- Le choix de q peut jouer un rôle important au niveau de nombre des classes ; (q est très proche de p donne un nombre des classe proche de n)

Chapitre 8

Conclusion et perspectives

Conclusion Dans ce travail, nous nous somme intéressés aux méthodes de classification au travers de trois approches.
Tout d'abord, nous avons tenté de fournir une présentation synthétique des méthodes de classification. Nous avons classé les méthodes de classification en plusieurs catégories. Ensuite nous avons présenté brièvement les principaux algorithmes de chacune de ces méthodes tout en essayant de montrer pour chaque méthode les points forts et les points faibles. Cette synthèse permettra aux lecteurs d'avoir une idée sur les méthodes de classification.

Dans la deuxième partie, nous avons développé trois différentes approches. La première approche est basée sur le principe de l'agrégation d'opinions. Un algorithme basé sur la méthode du recuit simulé a été proposé pour résoudre le problème d'optimisation rencontré. L'algorithme du recuit simulé comporte un aspect aléatoire, ce qui explique que deux exécutions successives sur le même exemple ne donnent pas forcément le même résultat. Pour cela nous appellerons "Résultat" de l'algorithme, le meilleur des résultats obtenus lors de plusieurs exécutions successives.
Le principal avantage de cette approche de classification, est que les données ne sont pas forcement quantitatives. Elle est basée sur l'opinion de l'expert du domaine, qui définit les fonctions de classement. Il faut noter aussi que cette méthode ne demande pas la fixation a priori du nombre de classes.

La deuxième approche que nous avons développé dans cette thèse est basée sur la prétopologie. Nous avons fait appel à la prétopologie comme un outil mathématique de modélisation du concept de proximité.
La théorie mathématique ayant pour objet la modélisation du concept de proximité est la

topologie ; qui relève les concepts classiques de :

- **Continuité :** Qui formalise le transport entre ensembles d'une structure topologique,
- **Compacité :** Qui combinée avec la notion de continuité permet de résoudre des problèmes d'existence de points particuliers tels que point fixe ou optimum,
- **Connexité :** Qui permet de modéliser le concept "d'homogénéité" d'une partie d'un ensemble donné.

En fait, dans de nombreuses situations pratiques, les utilisateurs se restreignent, dans leurs modèles, à l'utilisation de structures métriques, moins générales mais plus commodes à manipuler. Mais les exigences axiomatiques de la topologie - et a fortiori de la métrique - sont telles qu'elles sont souvent peu compatibles avec les réalités dont on cherche à modéliser le concept de proximité. D'où l'idée d'envisager la construction d'une théorie ayant une axiomatique moins contraignante que celle de la topologie : c'est ce que propose la prétopologie.

Nous avons consacré un paragraphe pour montrer la contribution concernant des développements du modèle prétopologique. Un algorithme à l'aide de la théorie des graphes à été proposé pour trouver une solution du problème de classification.

Dans la troisième approche, nous avons proposé d'analyser le problème de la classification à la manière des modèles d'agrégation des préférences. Notre objectif n'était pas de développer une méthode originale de classification mais plutôt d'interpréter le problème de fond de la classification par les concepts de l'agrégation des préférences et de produire des résultats qui permettent de qualifier la qualité d'une procédure de classification et, peut être d'introduire dans cette procédure une part d'intervention humaine permettant d'en contrôler le déroulement.

L'idée qui sous-tend cette approche s'appuie sur la constatation qu'il est en général aisé de classifier une population à partir de la donnée des observations d'une seule variable. En effet, les diverses méthodes habituellement utilisées de classification fournissent, la plupart du temps, des résultats quasiment identiques lorsqu'on les applique dans des situations où n'intervient qu'une seule variable. Il est alors naturel de se dire que lorsque k variables $X_1, X_2, \ldots, X_k$ sont en jeu, on obtient k classifications associées à chacune de ces variables et qu'obtenir une classification finale revient à agréger ces k classifications en une seule. On se ramène ainsi à un problème d'agrégation. Nous nous sommes situé dans cette lignée tout en notant que le principal résultat est qu'il n'existe pas de méthode d'agrégation,

mais des méthodes d'agrégation. Partant de ce constat, nous avons souhaité produire une réflexion s'appuyant sur le parallélisme qui existe entre les deux problèmes : classification et agrégation pour obtenir des résultats qui peuvent éclairer, sur le fond, l'utilisateur des méthodes de classification dans sa démarche et l'aider à se poser des questions.

perspectives :

Nous nous somme rendu compte qu'au cours de nos différentes expérimentations que quelques améliorations peuvent rendre ce travail plus général.

Nous avons remarqué pour notre première approche (agrégation de l'opinion), que les réglages du recuit simulé peuvent être pertinents (il aurait été intéressant d'indiquer dans quelle mesure les résultats sont sensibles aux réglages des paramètres), en particulier le calcul de la température initiale, qui est un peu élevée.

Une extension de la méthode utilisée dans notre deuxième approche vers d'autres applications comme l'analyse des images, l'aide à la décision et l'analyse multicritères peuvent rendre cette démarche plus intéressante.

Les motivations de notre démarche, dans la troisième approche, sont orientées vers la mise en évidence des liens théoriques entre la classification et l'agrégation des préférences. Ces liens théorique sont à améliorer et à compléter pour arriver à un algorithme ou une méthode de classification.

Index

adhérence, 100
Algorithme, 37
    ANTCLASS, 65
    BIRCH, 38
    centres mobiles, 46
    CLARA, 50
    CLARANS, 51
    CLIQUE, 63
    COBWEB, 64
    CURE, 37
    DBSCAN, 58
    DENCLUE, 60
    GGA, 65
    k-means, 45, 46
    k-medoids, 49
    nuées dynamiques, 47
    OPTICS, 60
    PAM, 50
    Recuit simulé, 83
    ROCK, 39
    SOM, 64
    SPC, 65
    STING, 61
    TAXY, 53
    TSVQ, 43
    WaveCluster, 62
    Williams et Lambert, 41

CAH, 35
CHD, 40
classification automatique, 25
classification hiérarchiques, 34
Cohérence, 90
concordance, 77
continuité, 112

Ensemble de variables décisives, 136
ensemble de variables décisives essentielles, 148

filtre, 101
Fonction d'agrégation, 134
Fonction de classement élémentaire , 133
Fonction de classification, 134

hiérarchie de parties, 24
hiérarchie indicée de parties, 24

indice de dissimilarité, 27
Indice de Rand, 66
indice de similarité, 27
intérieur, 100

mesures de homogénéité, 69
méthodes de classification, 34

partition, 23
propriété, 137
    Hf, 147
    Hi, 137
    Hi2, 140
    Hu, 138
    symétrie, 134
    transitivité, 135

préfiltre, 101
    dégénéré, 101
    engendré, 101
    propre, 101
    stable, 101
prétopologie, 7, 98
    induite, 114
    type V, 100
    type VD, 100
    type VDS, 100
prévoisinages, 102

recouvrement, 23
ressemblance, 89

saut maximal, 35
saut minimal, 35

taxinomie, 25
topologie, 100
topologie-classe, 107, 117

voisinage d'une partition, 83

Bibliographie

[ABF+95] R. Agarwala, V. Bafina, M. Farach, B. Narayanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy. Technical Report 95-46, DIMACS, Rutgers University, Piscataway, NJ 08855, USA, 1995.

[ABKS99] M. Ankerst, M.M. Breunig, H.P. Kriegel, and J. Sander. Optics : Ordering points to identify the clustering structure. In *SIGMOD Conference*, pages 49–60, 1999.

[AC92] C. Alippi and R. Cucchiara. Cluster partitioning in image analysis classification : a genetic algorithm approach. In *Proc. CompEuro 92*, pages 139–144. IEEE Computer Society Press, 1992.

[AF51] A. Appert and K. Fan. *Espaces topologiques intermédiaires.* Hermann, 1951.

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. pages 94–105, 1998.

[AGK82] D. Art, R. Gnanadesikan, and R. Kettenring. Data-based metrics for cluster analysis. *Utilitas Mathematica*, 21(A) :75–99, 1982.

[AH96] P. Arabie and L.J. Hubert. An overview of combinatorial data analysis. *Clustering and Classification*, pages 5–63, 1996.

[And73] M.R. Anderberg. *Cluster Analysis for Applications.* Academic Press, New York, 1973.

[Arc93] M. Archoun. *Modélisation prétopologique de la segmentation par croissance de régions des images à niveaux de gris.* PhD thesis, Université. Claud Bernard Lyon I, 1993.

[Arn79] S.J. Arnold. A test for clusters. *Journal of Marketing Research*, 16(4) :545–551, Nov 1979.

[Arr63] K.J. Arrow. *Social Choice and Individual Values*. John Wiley & Sons, New York, second edition, 1963.

[AS95] K.S. Al-Sultan. A tabu search approach to the clustering problem. *Pattern Recognition*, 28(9) :1443–1451, 1995.

[Aur82] J.P. Auray. *Contribution à l'étude des structures pauvres.* PhD thesis, Université Claud Bernard Lyon I, 1982.

[Bäc96] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[BBHB94] J.C. Bezdek, S. Boggavaparu, L.O. Hall, and A. Bensaid. Genetic algorithm guided clustering. In D.B. Fogel, editor, *Proceedings of The First IEEE Conference on Evolutionary Computation*, volume I, pages 34–40. IEEE Computer Society Press, 1994.

[BBL07] M. Boubou, A. Bounekkar, and M. Lamure. Classification basée sur l'agrégation d'opinions par la méthode de recuit simulé. In *XIVe Rencontre de la Société Francophone de Classification (SFC'07)*, September 2007.

[BBTL05] M. Boubou, A. Bounekkar, D. Tounissoux, and M. Lamure. Utilisation du recuit simulé pour la recherche d'une ultramétrique optimale. In *XIIe Rencontre de la Société Francophone de Classification (SFC'05)*, 2005.

[BCRR95] J.D. Bandfield, G. Celeux, A.E. Raftery, and C.P. Robert. Inference in model-based cluster analysis. Technical Report 285, Dpto. of Statistics, University of Washington, 1995.

[BCRR97] H. Bensmail, B. Celeux, A. Raftery, and C. Robert. Inference in model-based cluster analysis. *Biometrics*, 7 :1–10, 1997.

[BD85] P. Bretrand and E. Diday. A visual representation of the compatibility between an order and a dissimilarity index : the pyramids. *Computational Statistics Quarterly*, 2(1) :31–41, 1985.

[BD00] H.H. Bock and E. Diday. *Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data*. Springer Verlag, 2000.

[BDSY99] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4) :281–297, 1999.

[Bel93] Z. Belmandt. *Manuel de prétopologie et ses applications*. Hermès, 1993.

[Ben73] J.P Benzecri. *L'Analyse des données*, volume Tome 1 : La taxinomie. Dunod, 1973.

[Ber86] P. Bertrand. *Etude de la Répresentation Pyramidale*. Thèse de Doctorat. Université de Paris-Duphine, 1986.

[Bez74] J.C. Bezdeck. Numerical taxonomy with fuzzy sets. *Journal of Mathematical Biology*, 1 :57–71, 1974.

[BFR98] S.P. Bradley, U.M. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Knowledge Discovery and Data Mining*, pages 9–15, 1998.

[BH67] G.H. Ball and D.J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12 :153–155, 1967.

[BK86] N. Benamou and Y. Kodratoff. Conceptual hierarchical ascending classification. Technical report, LRI, Univ. Paris-Sud, 1986.

[BL95] J.P. Barthélemy and B. Leclerc. The median procedure for partitions. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 19 :3–33, 1995.

[BLCM96] P. Bertrand, Y. Lechevallier, M. Csernel, and A. Morineau. A·pyros : a pyramidal representation of data. In *IFCS-96, Proceedings of the fifth Conference of International Federation of Classification Societies*, volume I, pages 126–128. Kobe, Japón, 1996.

[BM94] G.P. Babu and M.N. Murty. Clustering with evolution strategies. *Pattern Recognition*, 27(2) :321–329, 1994.

[BP92] J.C. Bezdeck and S.K. Pal. *Fuzzy Models for Pattern Recognition*. IEEE Press, 1992.

[BR93] J.D. Bandfield and A.E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49 :803–821, 1993.

[Bri75] M. Brissaud. Les espaces prétopologiques. *C. R. Aead Sci*, 280 :705–708, 1975.

[Bri91] P. Brito. *Analyse des données symboliques et pyramides d'héritage*. PhD thesis, Université Paris IX-Dauphine, 1991.

[BWD97] M. Blatt, S. Wiseman, and E. Domany. Data clustering using a model granular magnet. *Neural Computation*, 9(8) :1805–1842, 1997.

[BWL95] G. Biswas, J. Weinberg, and C. Li. *A Conceptual Clustering Method for Knowledge Discovery in Databases*. ditions Technip, 1995.

[CBT04] S. Chelcea, P. Bertrand, and B. Trousse. Un nouvel algorithme de classification ascendante 2-3 hiérarchique. In *Actes de 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2004)*, volume 3, pages 1471–1480, Centre de Congrès Pierre BAUDIS, Toulouse, France, 28-30 January 2004.

[CC94] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1994.

[CDLR81] G. Celeux, E. Diday, Y. Lechevallier, and H. Ralambondrainy. *Classification automatique des données*. Dunod, 1981.

[CDS84] J.L. Chandon and G. De-Soete. Fitting a least squares ultrametric to dissimilarity data : approximation versus optimization. In E. Diday and col., editors, *Data Analysis and Informatics, III*, pages 213–221. North-Holland, Amsterdam, 1984.

[Cec66] E. Cech. *Topological spaces*. Wiley and son, 1966.

[CG95] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5) :781–793, 1995.

[Cha97] M. Chavent. *Analyse des données symboliques, Une méthode divisive de classification*. PhD thesis, Université Paris IX-Dauphine, 1997.

[CLP80] J.L. Chandon, J. Lemaire, and J. Pouget. Construction de l'ultramétrique la plus proche d'une dissimilarité au sens des moindres carrés. *Revue d'Automatique, d'Informatique et de Recherche Opérationnelle, RAIRO*, 14(2) :157 – 170, 1980.

[CM88] M.C. Cooper and G.W. Milligan. The effect of measurement error on determining the number of clusters in clusteranalysis. pages 319–328, 1988.

[CMS99] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1) :5–32, 1999.

[CP80] J.D. Caroll and S. Pruzansky. Discrete and hybrid scaling models. In E.D. Lenterman and H. Feger, editors, *Similarity and Choice*, pages 108–139. Huber, Bern, 1980.

[CS96] G. Celeux and G. Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2) :195–212, 1996.

[Cuc93] R. Cucchiara. Analysis and comparison of different genetic models for the clustering problem in image analysis. In R.F. Albretch, C.R. Reeves, and N.C. Steele, editors, *Artificial Neural Networks and Genetic Algorithms*, pages 423–427. Springer-Verlag, 1993.

[DABR92] S.G. De-Amorim, J.P. Barthélemy, and C.C. Ribeiro. Clustering and clique partitioning : simulated annealing and tabu search approaches. *Journal of Classification*, 9 :17–41, 1992.

[Dav70] H.A. David. *Order Statistics*. John Wiley & Sons, Inc., 1970.

[DFG01] I. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In G. Kamath R. Grossman and R. Naburu, editors, *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers, 2001.

[DFS02] Sandrine Dudoit, Jane Fridlyand, and Terence P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457) :77–87, 2002.

[DGLS80] E. Diday, G. Govaert, Y. Lechevallier, and J. Sidi. Clustering in pattern recognition. In *NATO Advanced study Institute on Digital Image Processing and Analysis, Bonas.*, 1980.

[DH73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[Did82] E. Diday. Crossings, orders and ultrametrics : Application to visualization of consensus for comparing classification. In *Proceedings in Computer Statistics*, pages 186–191. Physica-Verlag, 1982.

[Did86] E. Diday. Orders and overlapping clusters by pyramids. In *Multidimensional Data Analysis*, pages 201–234. DSWO Press, 1986.

[DKBM00] E. Diday, Y. Kodratoff, P. Brito, and M. Moulet. *Induction symbolique numérique à partir de données*. Cepadues, 2000.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1) :1–38, 1977.

[DM05] E. Diday and N. Murty. *"Symbolic Data Clustering" in Encyclopedia of Data Warehousing and Mining*. Idea Group, 2005.

[Dre86] R.M. Dreger. Microcomputer programs for the rand index of cluster similarity. *Educational and Psychological Measurement*, 46(3) :655–661, 1986.

[DS84] G. De-Soete. A least square algorithm for fitting an ultrametric tree to a dissimilarity matrix. *Pattern Recognition Letters*, 2 :133–137, 1984.

[Dur80] G. Duru. *Contribution à l'étude des structures des systèmes complexes dans les sciences humaines*. PhD thesis, Université. Claud Bernard Lyon I, 1980.

[ECS65] A.W.F. Edwards and L.L. Cavalli-Sforza. A method for cluster analysis. *Biometrics*, 21 :363–375, 1965.

[EF79] H. Emptoz and R. Fages. un approche ensembliste des problèmes de classification - une application à l'écologie. In *Journées internationales sur les développements récents en reconnaissance des formes - Méthodes et applications*, 1979.

[EF80] H. Emptoz and R. Fages. A set function to cluster analysis. In *EUSIPCO (1er congrès de l'association européenne de traitement de signal*, 1980.

[EFKS00] M. Ester, A. Frommelt, H. Kriegel, and J. Sander. Spatial data mining : Database primitives, algorithms and efficient dbms support, 2000.

[EKSX96] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[EMO96] W.F. Eddy, A. Mockus, and S. Oue. Approximate single linkage cluster analysis of large data sets in high-dimensional spaces. *Comput. Stat. Data Anal.*, 23(1) :29–43, 1996.

[Emp83] H. Emptoz. *Modèle prétopologique pour la reconnaissance des formes. Applications en neurophysiologie*. PhD thesis, Université. Claud Bernard Lyon I, 1983.

[Eve79] B.S. Everitt. Unresolved problems in cluster analysis. *Biometrics*, 35(1) :169–181, Mar 1979.

[Eve93] B.S. Everitt. *Cluster Analysis (3rd ed.)*. Edward Arnold, 1993.

[Fab94] V. Faber. Clustering and the continuous k-means algorithm. 1994.

[Fag74] R. Fages. Programme de classification automatique taxi et taco. Technical report, Rapport interne S.A.N.T.I. Université Lyon I, 1974.

[Fag78] R. Fages. la dispersion en classification automatique. In *Communication aux journées sur la Classification, Nice*, 1978.

[Fas99] D. Fasulo. An analysis of recent work on clustering algorithms, 1999.

[Fis87a] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 1987.

[Fis87b] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2 :139–172, 1987.

[Fle81] R. Fletcher. *Practical Methods of Optimization, Vol. 2*. John Wiley & Sons, 1981.

[For65] E.W. Forgy. Cluster analysis of multivariate data : efficency versus interpretability of classifications (abstract). *Biometrics*, 21 :768–769, 1965.

[Fre28] M. Frechet. *Espaces abstraits*. Hermann, 1928.

[FS92] M. Fraboni and R. Salstone. The wais-r number-of-factors quandary : A cluster analytic approach to construct validation. *Educational and Psychological Measurement*, 1992.

[FWZ01] A. Foss, W. Wang, and O. Zaane. A non-parametric approach to web log analysis, 2001.

[GG92] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer, Boston, 1992.

[GGR99] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS- clustering categorical data using summaries. In *KDD '99 : Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83, New York, NY, USA, 1999. ACM Press.

[Gir97] R. Girard. *Classification conceptuelle sur des données arborescentes et imprécises*. PhD thesis, Université de la Réunion, 1997.

[GKR00] D. Gibson, J.M. Kleinberg, and P. Raghavan. Clustering categorical data : An approach based on dynamical systems. *VLDB Journal : Very Large Data Bases*, 8(3–4) :222–236, 2000.

[Gor81] A.D. Gordon. *Classification. Methods for the exploratory analysis of multivariate data*. Chapman & Hall, 1981.

[Gor87] A.D. Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society, Series A*, 150(2) :119–137, 1987.

[Gor98] A.D. Gordon. Cluster validation. In C. Hayashi N. Ohsumi K. Yajima Y Tanaka H.H. Bock y Y. Baba, editor, *Data Science, Classification and Related Methods*, pages 22–39. Springer, 1998.

[GRS98] S. Guha, R. Rastogi, and K. Shim. Cure : an efficient clustering algorithm for large databases. In *SIGMOD '98 : Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84, New York, NY, USA, 1998. ACM Press.

[GRS00] S. Guha, R. Rastogi, and K. Shim. ROCK : A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5) :345–366, 2000.

[GW89] M. Grötschel and Y. Wakbayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming Series*, B 45 :59–96, 1989.

[Ham64] P.C. Hammer. Topologies of approximation. *Journal of the Society for Industrial and Applied Mathematics*, 1 :69–75, 1964.

[Har67] J.A. Hartigan. Representation of dissimilarity matrices by trees. *J. Amer. Statist. Ass.*, 62 :1140–1158, 1967.

[Har75] J.A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[Har85] J.A. Hartigan. Statistical theory in clustering. *Journal of Classification*, 2 :63–76, 1985.

[Has82] M.A. Hashom. *Plus proches voisins et classification automatique. Application à des données industrielles*. PhD thesis, Institut Nationale des sciences Appliquées de Lyon, 1982.

[HC01] J. Heer and E. Chi. Identification of web user traffic composition using multi-modal clustering and information scent, 2001.

[HE03] G. Hamerly and C. Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, volume 17, 2003.

[HK98] A. Hinneburg and D.A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *KDD*, pages 58–65, 1998.

[HK01] J. Han and M. Kamber. *Data Mining : Concepts and Techniques*. Morgan Kaufmann, 2001.

[HLG98] G. Harik, F.G. Lobo, and D.E. Goldberg. The compact genetic algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 523–528, 1998.

[HÖB99] L.O. Hall, I.B. Özyurt, and J.C. Bezdek. Clustering with a genetically optimized approach. *IEEE Trans. Evolutionary Computation*, 3(2) :103–112, 1999.

[HR82] A. Hardy and J.P. Rasson. Une nouvelle approche des problèmes de classification automatique. *Statistiques et Analyse des Données*, 7(2) :41–56, 1982.

[HSC90] R. Hanson, J. Stutz, and P. Cheesman. Bayesian classification theory. Technical Report FIA-90-12-7-01, NASA, Ames Research Center, 1990.

[Hub73] L.J. Hubert. Monotone invariant clustering procedures. *Psychometrika*, 38 :47–62, 1973.

[Jam99] M. Jambu. *Méthodes de base de l'analyse des données*. Eyrolles, 1999.

[JB91] D.R. Jones and M.A. Beltramo. Solving partitioning problems with genetic algorithms. In R. Belew and L. B. Booker, editors, *Proc. of the Fourth International Conference on Genetic Algorithms*, pages 442–449. Morgan Kaufmann, 1991.

[JD88] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[JJS67] C.J. Jardine, N. Jardine, and R. Sibson. The structure and construction of taxonomic hierarchies. *Math. Biosci.*, 1 :171–179, 1967.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999.

[Joh67] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32 :241–245, 1967.

[Kau75] A. Kaufmann. *Introduction à la théorie des sous-ensembles flous :Applications à la classification et à la reconnaissance des formes, aux automates et aux systèmes, au choix des critères*, volume 3. Masson, 1975.

[KD89] R.W. Kelin and R.C. Dubes. Experiments in projection and clustering by simulated annealing. *Pattern Recognition*, 22 :213–220, 1989.

[Ken66] M.G. Kendall. Discirmination and classification. In P.R. Krishnaiah, editor, *Multivariate Analysis*, pages 165–185. Academic Press, New York, 1966.

[KGV82] S. Kirpaktrick, C.D.Jr. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. Technical Report RC 9355, IBM Research Report, 1982.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598 :671–680, 1983.

[KL88] W.J. Krzanowski and Y.T. Lai. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, 44(1) :23–34, Mar 1988.

[Kla83] T.D. Klastorin. Assessing cluster analysis results. *Journal of Marketing Research*, 20(1) :92–98, Feb 1983.

[Kod92] Y. Kodratoff. Symbolic or numeric induction ? In *Journées Symboliques Numériques*, pages 31–50, 1992.

[Koh82] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 :59–69, 1982.

[Koh88] T. Kohonen. Self-organized formation of topologically correct feature maps. pages 509–521, 1988.

[Koh00] T. Kohonen. *Self-organizing Maps*. Springer Series in Information Sciences, 3 edition, Dec 2000.

[KR90] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data*. John Wiley & Sons, Inc., 1990.

[Lag95] J. Lagarde. *Initiation à l'analyse des données.* Dunod, 1995.

[Lam87] M. Lamure. *Espaces abstraits et reconnaissance de formes. Application au traitement des images digitales*. PhD thesis, Université. Claud Bernard Lyon I, 1987.

[LDK93] C.B. Lucasius, A.D. Dane, and G. Kateman. On k-medoid clustering of large data sets with the aid of a genetic algorithm : background, feasibility and comparison. *Analytica Chimica Acta*, 282 :647–669, 1993.

[Ler81] I.C. Lerman. *Classification et analyse ordinale des données*. Dunod, Paris, 1981.

[Les91] C. Leschi. *Aspects fondamentaux et évolutifs de la reconnaissance de formes*. PhD thesis, Institut Nationale des sciences Appliquées de Lyon, 1991.

[LH74] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[LKH93] F.T. Lin, C.Y. Kao, and C.C. Hsu. Applying the genetic approach to simulated annealing in solving some NP-hard problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6) :1752–1767, 1993.

[LLdo] J.A. Lozano and P. Larrañaga. Applying genetic algorithms to search for the best hierarchical clustering of a dataset. *Pattern Recognition Letters*, Enviado.

[LNP88] I.C. Lerman, J. Nicolas, and P. Peter. Classification conceptuele : une approche dans la combinaison de méthodes numériques et symboliques". In Actes des 2ème journées "Symboliques Numériques" pour l'apprentissage de connaissances à partir des données, editor, *E. Diday and Y. Kodratoff*, pages 79–93. LRI, Univ. de Paris-sud, Orsay, 1988.

[MA84] L.C. Morey and A. Agresti. The measurement of classification agreement : An adjustment to the rand statistics for chance agreement. *Educational and Psychological Measurement*, 44(1) :33–37, 1984.

[Mac67] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium*, volume 2, pages 281–297, 1967.

[Man88] M. Manago. *Intégration de techniques numériques et symboliques en apprentissage automatique*. PhD thesis, Université d'Orsay, 1988.

[MC85] G.W. Milligan and M.C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2) :159–179, 1985.

[MD88] M.O. Menessier and E. Diday. Approche symbolique pour la prévision de séries chronologiques pseudo-périodiques. In *Actes des journées "Symboliques - Numérique" pour l'apprentissage de connaissances à partir d'observations.* Y. Kodratoff and E. Diday, 1988.

[Mir96] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.

[Mit96] M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, Massachusetts, 1996.

[MK83] L. Massart and L. Kaufman. *Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*. Wiley, 1983.

[MM82] J.F. Marcotorchino and P. Michaud. Agrégation de similarités en classification automatique. *Revue de Statistique Appliquée*, 30(2), 1982.

[MS83a] R.S. Michalski and R.E. Stepp. Automated construccion of classifications : Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5 :219–243, 1983.

[MS83b] R.S. Michalski and R.E. Stepp. Learning from observation : Conceptual clustering. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning : An artificial intelligence approach*. Morgan Kaufmann, Los Altos, CA, 1983.

[MSD81] R.S. Michalski, R.E. Stepp, and E. Diday. A recent advances in data analysis : clustering objets into classes characterized by conjonctive concepts. *Progress in Pattern Recognition*, 1, 1981.

[MSWDM64] P. Macnaughton-Smith, W.T. Williams, M.B. Dale, and L.G. Mockett. Dissimilarity analysis : a new technique of hierarchical sub-division. *Nature*, 202 :1034–1035, 1964.

[MVS99] N. Monmarché, G. Venturini, and M. Slimane. Classification non supervisée par une population de fourmis artificielles. In *Actes des Colloques Insectes Sociaux*, volume 13, pages 43–52, Tours, France, 1-3 Septembre 1999.

[NH94] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In Jorgeesh Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile proceedings*, pages 144–155, Los Altos, CA 94022, USA, 1994. Morgan Kaufmann Publishers.

[NH02] R.T. Ng and J. Han. Clarans : A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.*, 14(5) :1003–1016, 2002.

[Nic88] N. Nicoloyannis. *Structure prétopologiques et classification automatique : le logiciel DEMON*. PhD thesis, Université. Claud Bernard Lyon I, 1988.

[OH05] O. Otto and P. Henning. Aggregation of ordinal judgements based on condorcet's majority rule. *Data Analysis and Decision Support*, I :108–112, 2005.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2) :559–572, 1901.

[Pol98] G. Polaillon. *Organisation et interprétation par les treillis de Galois de données de type multivalué, intervalle ou histogramme*. PhD thesis, Université Paris IX-Dauphine, 1998.

[Ran71] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336) :846–850, Dec 1971.

[RB79] V.V. Raghadan and J. Bhuyan. A clustering strategy based on a formalism of the reproductive process in natural systems. *SIGIR Forum*, 14 :10–22, 1979.

[Ree93] C.R. Reeves. *Modern Heuristic Techniques for Combinatorial Optimization*. Blackwell Scientific Publications, 1993.

[Reg65] S. Regnier. Sur quelques aspects mathématiques des problèmes de classification automatique. *I.C.C. Bulletin*, 175(4), 1965.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14 :465–471, 1978.

[RK93] J.P. Rasson and T. Kubushishi. The gap test : an optimal method for determining the number of natural classes in cluster analysis. In E. Diday,

Y. Lechevallier, M. Schader, P. Bertrand, and B. Burtschy, editors, *New Approaches in Classification and Data Analysis*, pages 186–193. Springer-Verlag, 1993.

[Rud94] G. Rudolph. Massively parallel simulated annealing and its relation to evolutionary algorithms. *Evolutionary Computation*, 1(4) :361–382, 1994.

[Sap90] G. Saporta. *Probabilités analyse des données et statistique*. Technip, 1990.

[Sar82] J.T. Sarsoh. *Fonctions structurantes et classification automatique. Application à des problèmes de génétique*. PhD thesis, Université. Claud Bernard Lyon I, 1982.

[SC04] R. Souza and F. Carvalho. Clustering of interval data based on city-block distances. *Pattern Recogn. Lett.*, 25(3) :353–365, 2004.

[Sch78] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 :461–464, 1978.

[Sch81] H.P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Inc., 1981.

[Sch97] P.C. Schuur. Classification of acceptance criteria for the simulated annealing algorithm. *Mathematics of operations research*, 22(2) :266–275, 1997.

[SCZ00] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster : a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 8(3-4) :289–304, 2000.

[Sid80] J. Sidi. *L'approche logique en classification automatique et reconnaissance des formes*. PhD thesis, Université Pierre et Marie Curie, 1980.

[SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[SM90] A. Srivastava and M. Murty. A comparison between conceptual clustering and conventional clustering. *Pattern Recogn.*, 23(9) :975–981, 1990.

[SS63] R.R. Sokal and P.H. Sneath. *Principles of Numerical Taxonomy*. W. H. Freeman and Compagny, 1963.

[SS73] P. Sneath and R. Sokal. *Numerical Taxonomy*. Freeman, 1973.

[SW81] F.J. Solis and R.J.B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6 :19–30, 1981.

[SYK97] M. Sarkar, B. Yegnanarayana, and D. Khemani. A clustering algorithm using evolutionary programming-based approach. *Pattern Recognition Letters*, 18 :975–986, 1997.

[TWH00] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic, 2000.

[VLA87] P.J.M. Van-Laarhoven and E.H.L. Aarts. *Simulated Annealing : Theory and Applications*. D. Reidel Publishing Company, Dordretch, Holland, 1987.

[VNCM99] E. Van-Nimwegen, J.P. Crutchfield, and M. Mitchell. Statistical dynamics of the royal road genetic algorithms. *Theoretical Computer Science*, 229(1-2) :41–102, 1999.

[Vra02] M. Vrac. *Analyse et modélisation de données probabilistes par décomposition de mélanges de copules : Application à une base de données climatologiques*. PhD thesis, Université Paris IX-Dauphine, 2002.

[WF87] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society*, 49(3) :240–265, 1987.

[WKD96] Chris S. Wallace, Kevin B. Korb, and Honghua Dai. Causal discovery via MML. In *International Conference on Machine Learning*, pages 516–524, 1996.

[WL59] W.T. Williams and J.M. Lambert. Multivariate methods in plan ecology. *Journal of Ecology*, 47(1) :83–101, 1959.

[WYM97] W. Wang, J. Yang, and R.R. Muntz. STING : A statistical information grid approach to spatial data mining. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *Twenty-Third International Conference on Very Large Data Bases*, pages 186–195, Athens, Greece, 1997. Morgan Kaufmann.

[XEKS98] X. Xu, M. Ester, H. P. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *ICDE '98 : Proceedings of the Fourteenth International Conference on Data Engineering*, pages 324–331, Washington, DC, USA, 1998. IEEE Computer Society.

[ZDG+01] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering, 2001.

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. Birch : an efficient data clustering method for very large databases. In *SIGMOD '96 : Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114, New York, NY, USA, 1996. ACM Press.

[ZRL97] T. Zhang, R. Ramakrishnan, and M. Livny. Birch : A new data clustering algorithm and its applications, 1997.

Résumé :
Le travail de thèse a porté sur une réflexion relative aux méthodes de classification automatique des données pour lesquelles il est bien connu qu'un effet "méthode" existe. Après une première partie qui présente la problématique générale de l'analyse des données et propose un survey des méthodes de classification, les travaux originaux de la thèse sont exposés. Ils relèvent de trois approches interconnectées : une approche basée sur l'agrégation d'opinions, une approche prétopologique et une approche basée sur l'agrégation des préférences. Chacune de ces approches se fonde sur un paradigme différent et propose une nouvelle vision des techniques de classification permettant d'apporter éventuellement de l'information exogène dans la méthode.

Title :
Contribution to the data clustering methods via pretopological approaches and of opinions aggregation.

Abstract :
The work of thesis concerned a reflection relating to the clustering data for which it is well-known that an effect "method" exists. After a first part who presents the general problems of data analysis and propose a survey methods of classification, originals work of the thesis are exposed. They concern three inter-connected approaches : an approach based on the opinions aggregation, a pretopological approach and an approach based on preferences aggregation. Each one is based on different paradigm and proposes a new vision of the classification techniques allowing to bring exogenic information in the method.

Discipline :
Statistiques - Informatique

Mots-clés :
Classification, analyse des données, prétopologie, agrégation des préférences, proximité.

Key-words :
Clustering, data analysis, pretopology, preferences aggregation, proximity.

LIRIS UMR 5205 CNRS,
Equipe Méthodes et Algorithmes pour l'Aide à la Décision (MA2D).
Bâtiment Nautibus, 43, boulevard du 11 novembre 1918, VILLEURBANNE